

## Geração de Trajetória para VANT com Resultados Experimentais

Wenderson Gustavo Serrantola\* Fernando Lizarralde\*  
Alessandro Jacoud Peixoto\*

\* Departamento de Engenharia Elétrica, COPPE - Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ - Brasil, (e-mail: wenderson.serrantola@coppe.ufrj.br, f.lizarralde@coppe.ufrj.br, jacoud@poli.ufrj.br).

---

**Abstract:** This paper proposes a new approach for generating smooth trajectories for unmanned aerial vehicles (UAVs or drones) that can be executed in real time with limited processing embedded hardware. The objective is to perform autonomous aerial image acquisition, ensuring repeatability. In addition, a state feedback control law based on previously identified commercial hexacopter nominal parameters, with integral and feedforward actions was implemented and validated for trajectory tracking in the presence of wind disturbances, in a real and simulated environment.

**Resumo:** Este artigo propõe uma nova abordagem para geração de trajetórias suaves para veículos aéreos não tripulados (VANTs ou *drones*) que possam ser executadas em tempo real com *hardware* embarcado apresentando processamento limitado. O objetivo é realizar aquisição de imagens aéreas de forma autônoma, assegurando repetibilidade. Além disso, uma lei de controle via realimentação de estados baseada nos parâmetros nominais de um hexacoptero comercial previamente identificados, com ação integral e antecipativa (feedforward) foi implementada e validada para o rastreamento de trajetória na presença de distúrbios de vento, em ambiente real e simulado.

**Keywords:** Trajectory Generation, Trajectory Tracking, UAV PI Control, Parametric Differentiable Curves

**Palavras-chaves:** Geração de Trajetória, Rastreamento de Trajetória, Controle PI de VANT, Curvas Paramétricas Diferenciáveis

---

### 1. INTRODUÇÃO

O número de veículos aéreos não tripulados (VANT) vem aumentando na última década. Isso se deve à sua alta capacidade de manobrabilidade que permite sua utilização em diferentes aplicações, o que vem atraindo a comunidade científica para vários tópicos relacionados ao planejamento de movimento e controle, em especial, de sistemas autônomos.

A geração de trajetória pode parecer um problema clássico com soluções já estabelecidas. Entretanto, a natureza não linear e a complexidade destes sistemas, destacam a necessidade de um planejamento de trajetória preciso que respeite a dinâmica do sistema (Stoican et al., 2017).

Sua aplicação em VANTs vem evoluindo de uma simples abordagem do caminho mais curto (Gao et al., 2013) para abordagens mais complexas, como os métodos probabilísticos e baseados em amostragem (Xu et al., 2021; Saravanakumar et al., 2021), e os métodos de otimização (Su et al., 2019; Carvalho et al., 2018; Li et al., 2019). Mais recentemente, algumas abordagens utilizando redes neurais vêm sendo utilizadas (Lai et al., 2020; Li et al., 2022).

Assim como o planejamento de movimento, o projeto de controladores inteligentes para VANTs vem ganhando a atenção de muitos pesquisadores, uma vez que existem muitas questões referentes à segurança dos sistemas e arredores durante o voo. Técnicas clássicas como os controladores PID ou LQR têm sido aplicadas à maioria dos sistemas de controle de voo existentes para aproximar a dinâmica do veículo a um modelo linear. Tais controladores possuem simplicidade no projeto e em sua implementação (Abdelhay and Zakriti, 2019; Bouabdallah et al., 2004).

Esse trabalho propõe a geração de trajetórias baseadas em curvas paramétricas diferenciáveis para um hexacoptero comercial. Além de fornecer uma perspectiva para o problema de rastreamento de trajetória utilizando um controlador PI convencional com ação antecipativa.

Este artigo está organizado da seguinte forma. A seção 2 apresenta a formulação do problema. Enquanto isso, na Seção 3 é apresentado o método proposto para a geração da trajetória. A Seção 4 mostra o Controle Proporcional e Integral (PI). A seção 5 analisa os resultados dos testes experimentais. Por fim, a Seção 6 conclui este artigo.

## 2. FORMULAÇÃO DO PROBLEMA

Assumindo que o VANT está em um modo de voo controlado por velocidade, o rastreamento assintótico de velocidade quase global é assegurado no sentido de que o vetor de velocidade do VANT rastrea assintoticamente um vetor de velocidade comandado. Além disso, também foi assumido que neste modo de voo a taxa de variação do ângulo de *yaw* também rastrea assintoticamente um sinal de velocidade de comando.

Por exemplo, Lee et al. (2010) apresentam uma possível malha de controle interna para um quadrotor garantindo o modo de rastreamento de velocidade mencionado, com um transiente de convergência de rastreamento tão rápido quanto maior o ganho interno de controle de velocidade.

Em geral, para um hexacóptero comercial, é possível encontrar quatro subsistemas fracamente acoplados, sendo que cada sistema pode ser representado por um sistema linear de primeira ordem com precisão aceitável para trajetórias com velocidade nominal. Este comportamento foi verificado com o hexacóptero comercial DJI M600 Pro por meio de exaustivos experimentos de identificação em um ambiente externo, com velocidades de até  $3m/s$ , que é um limite razoável para uma tarefa de monitoramento. Esse limite ( $10,8km/h$ ) permite o monitoramento do perímetro de um quadrado de  $1,8km^2$  em  $30min$ .

Assim, considere que cada subsistema de grau de liberdade é uma planta linear incerta de segunda ordem variante no tempo dada por

$$\dot{x}_1(t) = x_2(t) + d_1(t), \quad (1)$$

$$\dot{x}_2(t) = -a_p(t)x_2(t) + k_p(t)u(t) + d_2(t), \quad (2)$$

$$y(t) = x_1(t), \quad (3)$$

sendo  $y$  a posição/*yaw* do drone no sistema de referência inercial,  $\dot{y}$  a velocidade correspondente e  $d_1$  e  $d_2$  distúrbios que representam o acoplamento entre os subsistemas e a influência do vento. O ganho incerto de alta frequência  $k_p(t)$  pode variar no tempo, assim como o parâmetro incerto  $a_p(t)$ . Assumimos que  $k_p$  possui norma limitada inferiormente para evitar a perda de controlabilidade e as derivadas no tempo  $\dot{d}_1$  e  $\dot{d}_2$  são uniformemente limitadas em norma, a menos de um conjunto de medida nula.

O objetivo do controle é encontrar uma lei de controle  $u(t)$  para que a saída  $y(t)$  em (3), da planta incerta variável no tempo (1)–(2), rastrea uma trajetória desejada limitada  $y_m(t)$  o mais próximo possível, ou seja, para que o erro de rastreamento

$$e(t) := y(t) - y_m(t), \quad (4)$$

convirja para zero quando  $t \rightarrow +\infty$ , ou pelo menos, para uma vizinhança de zero (rastreamento prático). Além disso, o sistema em malha fechada deve ser globalmente estável, assegurando que todos os sinais do sistema sejam limitados e, para quaisquer condições iniciais e sinais de entrada limitados, o estado completo do sistema do erro tenda a um pequeno erro residual.

Foi considerado que as trajetórias são suaves o suficiente para que a dinâmica interna do Filtro de Kalman do hexacóptero possa ser desconsiderada, fazendo com que

realimentação completa de estado seja assumida ao longo do artigo. Assim, assume-se que  $y$  e  $\dot{y}$  estão disponíveis para realimentação, bem como a velocidade e aceleração da trajetória desejada ( $\dot{y}_m$  e  $\ddot{y}_m$ ).

Aproveitando algum conhecimento nominal da planta (sem perturbações externas e variações de parâmetros), assumimos adicionalmente que existem valores nominais conhecidos  $a_p^n$  e  $k_p^n$  para o parâmetro incerto  $a_p(t)$  e  $k_p(t)$ , respectivamente. Para o rastreamento da trajetória gerada, propõe-se neste artigo a implementação de um controlador nominal, via realimentação de estado, com ação antecipativa e integral.

## 3. GERAÇÃO DE TRAJETÓRIA

Esta seção descreve o esquema de geração da trajetória utilizada. A Interface de Esboço de Caminho (PSI, do inglês *Path Sketch Interface*) foi desenvolvida em *Python* para permitir que o usuário desenhe o caminho 2D desejado (projeção no plano horizontal) para o drone na tela. O programa é um aplicativo autônomo contendo imagens de satélite e coordenadas da região de voo desejada.

O usuário desenha (na tela) o caminho de projeção orientado no plano horizontal, resultando em uma coleção de pontos 2D convertidos do espaço de pixel da tela em suas coordenadas correspondentes de latitude e longitude no mundo real. A altitude para cada ponto 2D da projeção da trajetória também é fornecida pelo usuário, resultando em uma coleção de pontos 3D (latitude, longitude e altitude), chamados de *BufferPoints* 3D.

As seguintes etapas são executadas pela PSI: (i) os pontos 3D (latitude, longitude e altitude) são convertidos para o sistema de referência local da operação do drone ENU (coordenadas  $x, y, z$ ); (ii) pontos consecutivos no caminho muito próximos, ou seja, distantes menos que uma distância pré-definida  $D_{min} > 0$  são removidos, mantendo o primeiro ponto e eliminando o segundo; e (iii) se três pontos consecutivos forem colineares, o ponto intermediário é eliminado, de modo que quaisquer três pontos subsequentes não sejam colineares. A posição inicial do drone está incluída na primeira posição do *buffer* de armazenamento (*BufferPoints* 3D).

### Seções do Caminho 2D (Aspectos Geométricos)

Dado o conjunto de  $N > 3$  pontos fornecido pela PSI, o conjunto de todos três pontos subsequentes possui  $N - 2$  elementos. Assim, para três pontos subsequentes genéricos  $P, Q$  e  $R$  ao longo do caminho 2D ( $P, Q, R \in \mathbb{R}^2$ ), defini-se um ponto de chegada ao redor de  $Q$ , denominado  $Q_{in} \in \overline{PQ}$ , e um ponto de partida  $Q_{out} \in \overline{QR}$  por meio das seguintes combinações convexas:

$$Q_{in} := (1 - \mu)P + \mu Q$$

e

$$Q_{out} := \left[ 1 - \frac{\|Q - Q_i\|}{\|Q - R\|} \right] Q + \left[ \frac{\|Q - Q_i\|}{\|Q - R\|} \right] R,$$

respectivamente, onde  $\mu \in (0, 1)$ . Para este conjunto de pontos  $P \rightarrow Q \rightarrow R$ , o caminho associado é definido em duas partes:  $P \rightarrow Q_{in}$  e  $Q_{in} \rightarrow Q_{out}$ .

Seja  $S_d$  o *sentido de rotação* associado ao caminho  $P \rightarrow Q \rightarrow R$ . Se o caminho  $P \rightarrow Q \rightarrow R$  resultar em uma

rotação no sentido horário (anti-horário) define-se  $\mathcal{S}_d = -1$  ( $\mathcal{S}_d = 1$ ). O sentido de rotação pode ser calculado pela seguinte função de  $P, Q$  e  $R$ :

$$\mathcal{S}_d = \text{sign} \{ [0 \ 0 \ 1] [(\bar{Q} - \bar{P}) \times (\bar{R} - \bar{Q})] \},$$

onde  $\bar{P} := [P^T \ 0]^T$ ,  $\bar{Q} := [Q^T \ 0]^T$  e  $\bar{R} := [R^T \ 0]^T$ .

Observe que ambas as seções  $P \rightarrow Q_{in}$  e  $Q_{in} \rightarrow Q_{out}$  têm os mesmos valores para  $\mathcal{S}_d$  e  $\mu$ .

Assim, para cada três pontos subsequentes, o caminho é dividido em duas partes, resultando em um total de  $2(N-2)$  seções. Logo, todo o caminho  $xy$  é então dividido em  $2(N-2)$  seções implementadas por curvas circulares paramétricas diferenciáveis planares  $\gamma_1(t), \gamma_2(t), \dots, \gamma_{2(N-2)}(t) : \mathbb{R}^+ \rightarrow \mathbb{R}^2$ . Cada seção de caminho único  $\gamma$  começa em  $P_i$  ( $P$  ou  $Q_{in}$ ) e termina em  $P_f$  ( $Q_{in}$  ou  $Q_{out}$ ).

Além disso, cada seção de cada caminho  $k = 1, \dots, 2(N-2)$  tem parâmetro de chegada/partida  $\mu_k$ , comprimento  $\delta s_k$ , raio de curvatura  $r_k$ , centro de curvatura  $C_k$ , ângulo central  $\beta_k$  do arco que subtende o segmento  $\overline{P_i P_f}$  e sentido de rotação  $\mathcal{S}_{dk}$ .

A seção  $k$ -th é definida no intervalo de tempo  $[T_{k-1}, T_k]$ , onde  $T_k = T_{k-1} + \delta t_k$ ,  $T_0 := 0$  e  $\delta t_k$  é a duração do trecho, que depende de  $P_i, P_f, r_k$  e da velocidade desejada ao longo do caminho. Observe que, definindo valores grandes para  $r_k$ , o caminho se aproxima de uma linha reta de  $P_i$  a  $P_f$ .

A função

$$s_k(t) := \int_{T_{k-1}}^t \|\dot{\gamma}_k(\tau)\| d\tau, \quad T_{k-1} \leq t \leq T_k,$$

é o comprimento da  $k$ -th curva paramétrica no intervalo de tempo  $[T_{k-1}, t] \subset [T_{k-1}, T_k] \subset [0, T]$ , onde  $T := T_{2(N-2)}$  é o instante de tempo final para toda a trajetória (duração total da trajetória).

O comprimento  $\delta s_k$  da  $k$ -ésima curva paramétrica pode ser expresso como  $\delta s_k = s_k(T_k) := \int_{T_{k-1}}^{T_k} \|\dot{\gamma}_k(\tau)\| d\tau$ . Assim, o comprimento da curva paramétrica no intervalo de tempo  $[0, t]$  é definido pela seguinte função contínua

$$s(t) := \delta s_1 + \delta s_2 + \dots + \delta s_{k-1} + s_k(t),$$

onde  $k$  é o índice tal que  $T_{k-1} \leq t \leq T_k$ . Portanto, pode-se escrever que

$$\dot{s}(t) = \|\dot{\gamma}_k(t)\| > 0, \quad T_{k-1} \leq t \leq T_k.$$

### O Incremento do Índice $k$

Começando com  $k = 1$ , tem-se que

$$\dot{s}(t) = \|\dot{\gamma}_1(t)\| > 0, \quad T_0 = 0 \leq t \leq T_1,$$

ou

$$\dot{s}(t) = \|\dot{\gamma}_1(t)\| > 0, \quad s < \delta s_1,$$

onde  $\delta s_1$  é obtido *a priori* usando a informação geométrica.

Então, após  $s$  atingir  $\delta s_1$ , o índice  $k$  é incrementado. Logo, para  $k = 2$ ,

$$\dot{s}(t) = \|\dot{\gamma}_2(t)\| > 0, \quad s < \delta s_1 + \delta s_2.$$

Este procedimento permite implementar a trajetória desejada sem conhecer os instantes de comutação  $T_k$ . Apenas as informações geométricas são necessárias:

$$\dot{s}(t) = \|\dot{\gamma}_k(t)\| > 0, \quad \sum_1^{k-1} \delta s_j < s < \sum_1^k \delta s_j,$$

com  $\delta s_0 := 0$ .

Como  $\dot{s}(t) > 0, \forall t \geq 0$ ,  $s(t)$  sempre atinge os limites de comutação do índice  $k$   $\delta s_1, \delta s_1 + \delta s_2, \dots, \delta s_1 + \delta s_2 + \dots + \delta s_{2(N-2)-1}$ .

O vetor de posição é definido pela seguinte função contínua

$$p(t) := \gamma_k(t), \quad t \in [T_k, T_{k+1}],$$

ou, de forma equivalente,

$$p(t) := \gamma_k(t), \quad \sum_1^{k-1} \delta s_j < s < \sum_1^k \delta s_j.$$

No entanto,  $\dot{p}$  e  $\ddot{p}$  são funções contínuas de  $t$ , *módulo* de um conjunto de medidas zero (contínuo por partes).

A trajetória suave final  $p_f$  é obtida filtrando  $p$ , de acordo com

$$p_f = \frac{1}{(\tau_f s + 1)^3} p, \quad \tau_f > 0.$$

Os sinais suaves finais  $p_d, v_d, a_d$  são dados pelos estados desse filtro.

Para simplificar a notação, eliminamos o subscrito  $k$ , de modo que:  $\gamma_k \rightarrow \gamma, \delta s_k \rightarrow \delta s, r_k \rightarrow r, C_k \rightarrow C, \beta_k \rightarrow \beta, \mathcal{S}_{dk} \rightarrow \mathcal{S}_d$  e  $\mu_k \rightarrow \mu$ .

Além disso, foi definido  $t_i = T_k$  e  $t_f := T_{k+1}$ .

### Trajetória Baseada em Curvas Paramétricas Diferenciáveis: Uma Seção do Caminho

Considerando uma seção de caminho genérico  $P \rightarrow Q \rightarrow R$  dividido em duas partes:  $P \rightarrow Q_{in}$  e  $Q_{in} \rightarrow Q_{out}$ . E seja  $P_i$  e  $P_f$  os pontos inicial e final de cada parte, ou seja,  $P_i = P$  e  $P_f = Q_{in}$ , ou  $P_i = Q_{in}$  e  $P_f = Q_{out}$ . O centro de curvatura  $C$  pode ser determinado resolvendo o sistema de equações:

$$\|C - P_i\| = r \quad \text{e} \quad (P_i - P_f)^T C = (P_i - P_f)^T M,$$

onde  $M = (P_i + P_f)/2$  e  $r$  o raio de curvatura. A segunda igualdade vem do fato que  $(P_i - P_f) \perp (C - M)$ . Como existem duas soluções possíveis (para  $r > \|P_f - P_i\|/2$ ), deve-se verificar qual solução  $C$  pertence à região côncava definida por  $P \rightarrow Q \rightarrow R$ . Isso pode ser obtido encontrando qual solução é a mais próxima do ponto de interseção  $X$  da mediatriz dos segmentos de reta  $\overline{PQ}$  e  $\overline{QR}$ . Observe que  $X$  pode ser determinado resolvendo o sistema de equações:

$$(P - Q)^T (X - M_{pq}) = 0, \quad (Q - R)^T (X - M_{qr}) = 0,$$

onde  $M_{pq} := (P + Q)/2$  e  $M_{qr} := (Q + R)/2$  são os pontos médios dos segmentos de reta  $\overline{PQ}$  e  $\overline{QR}$ , respectivamente.

Em seguida, é possível obter: (i) o ângulo central  $\beta = 2\arcsin\left(\frac{\|P_f - P_i\|}{2r}\right)$ , do arco que compreende o segmento  $\overline{P_i P_f}$ ; (ii) o comprimento de cada seção de caminho  $\delta s := \beta r$ ; e (iii) a direção de rotação do caminho  $\mathcal{S}_d$ , ou seja, se o caminho  $P_i \rightarrow P_f$  resultar em uma rotação no sentido horário (anti-horário) configura-se  $\mathcal{S}_d = -1$  ( $\mathcal{S}_d = 1$ ).

Neste ponto, uma curva paramétrica é empregada para gerar o caminho  $P_i \rightarrow P_f$ , denominado:

$$\gamma(t) := C + \begin{bmatrix} r \cos[\alpha(t)] \\ r \sin[\alpha(t)] \end{bmatrix}, \quad t_i \leq t \leq t_f,$$

onde  $\alpha$  satisfaz

$$\dot{\alpha} := \mathcal{S}_d V(t)/r,$$

com  $V(t) > 0$  sendo definido posteriormente e a condição inicial

$$\alpha(t_i) := \arccos \left( [1 \ 0] \frac{(P_i - C)}{r} \right) \text{sign}(s_\alpha),$$

onde  $s_\alpha := [0 \ 1] (P_i - C)$ , sendo o ângulo de deslocamento necessário que garante que o vetor de posição  $\gamma$  comece no ponto correto  $P_i$ , ou seja,  $\gamma(t_i) = P_i$ .

A curva  $\gamma$  representa o caminho desejado a ser rastreado pelo drone e a velocidade instantânea desejada do drone em um determinado ponto  $P$  ao longo do caminho é expressa pelo vetor tangente à curva em  $P$ , ou seja, o vetor

$$\dot{\gamma}(t) := \begin{bmatrix} -\sin[\alpha(t)] \\ \cos[\alpha(t)] \end{bmatrix} S_d V(t).$$

A magnitude do vetor tangente  $\|\dot{\gamma}(t)\|$  é a velocidade no instante  $t$ . Assim, a velocidade é dada por  $\|\dot{\gamma}\| = rV$ . A aceleração é dada por

$$\ddot{\gamma}(t) := Q \begin{bmatrix} V^2 \\ S_d \dot{V} \end{bmatrix}, \quad Q := \begin{bmatrix} -\cos[\alpha(t)] & -\sin[\alpha(t)] \\ -\sin[\alpha(t)] & \cos[\alpha(t)] \end{bmatrix}$$

sendo  $Q$  uma matriz ortogonal. A aceleração será usada para definir um limite superior para o valor de pico.

#### Perfil de Velocidade $V(t)$

A velocidade é projetada como a função suave de  $t$  dada por:

$$V(t) := V_0 + (V_f - V_0) \text{Sig}(t - t_s) - (V_f - V_0) \text{Sig}(t_i - t_s) > 0,$$

onde

$$\text{Sig}(t) := \frac{1}{1 + e^{-k_s t}}, \quad k_s > 0,$$

é uma função sigmóide,  $t_s \in (t_i, t_f)$  é o tempo de comutação da sigmóide,  $k_s$  é o ganho da sigmóide e  $V_0, V_f$  recebem constantes positivas que representam a velocidade inicial do caminho e a velocidade em final do caminho, respectivamente. Observe que  $V(t_i) = V_0$  e

$$V(t_f) = V_0 + (V_f - V_0) \left[ \frac{1}{1 + e^{-k_s(t_f - t_s)}} - \frac{1}{1 + e^{k_s(t_s - t_i)}} \right].$$

Seja  $0 < \epsilon_f < 1$  uma constante de projeto que garanta uma precisão pré-definida para a velocidade no final do trecho, conforme:

$$V(t_f) = V_f - \epsilon_f \text{sign}(V_f - V_0).$$

Após algumas simples manipulações algébricas, foi verificado que a seguinte relação deve ser mantida

$$\left[ \frac{1}{1 + e^{-k_s(t_f - t_s)}} - \frac{1}{1 + e^{k_s(t_s - t_i)}} \right] = 1 - \frac{\epsilon_f}{|V_f - V_0|},$$

ou, de forma equivalente,

$$\frac{1}{1 + e^{-k_s(t_f - t_s)}} = 1 - \frac{\epsilon_f}{|V_f - V_0|} + \frac{1}{1 + e^{k_s(t_s - t_i)}},$$

de onde se obtém

$$t_f = t_s - \frac{1}{k_s} \ln \left\{ \frac{1}{\left[ 1 - \frac{\epsilon_f}{|V_f - V_0|} + \frac{1}{1 + e^{k_s(t_s - t_i)}} \right]} - 1 \right\}.$$

Além disso, pode-se obter

$$\int_{t_i}^{t_f} V(\tau) d\tau = \bar{V}_0(t_f - t_i) + \frac{(V_f - V_0)}{k_s} \ln \left( \frac{1 + e^{k_s(t_f - t_s)}}{1 + e^{k_s(t_i - t_s)}} \right),$$

$$\bar{V}_0 := V_0 - (V_f - V_0) \text{Sig}(t_i - t_s).$$

Agora, lembrando que

$$\dot{s}(t) = \|\dot{\gamma}(t)\| = V,$$

$\delta s$  pode ser definido por

$$\delta s = \bar{V}_0(t_f - t_i) + \frac{(V_f - V_0)}{k_s} \ln \left( \frac{1 + e^{k_s(t_f - t_s)}}{1 + e^{k_s(t_i - t_s)}} \right).$$

Substituindo  $t_f$  em  $\delta s$ , verifica-se que  $\delta s$  depende de  $k_s$  e  $t_s$ . Além disso,  $k_s$  é uma função crescente de  $t_s$ . Assim, a ideia é definir  $k_s$  e, posteriormente, encontrar o valor mínimo para  $t_s$  tal que a igualdade acima seja válida.

#### Incluindo o Componente de Altitude $z$ : Trajetória 3D

Primeiramente, a partir dos pontos obtidos pelo PSI, somamos a altitude desejada para cada ponto. Em seguida, determinamos a altitude dos pontos de chegada/saída ( $Q_{in}$  ou  $Q_{out}$ ) assumindo uma relação linear entre dois pontos consecutivos. Assim,  $P_i^z$  e  $P_f^z$  é a altitude de dois pontos consecutivos em uma única seção  $k$ .

Em seguida, a altitude  $z$  é definida em função do comprimento do caminho 2D  $s$ , de acordo com

$$z(t) = \frac{(P_f^z - P_i^z)}{(s_f - s_i)}(s(t) - s_i) + P_i^z, \quad s_i < s < s_f,$$

onde  $s_i := \sum_1^{k-1} \delta s_j$  e  $s_f := \sum_1^k \delta s_j$ . Observe que,

$$\dot{z} = \frac{(P_f^z - P_i^z)}{(s_f - s_i)} \dot{s} = \frac{(P_f^z - P_i^z)}{(s_f - s_i)} V.$$

Portanto, os vetores de posição e velocidade 3D, além da velocidade ao longo do caminho 3D são dados por

$$\begin{bmatrix} \gamma(t) \\ z(t) \end{bmatrix}, \quad \begin{bmatrix} \dot{\gamma}(t) \\ \dot{z}(t) \end{bmatrix} \quad \text{e} \quad V \sqrt{\left[ \frac{(P_f^z - P_i^z)}{(s_f - s_i)} \right]^2 + 1},$$

respectivamente. A trajetória 3D real é finalmente obtida filtrando o vetor de posição 3D como apresentado anteriormente.

## 4. CONTROLE

Como método de simplificação das notações, a variável independente  $t$  foi omitida. Com base nas medidas de  $y$  e  $\dot{y}$ , considere o **Sinal Auxiliar para Projeto do Controle** definido como

$$\sigma := \dot{e} + l_0 e, \quad (5)$$

e que satisfaz  $\dot{\sigma} = \ddot{y} + l_0 \dot{y} - (\ddot{y}_m + l_0 \dot{y}_m)$ . De (1)-(3), tem-se  $\dot{y} = x_2 + d_1$  e  $\ddot{y} = -a_p x_2 + k_p u + d_2 + \dot{d}_1$ . Então, a dinâmica  $\sigma$  é dada por

$$\dot{\sigma} = k_p u + d, \quad (6)$$

onde

$$d := [l_0 - a_p] \dot{y} + a_p d_1 + d_2 + \dot{d}_1 - (\ddot{y}_m + l_0 \dot{y}_m). \quad (7)$$

A ideia é projetar um sinal de controle  $u$  de modo que  $\sigma$  tenda a zero como  $t \rightarrow +\infty$ , ou pelo menos, próximo de zero, apesar da perturbação  $d$ . Assim, a convergência do erro de rastreamento para um conjunto residual é assegurada pelo ajuste de  $l_0 > 0$ , conforme (5).

A lei de controle nominal  $u^n$  é definida por

$$k_p^n u^n := -[l_0 - a_p^n] \dot{y} + (\ddot{y}_m + l_0 \dot{y}_m) - \theta \sigma, \quad (8)$$

onde  $\theta > 0$  é uma constante de projeto e os valores nominais  $a_p^n$  e  $k_p^n$  dos parâmetros variáveis no tempo  $a_p(t)$

e  $k_p(t)$  foram considerados. Agora, seja o sinal de controle composto por dois termos

$$u = u^n + u_s,$$

onde  $u_s$  será definido posteriormente. Então, a dinâmica  $\sigma$  pode ser reescrita como

$$\dot{\sigma} + \frac{k_p}{k_p^n} \theta \sigma = k_p u_s + D, \quad (9)$$

onde

$$D := \frac{k_p}{k_p^n} [-l_0 - a_p^n] \dot{y} + (\ddot{y}_m + l_0 \dot{y}_m) + d.$$

Considera-se uma simples ação de controle PI dada por

$$u_s = u_{pi} := -K_p \sigma - K_i \int_0^t \sigma(\tau) d\tau. \quad (10)$$

Neste caso, a dinâmica de malha fechada  $\sigma$  satisfaz

$$\ddot{\sigma} + [\theta + k_p^n K_p] \dot{\sigma} + [k_p^n K_i] \sigma = 0,$$

a partir do qual podemos projetar  $K_p$  e  $K_i$ .

## 5. RESULTADOS EXPERIMENTAIS

A trajetória desejada foi criada para ser executada no campo de futebol (livre de obstáculos e barreiras) ao lado do laboratório na Universidade Federal do Rio de Janeiro. O caminho foi obtido utilizando a interface PSI e uma trajetória suave foi gerada (via Matlab) pelas curvas paramétricas diferenciáveis descritas na Seção 3. Os pontos georreferenciados foram convertidos para o sistema de referência ENU (do inglês, *east-north-up*) que utiliza as coordenadas cartesianas ( $x, y, z$ ) para representar a posição do drone em relação a uma origem local previamente definida.

O objetivo principal é verificar o desempenho do rastreamento da trajetória previamente gerada para o hexacoptero comercial em um ambiente real (na presença de distúrbios de vento), utilizando o controlador PI em malha fechada. Inicialmente, o desempenho em ambiente simulado será observado.

O simulador considerado neste trabalho foi o DJI Assistant 2<sup>1</sup>, um programa desenvolvido pela empresa DJI que permite aos usuários fazerem upload de dados de voo, calibrar sensores de visão, além de fornecer um simulador com dinâmica muito próxima do drone real.

Nos experimentos em ambiente simulado, foram realizados testes com e sem perturbação constante de vento. Para o teste com perturbação, o vento foi adicionado ao longo dos três eixos ( $x, y$  e  $z$ ), aproximadamente no tempo  $t = 30s$ . Para os eixos  $x$  e  $y$ , a velocidade do vento de  $8 \text{ m/s}$  foi introduzida no sentido positivo do movimento. Para o eixo  $z$ , foi considerada a velocidade do vento de  $2 \text{ m/s}$  na direção ascendente.

Após o teste no simulador, a trajetória e o controlador PI foram avaliados em um ambiente representativo, no campo de futebol citado anteriormente. Os experimentos foram realizados no dia 20 de abril de 2022 com condições de vento moderado, ou seja, velocidade variando de  $3 \text{ m/s}$  a  $4 \text{ m/s}$ , de acordo com o anemômetro instalado em campo.

Os esforços de controle nos testes simulados são fornecidos na Figura 1 e nos testes em campo na Figura 2. Para uma

<sup>1</sup> <https://www.dji.com/downloads/softwares/assistant-dji-2>.

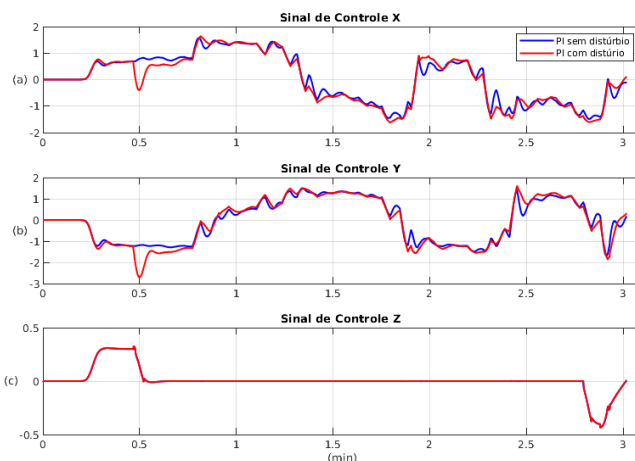


Figura 1. Simulador DJI Assistant. Sinal de controle ao longo dos três eixos. Em azul é apresentado o resultado sem perturbação de vento e em vermelho com perturbação de vento constante.

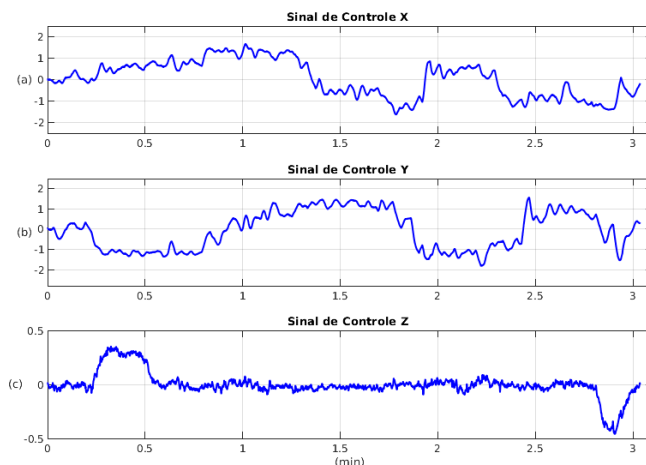


Figura 2. Teste de Campo. Sinal de controle ao longo dos três eixos.

melhor visualização dos resultados dos testes no simulador e no campo, o desempenho do rastreamento da trajetória de ambos os casos são representados na Figura 3, onde em azul é representado o experimento no simulador sem distúrbio, em vermelho o teste no simulador com distúrbio e em verde o teste no campo.

Para avaliar o efeito de distúrbios, um perturbação mais agressiva foi introduzida na simulação (vento de  $8 \text{ m/s}$ ). Tal efeito pode ser observado no gráfico  $xy$  apresentado na Figura 3, veja a linha vermelha perto do ponto  $(10, -20)$ . Entretanto, verifica-se que mesmo neste caso agressivo, a ação integral foi capaz de corrigir o efeito da perturbação no seguimento do caminho (em regime permanente) após um transitório. Por outro lado, este transitório comprometeu o valor RMSE (*root mean squared error*) (Chai and Draxler, 2014) final do erro, como pode ser visto na segunda coluna da Tabela 1. Esta tabela apresenta o erro RMSE para cada um dos testes (simulador e campo). Adicionalmente, pode-se observar na Tabela 1 que os testes no simulador sem distúrbio e no campo apresentaram baixo erro RMSE. Isso ilustra o bom desempenho do controlador

## REFERÊNCIAS

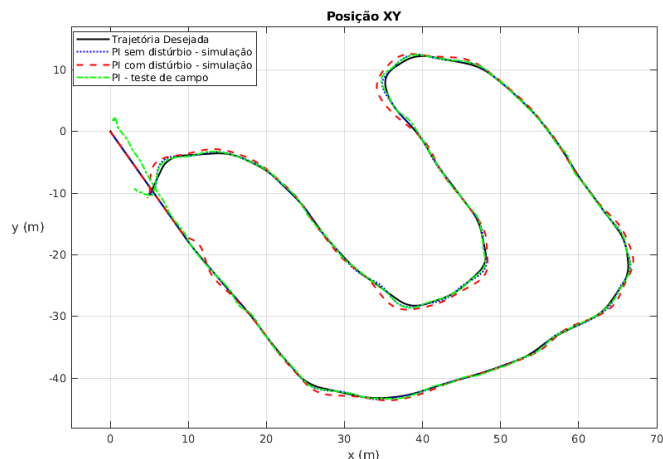


Figura 3. Desempenho do rastreamento de caminho ( $xy$ ) para o controlador PI. Em azul é apresentado o resultado sem perturbação de vento, em vermelho o resultado com perturbação constante de vento, ambos no simulador DJI Assistant 2, e em verde o resultado do teste de campo. O caminho desejado é apresentado em preto.

em campo na presença de vento da ordem de  $3 - 4\text{m/s}$  (medido no dia 20 de abril de 2022).

Tabela 1. RMSE

Eixos	Simulação		Teste de Campo
	sem distúrbio	com distúrbio	
$x$	0.0906	0.5046	0.1261
$y$	0.1156	0.5639	0.1475
$z$	0.0086	0.0086	0.0323

Comparações com dois outros esquemas de controle baseados em modos deslizantes podem ser encontrados em Serrantola et al. (2022): (i) controlador baseado em modo deslizante de primeira ordem capaz de fornecer sinal de controle suave por meio de filtragem do sinal de controle e o uso de uma malha interna de previsão, denominado SSC (*Smooth Sliding Control*), e (ii) controlador baseado em modo deslizante de segunda ordem utilizando o algoritmo *Super Twisting*.

## 6. CONCLUSÃO

Neste artigo, uma nova abordagem para geração de trajetória foi avaliada e implementada com sucesso. A principal ideia do trabalho foi a geração de trajetórias suaves e que possam ser executadas em tempo real, para a aquisição de imagens em ambientes relevantes sem obstáculos. Além disso, um controlador PI foi implementado e validado para o rastreamento da trajetória gerada na presença de distúrbios de vento, em ambiente real e simulado, para um hexacoptero comercial.

## AGRADECIMENTOS

Este trabalho foi financiado em parte pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001 e 88887.136349/2017-00, FAPERJ, CNPq, Repsol Sinopec Brasil e ANP.

- Abdelhay, S. and Zakriti, A. (2019). Modeling of a quadcopter trajectory tracking system using pid controller. *Procedia Manufacturing*, 32, 564–571.
- Bouabdallah, S., Noth, A., and Siegwart, R. (2004). Pid vs lq control techniques applied to an indoor micro quadrotor. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 3, 2451–2456. IEEE.
- Carvalho, B., Di Perna, M., and Rodrigues, L. (2018). Real-time optimal trajectory generation for a quadrotor uav on the longitudinal plane. In *2018 European Control Conference (ECC)*, 3132–3136. IEEE.
- Chai, T. and Draxler, R.R. (2014). Root mean square error (rmse) or mean absolute error (mae)?—arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3), 1247–1250.
- Gao, X.Z., Hou, Z.X., Zhu, X.F., Zhang, J.T., and Chen, X.Q. (2013). The shortest path planning for manoeuvres of uav. *Acta Polytechnica Hungarica*, 10(1), 221–239.
- Lai, R., Zhao, D., and Mishra, S. (2020). A machine-learning approach for time-optimal trajectory generation for uav’s. In *Proceedings of the 76th Vertical Flight Society Annual Forum, Virginia Beach, Virginia*.
- Lee, T., Leok, M., and McClamroch, N.H. (2010). Control of complex maneuvers for a quadrotor uav using geometric methods on se (3). *arXiv preprint arXiv:1003.2005*.
- Li, Y., Eslamiat, H., Wang, N., Zhao, Z., Sanyal, A.K., and Qiu, Q. (2019). Autonomous waypoints planning and trajectory generation for multi-rotor uavs. In *Proceedings of the Workshop on Design Automation for CPS and IoT*, 31–40.
- Li, Y., Fang, H., Li, M., Ma, Y., and Qiu, Q. (2022). Neural network pruning and fast training for drl-based uav trajectory planning. In *2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 574–579. IEEE.
- Saravanakumar, A., Kaviyarasu, A., and Ashly Jasmine, R. (2021). Sampling based path planning algorithm for uav collision avoidance. *Sādhanā*, 46(3), 1–8.
- Serrantola, W.G., Lizarralde, F., and Peixoto, A.J. (2022). From the modified smooth sliding control to the super-twisting algorithm: UAV trajectory tracking experimental results. In *16th International Workshop on Variable Structure Systems and Sliding Mode Control (VSS)*, To appear. IEEE.
- Stoican, F., Prodan, I., Popescu, D., and Ichim, L. (2017). Constrained trajectory generation for uav systems using a b-spline parametrization. In *2017 25th Mediterranean Conference on Control and Automation (MED)*, 613–618. IEEE.
- Su, J., Bak, J.H., and Hyun, J. (2019). Optimal trajectory generation for quadrotor with suspended load under swing angle constraint. In *2019 IEEE 15th International Conference on Control and Automation (ICCA)*, 549–554. doi:10.1109/ICCA.2019.8899662.
- Xu, Z., Deng, D., and Shimada, K. (2021). Autonomous uav exploration of dynamic environments via incremental sampling and probabilistic roadmap. *IEEE Robotics and Automation Letters*, 6(2), 2729–2736.