Revista de Informática Teórica e Aplicada - RITA - ISSN 2175-2745
Vol. 29, Num. 03 (2022) 21-35

RESEARCH ARTICLE

Optimal Control and Adaptive Learning for Stabilization of a Quadrotor-type Unmanned Aerial Vehicle via Approximate Dynamic Programming

Controle Ótimo e Aprendizagem Adaptativa para Estabilização de um Veículo Aéreo Não Tripulado do Tipo Quadrirrotor Via Programação Dinâmica Aproximada

Joelson Miller Bezerra de Sousa[1]*, Patrícia Helena Moraes Rêgo[2], Guilherme Bonfim de Sousa[1], Janes Valdo Rodrigues Lima[1]

Resumo: The development of an optimal controller for stabilization of a quadrotor system using an adaptive critic structure based on policy iteration schemes is proposed in this paper. This approach is inserted in the context of Approximate Dynamic Programming and it is used to solve optimal decision problems on-line, without requiring complete knowledge of the system dynamics model to be controlled. The main feature of the adaptive critic design method that allows for on-line implementation is that it solves the Bellman optimality equation in a forward-in-time fashion, whereas traditional dynamic programming requires a backward-in-time procedure. This feedback control design technique is able to tune the controller parameters on-line in the presence of variations in plant dynamics and external disturbances using data measured along the system trajectories. Computational simulation results based on a quadrotor model demonstrate the effectiveness of the proposed control scheme.
Keywords: Optimal Control — Quadrotor — Policy Iteration — Adaptive Critic Design — Approximate Dynamic Programming

Resumo: Neste artigo é proposto o desenvolvimento de um controlador ótimo para estabilização de um sistema quadrirrotor por meio da utilização de uma estrutura crítico-adaptativa baseada em esquemas de iteração de política. Esta abordagem está inserida no contexto de Programação Dinâmica Aproximada e é usada para resolver problemas de decisão ótima *on-line*, sem requerer o conhecimento completo do modelo da dinâmica do sistema a ser controlado. A principal característica do método de projeto crítico-adaptativo que permite a implementação *on-line* é que ele resolve a equação da otimalidade de Bellman em uma maneira "para frente no tempo", enquanto a programação dinâmica tradicional requer um procedimento "para trás no tempo". Esta técnica de projeto de controle de realimentação é capaz de sintonizar *on-line* os parâmetros do controlador na presença de variações na dinâmica da planta e perturbações externas a partir dos dados medidos ao longo das trajetórias do sistema. Resultados de simulação computacional baseados em um modelo do quadrirrotor demonstram a eficácia do esquema de controle proposto.
Palavras-Chave: Controle Ótimo — Quadrirrotor — Iteração de Política — Projeto Crítico-Adaptativo — Programação Dinâmica Aproximada

[1] *Programa de Pós-Graduação em Engenharia da Computação e Sistemas (PECS), Universidade Estadual do Maranhão (UEMA), São Luís - Maranhão, Brasil*
[2] *Departamento de Matemática e Informática (DEPMAT) , Universidade Estadual do Maranhão (UEMA), São Luís - Maranhão, Brasil*
***Corresponding author**: joelsonmiller21@gmail.com
DOI: http://dx.doi.org/10.22456/2175-2745.121388 • **Received:** 05/01/2022 • **Accepted:** 19/09/2022

1. Introdução

O estado da arte em Veículos Aéreos Não Tripulados (VANTS), *Unmanned Aerial Vehicle* (UAV), tem recebido diversas contribuições e avanços em pesquisas nos últimos anos [1, 2, 3, 4, 5, 6]. Tais avanços têm sido motivados, principalmente, pelos desenvolvimentos em tecnologia de sensoriamento, armazenamento de energia de alta densidade e processamento de dados [7]. UAV é caracterizado por ser uma aeronave sem pilotos a bordo. O voo desse veículo pode ser controlado autonomamente por computador de bordo ou por controle remoto de um piloto no solo ou em outro veículo. Um dos

modelos de VANTs mais populares é o helicóptero quadrirrotor, também chamado quadricóptero [8]. A pesquisa em UAV tem se focado bastante na estrutura mecânica dos quadricópteros, estabilidade a pertubações externas, desvio de obstáculos, controle tolerante a falhas, modelagem e tarefa com múltiplos agentes [8, 9].

Os quadricópteros se tornaram interessantes para o ramo da pesquisa a partir do século XX, ao mesmo tempo que outros veículos aéreos, devido ao fato da capacidade destes de decolarem e pousarem verticalmente, e conseguirem realizar o movimento de pairarem durante a execução do voo, isto é, maior manobrabilidade e versatilidade quando comparados a aviões, que necessitam de velocidade para manter a força de empuxo entre as asas com o propósito de gerarem sustentação [10]. Ao contrário do helicóptero de rotor único, o rápido ajuste de elevação é realizado pelo controle das velocidades angulares das hélices. Devido à estrutura de múltiplos rotores, seus momentos ante-torques podem ser cancelados entre si. Devido à estrutura simples, um quadricóptero é fácil de usar e apresenta alta confiabilidade e baixo custo de manutenção [3, 4]. Estes podem ser organizados em rotores coplanares, em que ambos fornecem empuxo para cima, mas com pares que giram para direções opostas, com o propósito de equilibrar os torques exercidos sobre o corpo do veículo [11]. O controle do quadricóptero tem sua complexidade no fato dele possuir os movimentos rotacional e translacional acoplados, o que gera os seis graus de liberdade e quatro entradas independentes (as velocidades angulares das hélices – rotores) [6].

Segundo [9], grande parte das publicações científicas acerca de quadricópteros tem se concentrado na solução de algoritmos de controle. Dentre as técnicas de controle aplicadas, pode-se destacar as seguintes: a abordagem de Lyapunov [12], a qual garante, sob certas condições, a estabilidade assintótica do quadricóptero; a estrutura de realimentação PD, Proporcional-Derivativo [13, 14], com propriedade de convergência exponencial devido à compensação dos termos Coriolis e giroscópicos, e a estrutura PID, Proporcional-Integral-Derivativo [7, 15], a qual não requer o conhecimento de parâmetros específicos do modelo e a lei de controle é muito mais fácil de implementar, porém com robustez limitada contra perturbações; o controle RLQ (Regulador Linear Quadrático) [16], e controle H_∞ [4], cuja vantagem é que exibem boas propriedades de robustez: margem de ganho infinitamente crescente, margem de fase entre $\pm 60^\circ$ e boa tolerância à não-linearidades; controle adaptativo [17, 18, 19], que fornecem bom desempenho com parâmetros incertos e dinâmicas não modeladas. Existem outros algoritmos de controle para melhorias do desempenho de sistemas quadrirrotores, tais como técnicas *fuzzy* [1, 20], redes neurais [21], controle *backstepping* [22, 23], controle baseado em realimentação visual [24, 25], e controle baseado em aprendizagem por reforço [26, 27, 28]. Em [29] são destacados vários projetos de melhoria de estabilidade para quadricópteros, que são capazes de realizar voos autônomos apenas com o uso de sensores a bordo para estimação da atitude, altitude, posição horizontal e voos translacionais.

A otimização do esforço de controle no processo de decolagem e aterrissagem vertical – controle de altitude – e o projeto de um controlador ótimo com características de adaptabilidade tem um peso relevante, principalmente para o controle de trajetórias de voo, devido ao seu regulador auto ajustável. Além disso, aeronaves de asas rotativas necessitam de um controle permanentemente atuante para manterem sua estabilidade e executarem manobras, ou seja, eles devem fornecer seu próprio amortecimento para parar de se moverem e permanecerem estáveis. A dinâmica resultante é não linear, especialmente depois de contabilizar os efeitos aerodinâmicos [30]. Um controlador ótimo visa minimizar uma métrica (índice de desempenho) do sistema dinâmico no tempo de modo a obter um sinal de controle que seja capaz de executar seu objetivo ao mesmo tempo que respeita as restrições e requisitos exigidos no projeto. A equação de Hamilton-Jacobi-Bellman (HJB) é a base para implementação de algoritmos de programação dinâmica [31].

A Programação Dinâmica Aproximada (PDA), proposta por Werbos [32], é um paradigma para o aprendizado *on-line* de controle ótimo. A PDA usa estruturas crítico-adaptativas para aproximar a solução da equação HJB e pode ser implementada por dois dos métodos mais populares: iteração de valor e iteração de política [33]. A premissa de que existe uma relação de causa e efeito entre ações e recompensas é inerente à aprendizagem animal. Tal princípio, caracterizado por fortes capacidades de autoaprendizagem e adaptação, substancia a Programação Dinâmica Aproximada. A PDA permite projetar controladores que aprendem as soluções para problemas de controle ótimo sem que haja conhecimento completo do modelo da dinâmica do sistema e de forma *on-line*, ou seja, com base nos dados medidos, em tempo real, ao longo da trajetória do sistema [34, 35]. A PDA resolve os problemas de maneira "para frente no tempo". Este método é baseado em Aprendizagem por Reforço (*Reinforcement Learning* - RL) para aproximar a solução ótima de uma função de custo (função valor) que garanta otimalidade ao longo do tempo. Uma visão geral das técnicas de PDA e seus avanços focados em controle ótimo são apresentados em [36, 37].

Apesar das técnicas de controle ótimo terem alcançado sucesso em muitas aplicações em sistemas quadrirrotores, um aspecto que deve ser observado é que os projetos de controladores resultantes desses métodos, em geral, são realizados *off-line*, ou seja, os ganhos dos controladores não dependem das trajetórias dos estados e são calculados baseados no modelo detalhado da dinâmica do quadrirrotor [38], como observado em [39], [40], [41] e [42]. Neste artigo, o problema de estabilização de um quadrirrotor é enfatizado sob o ponto de vista da pesquisa e desenvolvimento de sistemas de controle ótimo baseados em PDA. Os benefícios de tal abordagem são que, além de promover a redução no esforço de controle durante o movimento de voo pairado do veículo, os ganhos do controlador são automaticamente ajustados à medida que o processo dinâmico está em operação, o que é viável para situa-

ções tais como variações abruptas na massa e no momento de inércia do veículo devido a adição de uma carga de trabalho. Neste contexto, é apresentado um método de aproximação de Iteração de Política baseado em redes neurais treinadas pelo algoritmo recursivo dos mínimos quadrados (RMQ) para resolver *on-line* a equação algébrica de Riccati (EAR) discreta. A EAR pode ser vista como uma forma particular da equação HJB associada ao problema do regulador linear quadrático (RLQ) discreto. Segundo [43], a abordagem RMQ é um treinamento eficiente de segunda ordem que leva a uma convergência mais rápida em comparação com as abordagens de primeira ordem, tais como o algoritmo de retro propagação (BP - *backpropagation*) e algoritmo do gradiente descendente, comumente usados em aprendizado de redes neurais [44, 45].

A estrutura de controle que será usada neste trabalho é ilustrada na Figura 1, em que o agente de aprendizagem (controlador) interage com o ambiente (quadricóptero) inicialmente desconhecido. O componente ator aplica uma ação ou política de controle ao ambiente, e o componente crítico avalia o valor daquela ação. Baseado nesta avaliação, vários esquemas podem então ser usados para modificar ou melhorar a ação no sentido que a nova política produz um valor que é melhorado sobre o valor anterior. Desta forma, o agente é capaz de aprender a política ótima a partir de suas experiências sem conhecer os parâmetros do sistema dinâmico.

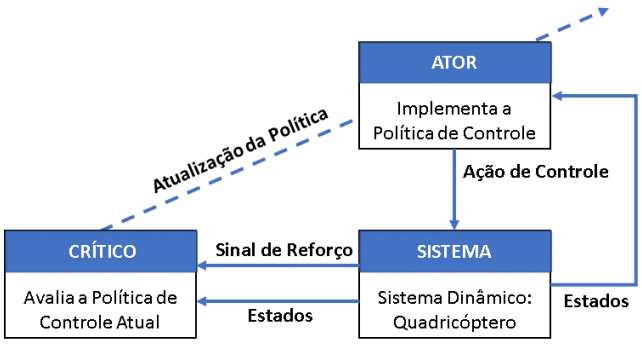

Figura 1. Estrutura de Controle RL Ator-Crítico

O restante deste artigo está organizado como descrito a seguir. A Seção 2 apresenta o modelo matemático de um sistema quadrirrotor. A metodologia e algoritmos de programação dinâmica aproximada que são aplicados ao sistema de quadrirrotor são apresentados na Seção 3. Nesta seção, as formulações são expostas desde a abordagem de Bellman da programação dinâmica clássica até a abordagem que emprega aproximações da solução da equação da otimalidade de Bellman via estrutura crítico-adaptativa. Na Seção 4, são mostrados os resultados de simulação computacional que avaliam o esquema de controle proposto para o sistema de quadrirrotor. E, por fim, a Seção 5, traz a conclusão do trabalho.

2. Descrição Matemática do Sistema Quadrirrotor

Os quadricópteros são caracterizados por possuírem seis graus de liberdade: três graus de liberdade nas posições lineares – translacionais – X, Y e Z, e três nas posições angulares – rotacionais – definidos pelos ângulos do sistema de coordenadas do corpo em relação ao sistema fixo à terra, representado pelos ângulos de Euler ϕ, θ e ψ, que são conhecidos como rolagem (*roll*), arfagem (*pitch*) e guinada (*yaw*), respectivamente, e descrevem a orientação do veículo [9]. Na Figura 2, pode-se visualizar os dois sistemas de coordenadas: *E-frame*, Sistema de Coordenadas Fixo à Terra (*Earth Inertial Reference*), e *B-frame*, Sistema de Coordenadas Fixo ao Corpo (*Body-fixed Reference*). O *B-frame* é definido pela orientação do quadricóptero, no qual os eixos dos rotores apontam na direção positiva de z e os braços nas direções x e y. Este sistema está acoplado ao corpo do veículo e coincide com o centro da estrutura do quadricóptero.

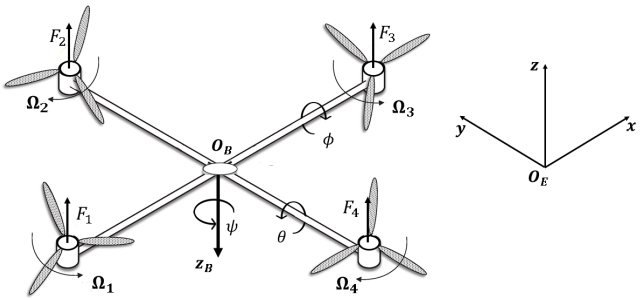

Figura 2. Sistemas de Coordenadas de um quadricóptero

2.1 Descrição Cinemática

A atitude ou posição angular de um quadricóptero é definida pela orientação do sistema *B-frame* em relação ao sistema *E-frame*. O vetor de atitude é definido por

$$\Theta = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}. \tag{1}$$

É possível combinar as quantidades linear e angular dos sistemas para dar uma representação completa do corpo no espaço. Assim, são definidos os seguintes vetores

$$\xi = \begin{bmatrix} \Gamma^E \\ \Theta^E \end{bmatrix} = \begin{bmatrix} X & Y & Z & \phi & \theta & \psi \end{bmatrix}^\intercal \tag{2}$$

$$\nu = \begin{bmatrix} V^B \\ \omega^B \end{bmatrix} = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^\intercal, \tag{3}$$

em que ξ é o vetor de posição generalizada em relação ao sistema *E-frame*, composição dos vetores de posição linear $\Gamma^E = \begin{bmatrix} X & Y & Z \end{bmatrix}^\intercal$ e angular $\Theta^E = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^\intercal$, e ν é o vetor de velocidade generalizada em relação ao sistema *B-frame*, composição dos vetores de velocidade linear $V^B = \begin{bmatrix} u & v & w \end{bmatrix}^\intercal$ e angular $\omega^B = \begin{bmatrix} p & q & r \end{bmatrix}^\intercal$ do quadricóptero.

A Equação (4) descreve a cinemática de um corpo rígido de seis graus de liberdade

$$\dot{\xi} = J_\Theta \nu, \tag{4}$$

em que $\dot{\xi}$ é o vetor de velocidade generalizada em relação ao sistema *E-frame*, e J_Θ é a matriz generalizada, a qual é composta de quatro sub-matrizes de acordo com a seguinte equação

$$J_\Theta = \begin{bmatrix} R_\Theta & 0_{3\times3} \\ 0_{3\times3} & T_\Theta \end{bmatrix}, \tag{5}$$

em que R_Θ é a matriz de rotação e T_Θ é a matriz de transferência dadas por

$$R_\Theta = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi s_\theta c_\phi \\ s_\psi c_\theta & c_\psi c_\phi + s_\psi s_\theta s_\phi & -c_\psi s_\phi + s_\psi s_\theta c_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \tag{6}$$

$$T_\Theta = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix}. \tag{7}$$

Nas Equações (6) e (7), as seguintes notações são adotadas: $c_k = \cos k$, $s_k = \sin k$ e $t_k = \tan k$

2.2 Descrição Dinâmica

As equações de movimentos são mais convenientemente formuladas no sistema fixo ao corpo pelas seguintes razões [46]:

- A matriz de inércia é invariante no tempo;
- A simetria do corpo do veículo pode ser usada para simplificar as equações, com I_{yz} e I_{xy} nulas;
- As forças de controle são geralmente dadas no sistema fixo ao corpo;
- As medições tomadas em voo são facilmente convertidas para o sistema fixo ao corpo.

A dinâmica de um corpo rígido de seis graus de liberdade leva em consideração a massa m e a matriz de inércia I do corpo. A partir do Primeiro Axioma de Euler, da Segunda Lei de Newton seguem as derivações dos componentes lineares do movimento do corpo de acordo com a Equação (8) e dos componentes angulares, extraídos do Segundo Axioma de Euler e da Segunda Lei de Newton, conforme a Equação (9)

$$m(\dot{V}^B + \omega^B \times V^B) = F^B \tag{8}$$

$$I\dot{\omega}^B + \omega^B \times (I\omega^B) = T_\Theta \tau^B. \tag{9}$$

Na forma matricial, a dinâmica é descrita pela seguinte equação

$$\begin{bmatrix} mI_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I \end{bmatrix} \begin{bmatrix} \dot{V}^B \\ \dot{\omega}^B \end{bmatrix} + \begin{bmatrix} \omega^B \times (mV^B) \\ \omega^B \times (I\omega^B) \end{bmatrix} = \begin{bmatrix} F^B \\ \tau^B \end{bmatrix}. \tag{10}$$

A Equação (10) é usada para representar o modelo do quadricóptero. Considera-se que o ponto central do quadricóptero coincide com seu centro de massa e que a sua estrutura em cruz coincide com os eixos principais de inércia. Os vetores F^B e τ^B contém informações que estão divididas em três contribuições: i) vetor gravitacional, gerado a partir da aceleração da gravidade; ii) efeitos giroscópicos, produzidos pela rotação das hélices, e iii) forças e torques produzidos diretamente pelas entradas do movimento principal.

Expressando os efeitos da força gravitacional, efeitos giroscópicos e a matriz de movimento, pode-se representar a dinâmica do quadricóptero na forma de um sistema de equações dado por [47, 48]

$$\begin{aligned} \dot{u} &= (\nu r - wq) + g\, s_\theta \\ \dot{v} &= (wp - ur) - g\, c_\theta s_\phi \\ \dot{w} &= (uq - vp) - g c_\theta s_\phi + \frac{U_1}{m} \\ \dot{p} &= \frac{I_{yy} - I_{zz}}{I_{xx}} qr - \frac{J_{TP}}{I_{xx}} q\Omega + \frac{U_2}{I_{xx}} \\ \dot{q} &= \frac{I_{zz} - I_{xx}}{I_{yy}} pr + \frac{J_{TP}}{I_{yy}} p\Omega + \frac{U_3}{I_{yy}} \\ \dot{r} &= \frac{I_{xx} - I_{yy}}{I_{zz}} pq + \frac{U_4}{I_{zz}}. \end{aligned} \tag{11}$$

No sistema (11), leva-se em consideração que as relações entre as entradas de controle U_1, U_2, U_3 e U_4 e as velocidades das hélices Ω_1, Ω_2, Ω_3 e Ω_4 podem ser aproximadas por

$$\begin{aligned} U_1 &= b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 &= bl(\Omega_4^2 - \Omega_2^2) \\ U_3 &= bl(\Omega_3^2 - \Omega_1^2) \\ U_4 &= d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \\ \Omega &= -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4, \end{aligned} \tag{12}$$

em que b o coeficiente de empuxo, d o coeficiente de arrasto e l é a distância entre os eixos de rotação de duas hélices opostas. J_{TP} é o momento de inércia total em torno do eixo da hélice.

O sistema de Equações (11) descreve a dinâmica do quadricóptero no sistema *B-frame*, mas pode-se representar em um sistema híbrido, composto de equações lineares em relação ao sistema *E-frame* e equações angulares em relação ao sistema *B-frame*. Este novo referencial é chamado *H-frame*. Este sistema é adotado porque é fácil de expressar a dinâmica combinada com o controle (em particular para a posição vertical no referencial inercial da Terra). O seguinte sistema de equações descreve a dinâmica do veículo em relação ao

sistema *H-frame*

$$\begin{aligned}
\ddot{X} &= (\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi)\frac{U_1}{m} \\
\ddot{Y} &= (-\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi)\frac{U_1}{m} \\
\ddot{Z} &= -g + (\cos\theta\cos\phi)\frac{U_1}{m} \\
\dot{p} &= \frac{I_{yy}-I_{zz}}{I_{xx}}qr - \frac{J_{TP}}{I_{xx}}q\Omega + \frac{U_2}{I_{xx}} \\
\dot{q} &= \frac{I_{zz}-I_{xx}}{I_{yy}}pr - \frac{J_{TP}}{I_{yy}}p\Omega + \frac{U_3}{I_{yy}} \\
\dot{r} &= \frac{I_{xx}-I_{yy}}{I_{zz}}pq + \frac{U_4}{I_{zz}}.
\end{aligned} \tag{13}$$

Para projetar um controlador linear, o modelo não linear (13) deve ser linearizado em torno de um ponto de operação. No controle de decolagem e aterrissagem vertical, geralmente, é escolhido para ficar em voo pairado. Se for feita a suposição de que o veículo está em condições de voo pairado, pode-se considerar as seguintes aproximações:

$$\begin{cases}
U_1 \approx mg \\
p \cong q \cong r \cong 0 \\
\sin(\psi) \cong 0 \\
\sin(\theta) \cong \theta \\
\sin(\phi) \cong \phi
\end{cases} \tag{14}$$

Dessa forma, reescreve-se a Equação (13) na seguinte forma:

$$\begin{cases}
\ddot{X} = g\theta \\
\ddot{Y} = -g\phi \\
\ddot{Z} = -g + \dfrac{U_1}{m} \\
\ddot{\phi} = \dot{p} = \dfrac{U_2}{I_{xx}} \\
\ddot{\theta} = \dot{q} = \dfrac{U_3}{I_{yy}} \\
\ddot{\psi} = \dot{r} = \dfrac{U_4}{I_{zz}}
\end{cases} \tag{15}$$

3. Abordagem de Controle

A Programação Dinâmica Aproximada (PDA) consiste em formular métodos para resolver o problema de programação dinâmica "para frente no tempo", em tempo real, com o uso dos dados medidos ao longo das trajetórias do sistema [33]. PDA conduz a uma família de controladores que tem a capacidade de aprenderem *on-line* as soluções de problemas de controle ótimo quando o modelo explícito do sistema não está disponível. Esses métodos são baseados em aprendizagem por diferença temporal e aproximação da função valor. Esta técnica atualiza o valor em cada passo de tempo à medida em que as observações de dados são tomadas ao longo de uma trajetória do sistema. Nesta seção, a solução do Problema do Regulador Linear Quadrático (RLQ) em tempo discreto é apresentada sob a óptica de Programação Dinâmica Aproximada.

3.1 Equações de Bellman para o Sistema de Controle RLQ

Considere o seguinte sistema de tempo discreto

$$x_{k+1} = f(x_k,\, u_k) \tag{16}$$

e

$$u_k = h(x_k), \tag{17}$$

em que a função de transição de estados $f(\cdot):\ \mathbb{R}^n \to \mathbb{R}^n$ e a política de controle $h(\cdot):\ \mathbb{R}^n \to \mathbb{R}^m$ são mapeamentos lineares dados por $f(x_k,\, u_k) = Ax_k + Bu_k$ e $h(x_k) = -Kx_k$, com $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, x_k é o vetor de estados, u_k é o vetor de entradas de controle, e $K(\cdot) \in \mathbb{R}^{m\times n}$ é a matriz de ganhos de realimentação de estados. Uma política $h(\cdot)$ será identificada simplesmente por K.

A ideia de comportamento ótimo orientado à objetivos é extraída da definição de Medida de Desempenho ou Função de Custo, a qual é dada por

$$V^K(x_k) = \sum_{i=k}^{\infty} \gamma^{i-k} r(x_i, u_i), \tag{18}$$

em que $0 < \gamma \leq 1$ um fator de desconto, e r é uma função de utilidade que possui a seguinte estrutura

$$r(x_k, u_k) = x_k^{\intercal} Q x_k + u_k^{\intercal} R u_k, \tag{19}$$

com $Q = Q^{\intercal} \geq 0$ e $R = R^{\intercal} > 0$ matrizes de ponderação do estado e do controle. Uma política de controle $h(x_k) = -Kx_k$ diz-se admissível quando o ganho K é estabilizador e produz um custo finito. Uma forma equivalente à Equação (18) é dada pela equação de diferenças [45]

$$V^K(x_k) = r(x_k, u_k) + \gamma \sum_{i=k+1}^{\infty} \gamma^{i-(k+1)} r(x_i, u_i). \tag{20}$$

$$V^K(x_k) = r(x_k, u_k) + \gamma V^K(x_{k+1}). \tag{21}$$

A Equação (21) é conhecida como equação de Bellman. Ao invés de avaliar a soma infinita (18), pode-se resolver a Equação (21) para obter o valor de usar uma política atual K. Determinar o valor de uma política atual através da equação de Bellman é o conceito-chave no desenvolvimento de técnicas de PDA.

O RLQ tem por objetivo fornecer uma lei de controle que minimiza a função de custo $V^K(x_k)$ para o estado atual x_k. Em RLQ, sabe-se que a função valor de qualquer política de controle admissível K é quadrática no estado atual, ou seja,

$$V^K(x_k) = x_k^T P x_k, \tag{22}$$

para alguma matriz simétrica $P > 0$. Assim, considerando a função de utilidade (19) e a representação (22) da função de custo, a equação de Bellman (21) para o RLQ pode ser escrita na forma

$$x_k^\intercal P x_k = x_k^\intercal Q x_k + u_k^\intercal R u_k + \gamma x_{k+1}^\intercal P x_{k+1}. \quad (23)$$

Em termos da matriz de ganho de realimentação, a Equação (23) é dada na forma

$$x_k^\intercal P x_k = x_k^\intercal \left(Q + K^\intercal R K + \gamma (A - BK)^\intercal P (A - BK)\right) x_k. \quad (24)$$

Uma vez que a Equação (24) deve ser satisfeita para todos os estados atuais x_k, tem-se

$$\gamma (A - BK)^T P (A - BK) - P + Q + K^T R K = 0. \quad (25)$$

Fixado o ganho K, esta equação matricial é linear em P e é conhecida como equação de Lyapunov. Agora, escrevendo a Equação (23) como

$$\begin{aligned} x_k^\intercal P x_k \; &= x_k^\intercal Q x_k + u_k^\intercal R u_k \\ &\quad + \gamma (A x_k + B u_k)^\intercal P (A x_k + B u_k), \end{aligned} \quad (26)$$

a diferenciação com respeito à u_k é aplicada para impor uma política de decisão $u_k = -K x_k$ que minimize a função de custo. Então, tem-se que a política ótima deve satisfazer

$$R u_k + \gamma B^\intercal P (A x_k + B u_k) = 0, \quad (27)$$

ou seja, o ganho de realimentação ótimo é dado por

$$K = \gamma (R + \gamma B^\intercal P B)^{-1} B^\intercal P A. \quad (28)$$

Substituindo a Equação (28) na Equação (26), e após simplificação, tem-se

$$\gamma (A^\intercal P A) - P + Q - \gamma \left[A^\intercal P B \left(\frac{R}{\gamma} + B^\intercal P B \right)^{-1} B^\intercal P A \right] = 0. \quad (29)$$

Esta equação é conhecida como equação algébrica de Riccati (EAR) discreta ou equação da otimalidade de Bellman para o RLQ.

3.2 Iteração de Política para o RLQ

A iteração de política (IP) consiste em um processo iterativo de busca da política de controle ótima K^* em que a cada iteração j, a função valor V_{K_j} para a política atual K_j é determinada (avaliação de política), e, em seguida, uma nova política K_{j+1} é obtida de modo a ser melhor, ou pelo menos igual à política atual (melhoria de política). A avaliação de política é feita através da solução da equação

$$\gamma (A - BK_j)^\intercal P_{j+1} (A - BK_j) - P_{j+1} + Q + K_j^\intercal R K_j = 0. \quad (30)$$

A Equação (30) equivale a equação de Lyapunov (25). Pode-se demonstrar que a equação de Bellman é uma equação do ponto fixo [49]. Assim, dada uma política admissível K_j, existe um único ponto fixo P_{j+1}, e o seguinte mapeamento de contração

$$P_j^{i+1} = \gamma (A - BK_j)^\intercal P_j^i (A - BK_j) + Q + K_j^\intercal R K_j, \quad (31)$$

pode ser iterado a partir de qualquer valor P_j^0 e resulta no limite $P_j{}^i \to P_{j+1}$ quando $i \to \infty$.

A etapa de melhoria de política é feita através da expressão:

$$h_{j+1}(x_k) = -\gamma (R + \gamma B^\intercal P_{j+1} B)^{-1} B^\intercal P_{j+1} A x_k. \quad (32)$$

3.3 Iteração de Política Generalizada para o RLQ

A Iteração de Política Generalizada (IPG) aplica um número M finito de recursões de Lyapunov em cada passo j. O algoritmo é iniciado com a escolha arbitraria de uma política de controle K_0, não necessariamente admissível. A cada passo j do processo iterativo, o valor P_j^i é atualizado através de

$$P_j^{i+1} = \gamma (A - BK_j)^\intercal P_j^i (A - BK_j) + Q + K_j^\intercal R K_j, \quad (33)$$

com $i = 0,\ 1,\ \ldots,\ M-1$, para algum M finito, com $P_j^0 = P_j$ como condição inicial. Em seguida, estabelecendo $P_{j+1} = P_j^M$, realiza-se a etapa de Melhoria de Política, em que se determina uma politica melhorada através de

$$K_{j+1} = \gamma (R + \gamma B^\intercal P_{j+1} B)^{-1} B^\intercal P_{j+1} A. \quad (34)$$

Estabelecer $M = 1$ requer apenas uma recursão de Lyapunov em cada passo j, ou seja, somente uma iteração do mapeamento (33) é tomada no passo de atualização da função valor. Enquanto, estabelecer $M \to \infty$, ou seja, executar o mapeamento (33) até a convergência, resulta no algoritmo IP, que resolve a equação de Lyapunov (25) em cada passo j (a política atual K_j é fixada até que a convergência seja alcançada). Assim, o algoritmo IP apresenta um custo computacional (número de operações aritméticas em ponto flutuante (FLOPs – *floating point operations*) realizadas pelo algoritmo em cada passo j) maior comparado com o caso $M = 1$, no entanto, o algoritmo IP converge em um número de iterações (número de passos j) significativamente menor. Mais detalhes sobre custo computacional e tempo de convergência de algoritmos de iteração de política podem ser encontrados em [50].

3.4 Iteração de Política Aproximada para o RLQ

Iteração de política aproximada (IPA) envolve encontrar uma solução aproximada da equação de Bellman através de uma representação paramétrica da função valor. Em particular, a função valor V^K é uma aproximação paramétrica linear $\hat{V}^K : \mathbb{R}^n \times \mathbb{R}^s \to \mathbb{R}$ dada por

$$\hat{V}^K (x_k, \theta) = x_k^\intercal P x_k = (x_k \otimes x_k)^\intercal vec\,(P) \equiv \bar{x}_k^\intercal \theta, \quad (35)$$

em que $\otimes$ é o produto de Kronecker e $vec(P)$ é o vetor formado pelo empilhamento das colunas da matriz P. Observe

que $\bar{x}_k = x_k \otimes x_k$ é o vetor polinomial quadrático que contém todos os possíveis produtos das n componentes de x_k. Os termos redundantes em $x_k \otimes x_k$ são removidos para definir um conjunto de base quadrática $\bar{x}_k$, ou seja,

$$\bar{x}_k^{\intercal} = \left[x_{k1}^2 \cdots x_{k1}\, x_{kn}\, x_{k2}^2 \cdots x_{k2}\, x_{kn} \cdots x_{k2}\, x_{kn}\, x_{k2}^2\right]. \quad (36)$$

O vetor de parâmetro ajustável é θ, cujas componentes estão em correspondência com os elementos da matriz P. Levando em consideração que P é simétrica, existem somente $n(n+1)/2$ elementos em θ, que são as n entradas diagonais de P e as $n(n+1)/2-n$ somas distintas $p_{ij}+p_{ji}$. Entende-se a forma apresentada na Equação (35) como uma rede neural de ligação funcional (*Function Link Neural Network* - FLNN), que trata-se de uma rede neural cuja resposta produzida é linear nos parâmetros da última camada. Neste caso, tem-se que $\bar{x}_k$ é o vetor de funções de ativação [51, 52].

De acordo com a Equação (23), o parâmetro θ é ajustado para minimizar, em um sentido dos mínimos quadrados, o erro residual (erro de diferença temporal) definido por

$$e_k = x_k^{\intercal} Q x_k + u_k^{\intercal} R u_k + \gamma \bar{x}_{k+1}^{\intercal}\theta - \bar{x}_k^{\intercal}\theta. \quad (37)$$

A função e_k é um erro de predição entre o desempenho observado $x_k^T Q x_k + u_k^T R u_k + \gamma \bar{x}_{k+1}^T \theta$ e o desempenho predito $\bar{x}_k^T \theta$ em resposta a uma ação de controle aplicada ao sistema. Os valores de e_k podem ser considerados como uma indicação de ajuste para satisfazer a equação de Bellman. Isso produz a melhor aproximação para o valor V^{K_j} correspondente ao usar a política atual K_j.

No algoritmo IPA, inicialmente, seleciona-se uma política de controle admissível K_0. Em seguida, avalia-se essa política, através da solução de mínimos quadrados θ_{j+1} para

$$\begin{aligned}(\bar{x}_k - \gamma \bar{x}_{k+1})^{\intercal}\theta_{j+1} &= r\left(x_k, h_j(x_k)\right) \\ &= x_k^{\intercal}\left(Q + K_j^{\intercal} R K_j\right) x_k\end{aligned} \quad (38)$$

e, realiza-se a etapa de melhoria de política por meio da Equação (34). Observe que a estimação da função valor para uma dada política K_j requer somente amostras dos estados e do custo instantâneo r, enquanto que o modelo da dinâmica da planta é necessário para atualização da política de controle. Para tornar a regra de decisão independente do modelo da planta, pode-se usar uma estrutura de rede neural para representar a política de controle. Em [33], os autores propõem um esquema IPA com uma parametrização linear dada por

$$u_k = h(x_k) = U^{\intercal}\sigma(x_k), \quad (39)$$

com $\sigma(x): \mathbb{R}^n \to \mathbb{R}^M$ um vetor de M funções de ativação e $U \in \mathbb{R}^{M\times m}$ uma matriz de pesos ou parâmetros desconhecidos. Isso fornece um algoritmo *on-line* de aprendizagem por reforço para resolver o problema de controle ótimo por meio da Iteração de Política, baseado em dados medidos ao longo das trajetórias do sistema. No presente artigo, usamos a estrutura dada nas Equações (35) e (36) para aproximar a função valor $V(x_k)$, e a política $h(x_k)$ é calculada por meio da Equação (34).

3.4.1 Implementação *On-line* do Método IPA

A solução dos mínimos quadrados para a Equação (38) pode ser resolvida em tempo real coletando-se tuplas de dados da forma $(x_k, x_{k+1}, d(x_k,\theta_j))$, em que $d(x_k,\theta_j)$ é a função objetivo desejada para a qual a estimativa atualizada da função de custo, $\hat{V}\left(x_k,\theta_{j+1}\right)$, precisa corresponder. Tal função consiste do custo imediato (instantâneo) e da estimativa atual da função de custo para o estado seguinte x_{k+1}, ou seja,

$$d(x_k,\theta_j) = x_k{}^{\intercal} Q x_k + u_k{}^{\intercal} R u_k + \gamma \bar{x}_{k+1}^{\intercal}\theta_j. \quad (40)$$

Um aspecto relacionado a esta abordagem é que as estimativas da função de custo associada a uma política K_j são atualizadas a cada passo de tempo observando os dados do sistema. Isso pode ser determinado por meio de simulação, ou, em aplicações em tempo real. Portanto, a estrutura paramétrica da função de custo RLQ tem seu parâmetro θ estimado utilizando o algoritmo recursivo de mínimos quadrados (RMQ). A abordagem RMQ considerada é para viabilizar a solução *on-line* da equação EAR associada ao projeto de controle ótimo RLQ [53]. O algoritmo RMQ é dado por

$$e_j(i) = d(x_k,\theta_j) - \bar{x}_k^{\intercal}\theta_{j+1}(i-1) \quad (41)$$

$$\theta_{j+1}(i) = \theta_{j+1}(i-1) + \frac{\Gamma_j(i-1)\bar{x}_k e_j(i)}{\mu + \bar{x}_k^{\intercal}\Gamma_j(i-1)\bar{x}_k} \quad (42)$$

$$\Gamma_j(i) = \frac{1}{\mu}\left[\Gamma_j(i-1) - \frac{\Gamma_j(i-1)\bar{x}_k\bar{x}_k^{\intercal}\Gamma_j(i-1)}{\mu + \bar{x}_k^{\intercal}\Gamma_j(i-1)\bar{x}_k}\right] \quad (43)$$

em que j é o índice de atualização de política, i é o índice das recursões do RMQ, k é o tempo discreto, Γ é a matriz de covariância da recursão, $e(i)$ é o erro de estimação do RMQ, e μ é o fator de esquecimento, $0 < \mu \leq 1$. Uma forma de interpretar o fator $\mu \leq 1$ é que as observações mais recentes são mais influentes na estimação dos parâmetros, uma vez que elas contêm informação mais atualizada. Se $\mu = 1$, o mesmo peso é atribuído às observações. As componentes do parâmetro θ estão associadas com os elementos da matriz P de Ricatti. Na Seção 3.4, nota-se a quantidade de elementos independentes dessa matriz, que são formados a partir da expressão $n(n+1)/2$, onde n é o número de estados do quadrirrotor ($n = 12$). Então, serão 78 elementos considerados para a avaliação do comportamento de convergência da matriz P de Riccati, no entanto, serão apresentados aqui somente os mais relevantes. Para satisfazer a condição de excitação persistente necessária para a convergência do estimador RMQ, isto é, evitar a singularidade da matriz de covariância do RMQ, usou-se o esquema padrão de reinicialização dos estados a cada intervalo $n_{revit} = n(n+1)/2$. Outros esquemas padrões poderiam ser usados, tais como a reinicialização da matriz de covariância ou a adição de um pequeno sinal de ruído de prova na entrada de controle [54].

4. Resultados de Simulação

Nesta seção são apresentados os resultados de simulação computacional que ilustram o projeto de controle ótimo via métodos de Programação Dinâmica Aproximada para um sistema de quadricóptero, com seis graus de liberdade (6-DOF). Como discutido em [55], o uso de simuladores computacionais é bastante atraente nas pesquisas que envolvem algoritmos de aprendizado, tais como as técnicas de RL, por tornar o desenvolvimento de controladores mais barato, rápido e seguro. Contudo, transferir a política aprendida no simulador para o protótipo real nem sempre é possível, problema este conhecido como *gap* de realidade e é um tópico bastante discutido na comunidade científica e ainda sem solução definitiva. Em [56], os autores mostram as metodologias mais utilizadas para transferir e/ou adaptar as políticas aprendidas no simulador ao mundo real, dentre elas pode-se citar: a utilização de simuladores mais realísticos, o uso de métodos de randomização de domínio e a obtenção de agentes mais robustos a pertubações externas, como realizado neste estudo. Isto posto, uma das questões de interesse na análise dos resultados, é verificar, a partir de uma condição de operação anterior estável, a adaptabilidade do controlador baseado em PDA perante variações nos parâmetros da planta (variação na massa do quadricóptero), durante o movimento de voo pairado do veículo. Uma vez que o controlador ótimo proposto neste artigo é projetado com base em um sistema de tempo discreto, o modelo da planta foi discretizado mediante o segurador de ordem zero [57], com período de amostragem $T_s = 0,001\ s$. Para avaliar a precisão da solução do controlador proposto (política de decisão), são realizadas comparações com a resposta fornecida pelo método IP *off-line* (política de referência). Tal política é usada para mostrar que o algoritmo IPA tem a capacidade de alcançar uma solução suficientemente próxima da solução exata. Os experimentos computacionais foram realizados usando o ambiente MATLAB® (*Matrix Laboratory*) R2018a em um computador pessoal (*Personal Computer* - PC) com CPU Intel Core i7-8550U 1.99 GHz e 16 GB RAM.

A estrutura paramétrica da função de custo RLQ tem seu parâmetro θ estimado por meio do algoritmo recursivo de mínimos quadrados (RMQ). A abordagem RMQ considerada é para viabilizar a solução *on-line* da EAR associada ao projeto de controle ótimo RLQ [58]. As componentes do parâmetro θ estão associadas com os elementos da matriz P de Ricatti. Na Seção 3.4, nota-se a quantidade de elementos independentes dessa matriz, que são formados a partir da expressão $n(n+1)/2$, onde n é o número de estados do veículo $(n = 12)$. Então, serão 78 elementos considerados para a avaliação do comportamento de convergência da matriz P de Riccati, no entanto, serão apresentados aqui somente os mais relevantes. Para satisfazer a condição de excitação persistente necessária para a convergência do estimador RMQ, isto é, evitar a singularidade da matriz de covariância do RMQ, usou-se o esquema padrão de reinicialização dos estados a cada intervalo $n_{revit} = n(n+1)/2$.

4.1 *Setup* da Simulação

O *setup* do algoritmo PDA consiste da informação que está relacionada aos parâmetros do sistema dinâmico (simulador do ambiente), matrizes de ponderação de estado e controle (Q, R), condições iniciais e parâmetros do processo iterativo. Os parâmetros do sistema dinâmico (quadrirrotor) foram escolhidos conforme [59] e são apresentados no Apêndice A.

Os métodos de busca das matrizes Q e R são orientados para impor um objetivo de controle especificado. A classificação desses métodos é baseada em suas características de busca, tais como busca heurística [60, 61], busca baseada em inteligência computacional [62, 63] e busca empírica [64]. Neste artigo, essas matrizes foram definidas pelo método de busca empírica (por tentativa e erro), de acordo com a experiência do usuário. Considerou-se Q e R com variações de testes em que $Q = 10^{q_i} I_{12\times12}$ e $R = 10^{r_i} I_{4\times4}$, onde q_i e r_i assumem valores de peso para avaliar os custos das políticas de controle e desvios dos estados durante o processo iterativo do método IPA ($q_i = -2$ e $r_i = -4$). A escolha adequada das matrizes Q e R tem influência relevante na estabilidade do sistema em malha fechada. Observou-se que os valores de $q_i > 0$, enquanto os valores de r_i foram mantidos constantes, levam à situações de instabilidade. Por outro lado, variações nos valores de q_i, com $q_i < 0$ levam à mapeamentos dos polos em malha fechada na região de estabilidade, ou seja, dentro do círculo unitário de raio igual à 1. Em termos do transitório do sistema de malha fechada, escolheu-se $q_i = -2$, em comparação com $q_i = -1$, que exibiu uma resposta mais oscilatória para algumas variáveis de estado, Z e w. Uma investigação para mapear outras regiões do plano-Z estável envolve o desenvolvimento de heurísticas para a seleção das matrizes Q e R, por exemplo, heurísticas que levam em consideração não somente os elementos diagonais das matrizes de ponderação, mas também elementos fora da diagonal com combinação de diferentes elementos de Q e R [61]. Outro parâmetro importante da função de custo é o fator de desconto. Para garantir que o índice de desempenho dado na Equação (18) seja limitado, um fator de desconto $0 < \gamma < 1$ foi selecionado ($\gamma = 0,5$). O significado de $\gamma < 1$ é que os custos orientados para o futuro são menos relevantes do que os custos incorridos no presente.

Com respeito ao estimador RMQ, considerou-se a condição inicial da matriz de covariância dada de acordo com [65], em que $\Gamma_j(0) = \beta I_{78\times78}$, para alguma constante positiva β com valor grande ($\beta = 10^2$), enquanto o vetor de parâmetros inicial θ_0 pôde ser selecionado aleatoriamente. Assim, as componentes de θ_0 associadas aos elementos diagonais da matriz P de Riccati foram estabelecidas iguais a 10, e as demais componentes iguais a zero. Nas simulações realizadas pôde-se também observar a grande influência do fator de esquecimento μ no comportamento de convergência do parâmetro θ durante o processo de aprendizagem. Na prática o fator μ recebe valores na faixa $0,95 \leq \mu \leq 0,999$ [66]. Nas simulações foram testados valores nesta faixa com incremento de 0,01. Os melhores resultados foram obtidos com $\mu = 0,98$ e $\mu = 0,99$, que levaram à convergência do estimador. Já os valores $\mu = 0,96$

e $\mu = 0,97$ causaram uma divergência abrupta nos valores do parâmetro θ. Nesta seção, os resultados são apresentados para o fator $\mu = 0,99$. Neste projeto, a reinicialização dos estados, x_{revit}, é usada para satisfazer a condição de excitação persistente do estimador RMQ. Para tal, selecionou-se $x_{revit} = \left[0{,}04\ 0{,}04\ 0{,}04\ 0{,}01\ 0{,}01\ 0{,}01\ 0{,}01\ 0{,}01\ 0{,}01\ 0{,}0001\ 0{,}0001\ 0{,}0001\right]^{\mathsf{T}}$.

4.2 Processo Iterativo da Solução *On-line* da EAR

A evolução do processo iterativo para a solução da EAR é dada nas Figuras 3-8 para um ciclo de 60000 iterações ou 60 s. As informações representam o comportamento de convergência dos elementos p_{11}, p_{22}, p_{33}, p_{44}, p_{55} e p_{66} da matriz P, correspondentes aos componentes θ_1, θ_{13}, θ_{24}, θ_{34}, θ_{43} e θ_{51} do vetor de parâmetro θ, respectivamente. Para melhor visualizar, ilustra-se no gráfico superior o comportamento de convergência da iteração 0 a 20000, e no gráfico inferior, tem-se 20000 a 60000. Percebe-se claramente que os elementos convergem para o valor de referência (linha tracejada), com solução admissível em torno de 50000 iterações, ou seja, 50 s.

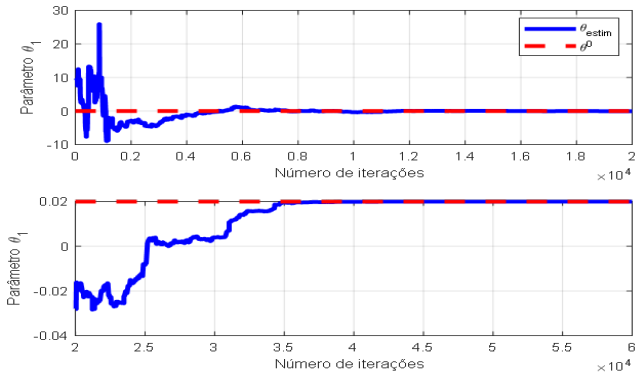

Figura 3. Evolução do processo iterativo do parâmetro θ_1 para um ciclo de 60000 iterações.

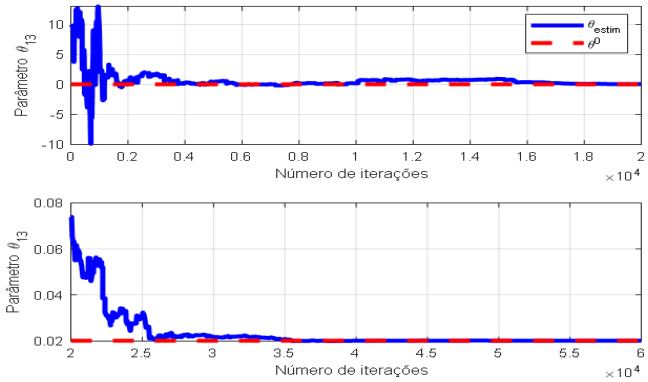

Figura 4. Evolução do processo iterativo do parâmetro θ_{13} para um ciclo de 60000 iterações.

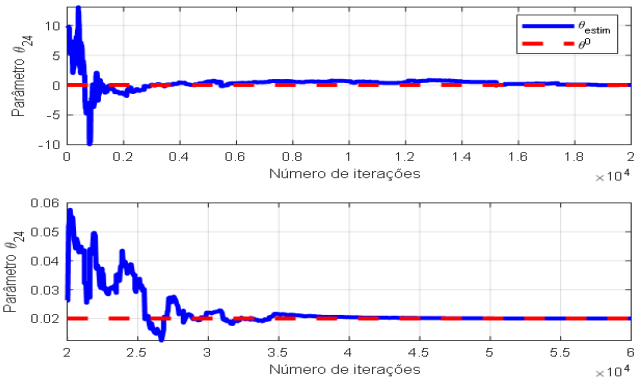

Figura 5. Evolução do processo iterativo do parâmetro θ_{24} para um ciclo de 60000 iterações.

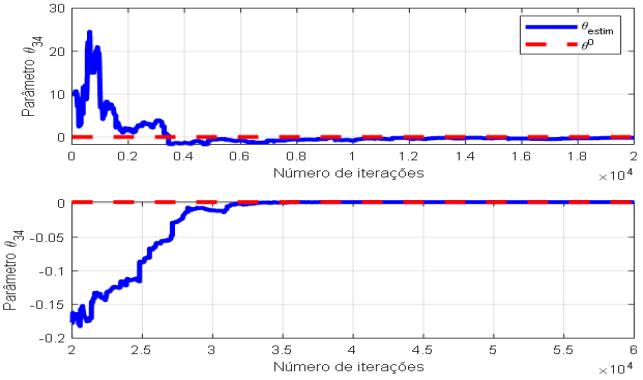

Figura 6. Evolução do processo iterativo do parâmetro θ_{34} para um ciclo de 60000 iterações.

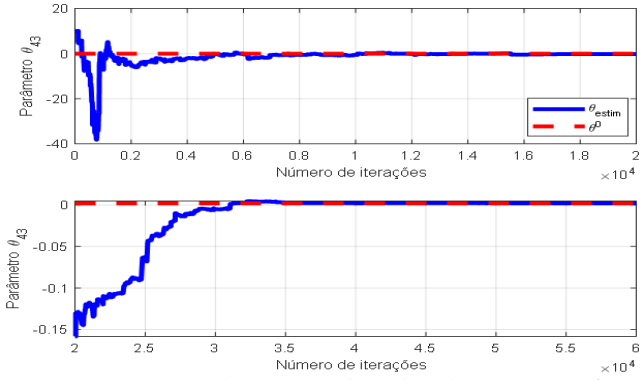

Figura 7. Evolução do processo iterativo do parâmetro θ_{43} para um ciclo de 60000 iterações.

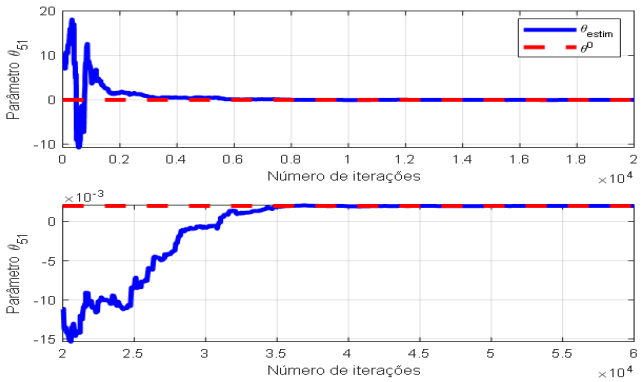

Figura 8. Evolução do processo iterativo do parâmetro θ_{51} para um ciclo de 60000 iterações.

Para efeito de análise do desempenho do controlador baseado em PDA, mostra-se na Figura 9 o comportamento da função de custo durante o processo de aprendizagem. Tal função avalia a qualidade das políticas de controle ao longo do processo iterativo da EAR.

Figura 9. Função de custo para um ciclo de 60000 iterações.

4.3 Matriz de Ganhos *K*

Mostra-se a seguir a evolução dos elementos da matriz de ganhos K durante o processo de aprendizagem, oriunda da política de controle $u = -Kx$, na qual dos 48 elementos que compõem essa matriz, somente os mais relevantes serão apresentados. Observa-se nas Figuras 10 e 11 que a estabilidade dos elementos da matriz K é alcançada, em geral, depois de 50000 iterações. O objetivo de controle, que é estabilizar o quadricóptero na condição de voo pairado (ponto de equilíbrio nulo), é alcançado.

As Figuras 12 e 13 mostram as trajetórias representativas seguidas pelos estados após o processo de aprendizagem da política de controle ótima. Percebe-se claramente que os estados alcançaram a condição de equilíbrio (estado nulo).

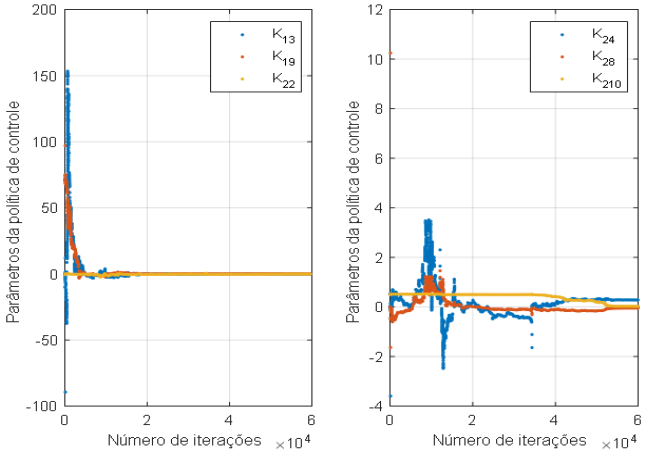

Figura 10. Comportamento de convergência dos elementos K_{ij} ($i = 1,\ 2$ e $j = 2,\ 3,\ 4,\ 8,\ 9,\ 10$) da matriz de ganhos K durante o processo de aprendizagem para um ciclo de 60000 iterações.

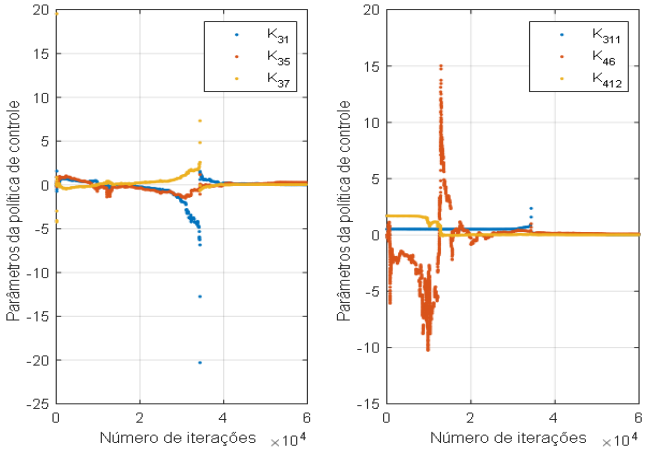

Figura 11. Comportamento de convergência dos elementos K_{ij} ($i = 3,\ 4$ e $j = 1,\ 5,\ 6,\ 7,\ 11,\ 12$) da matriz de ganhos K durante o processo de aprendizagem para um ciclo de 60000 iterações.

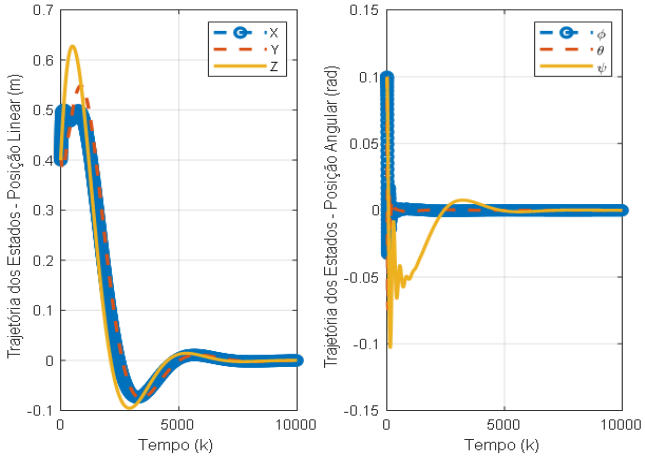

Figura 12. Trajetória dos estados $x_1 = X$, $x_2 = Y$, $x_3 = Z$, $x_4 = \phi$, $x_5 = \theta$ e $x_6 = \psi$ para um período de 10 s ou 10000 iterações.

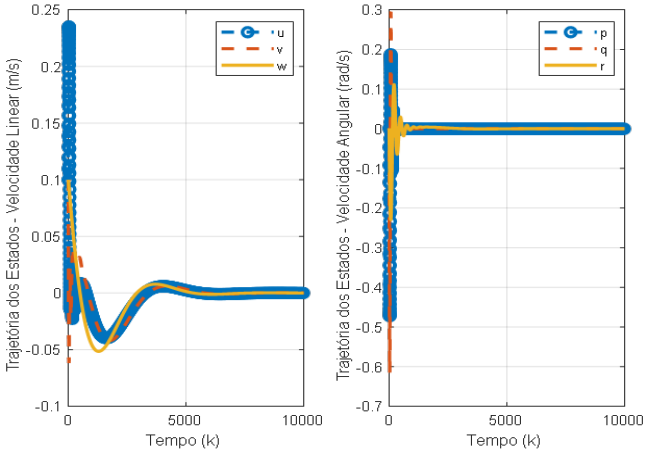

Figura 13. Trajetória dos estados $x_7 = u$, $x_8 = v$, $x_9 = w$, $x_{10} = p$, $x_{11} = q$ e $x_{12} = r$ para um período de 10 s ou 10000 iterações.

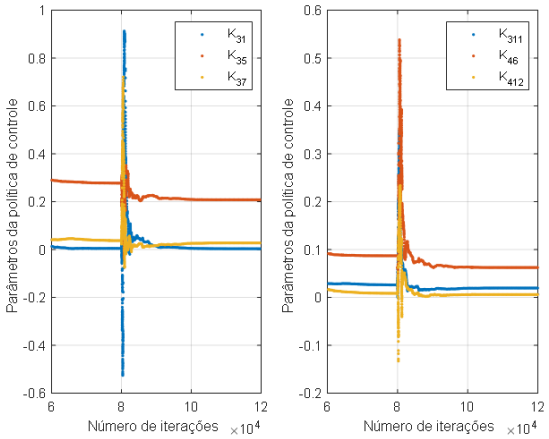

Figura 15. Comportamento de convergência dos elementos K_{ij} ($i = 3,\ 4$ e $j = 1,\ 5,\ 6,\ 7,\ 11,\ 12$) da matriz de ganhos K durante o processo de aprendizagem para um ciclo de 120000 iterações.

4.4 Testes com Variações nos Parâmetros do Quadrirrotor

Para demonstrar a adaptabilidade do controlador RLQ baseado em PDA, foram realizados testes com variações bruscas na massa do quadrirrotor, regidas por uma função degrau. Em particular, as Figuras 14 e 15 ilustram a resposta do controlador devido a um acréscimo de 30% no valor da massa do quadrirrotor (o que corresponde à uma variação de 0,8 kg para 1,04 kg), que ocorreu na iteração 80000. Pode-se observar uma mudança drástica nos parâmetros do controlador nas primeiras iterações onde é inserida a perturbação, no entanto, o controlador é capaz de ajustar automaticamente seus parâmetros para levar o quadrirrotor para o mesmo ponto de operação estável (estado nulo), ou seja, a política de controle atua levando o sistema novamente a estabilizar-se para a condição de voo pairado. As trajetórias dos estados com os novos ganhos estabilizadores são mostradas nas Figuras 16 e 17.

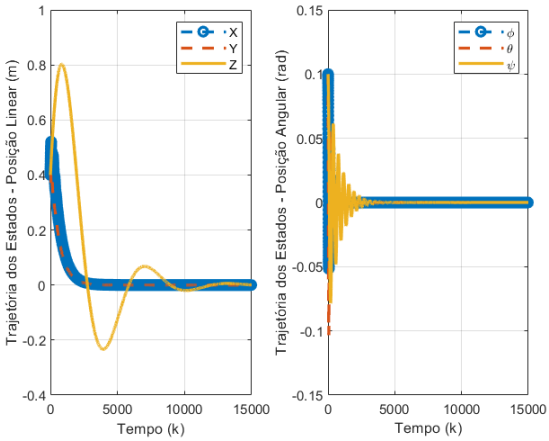

Figura 16. Trajetória dos estados $x_1 = X$, $x_2 = Y$, $x_3 = Z$, $x_4 = \phi$, $x_5 = \theta$ e $x_6 = \psi$ para um período de 15 s ou 15000 iterações.

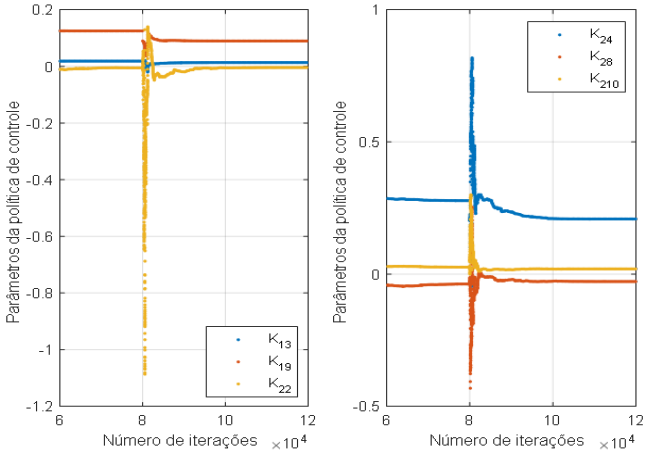

Figura 14. Comportamento de convergência dos elementos K_{ij} ($i = 1,\ 2$ e $j = 2,\ 3,\ 4,\ 8,\ 9,\ 10$) da matriz de ganhos K durante o processo de aprendizagem para um ciclo de 120000 iterações.

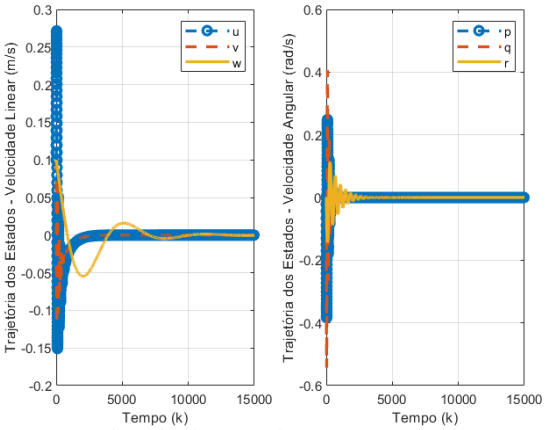

Figura 17. Trajetória dos estados $x_7 = u$, $x_8 = v$, $x_9 = w$, $x_{10} = p$, $x_{11} = q$ e $x_{12} = r$ para um período de 15 s ou 15000 iterações.

5. Conclusão

Este trabalho objetivou o desenvolvimento de uma metodologia e um algoritmo de programação dinâmica aproximada baseado em esquemas de iteração de política e estimação RMQ para o projeto de um controlador ótimo adaptativo de um sistema de helicóptero quadrirrotor. As habilidades do controlador proposto foram avaliadas em um modelo do quadrirrotor simulado no ambiente MATLAB®. O desempenho do método IPA baseado em estimação RMQ mostrou que em termos de acurácia da estimação do parâmetro P associado à solução da EAR, os valores estimados se ajustaram adequadamente ao valor verdadeiro de P em regime permanente. Para testar a adaptabilidade do controlador proposto, as variações nos parâmetros do quadrirrotor foram modeladas para representar situações que podem ocorrer durante a operação desse sistema, tais como o acoplamento repentino de massa e/ou retirada repentina de massa acoplada ao quadrirrotor. Os resultados de simulação apresentados mostraram que os estados do sistema foram regulados rapidamente para o ponto de equilíbrio estável (condição de voo pairado) após a variação na massa do quadrirrotor, demonstrando que o controlador RLQ baseado em PDA foi capaz de ajustar seus ganhos e adaptar-se à nova configuração da planta. Como escopo de trabalho futuro deste estudo de simulação, sugere-se uma investigação sobre a metodologia de implementação, conforme abordado na Seção 4, do controlador de estabilização discutido nesta pesquisa em um protótipo real para a condição de voo pairado de um sistema quadrirrotor.

6. Contribuição dos Autores

Todos os autores contribuiram igualmente para este artigo.

Referências

[1] ALABAZARES, D. L. et al. Quadrotor UAV attitude stabilization using fuzzy robust control. *Transactions of the Institute of Measurement and Control*, v. 43, n. 12, p. 2599–2614, abril 2021.

[2] NOORDIN, A.; BASRI, M. A. M.; MOHAMED, Z. Sliding mode control for altitude and attitude stabilization of quadrotor UAV with external disturbance. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, v. 7, n. 2, p. 203–210, junho 2019.

[3] OKYERE, E. et al. LQR controller design for quad-rotor helicopters. *The Journal of Engineering*, v. 2019, n. 17, p. 4003–4007, junho 2019.

[4] NOORMOHAMMADI-ASL, A. et al. System identification and H_∞-based control of quadrotor attitude. *Mechanical Systems and Signal Processing*, v. 135, p. 1–16, janeiro 2020.

[5] FARZANEH, M.; TAVAKOLPOUR-SALEH, A. Stabilization of a quadrotor system using an optimal neural network controller. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, v. 44, p. 1–12, dezembro 2021.

[6] NOORDIN, A. et al. Adaptive PID controller using sliding mode control approaches for quadrotor UAV attitude and position stabilization. *Arabian Journal for Science and Engineering*, v. 46, p. 963–981, julho 2021.

[7] LIMA, G. V. et al. Stabilization and path tracking of a mini quadrotor helicopter: Experimental results. *IEEE Latin America Transactions*, v. 17, n. 3, p. 485–492, março 2019.

[8] ABDELMAKSOUD, S. I.; MAILAH, M.; ABDALLAH, A. M. Control strategies and novel techniques for autonomous rotorcraft unmanned aerial vehicles: A review. *IEEE Access*, v. 8, p. 195142–195169, outubro 2020.

[9] IDRISSI, M.; SALAMI, M.; ANNAZ, F. A Review of Quadrotor Unmanned Aerial Vehicles: Applications, Architectural Design and Control Algorithms. *Journal of Intelligent and Robotic Systems*, v. 104, n. 22, p. 1–33, janeiro 2022.

[10] QUAN, Q. *Introduction to Multicopter Design and Control*. 1. ed. Singapore: Springer, 2017.

[11] TALEB, A. Y. A.; EL-SAYED, H. S.; YASSIN, H. A. Analysis of a practical quad copter robot using linear quadratic regulator controller. *Engineering Research Journal (ERJ)*, v. 39, n. 2, p. 89–98, abril 2016.

[12] SAFAEI, A.; MAHYUDDIN, M. N. Lyapunov-based nonlinear controller for quadrotor position and attitude tracking with GA optimization. In: *2016 IEEE Industrial Electronics and Applications Conference (IEACon)*. Kota Kinabalu, Sabá, Malásia: IEEE, 2016. p. 342–347.

[13] CURI, S.; MAS, I.; PENA, R. S. Autonomous flight of a commercial quadrotor. *IEEE Latin America Transactions*, v. 12, n. 5, p. 853–858, agosto 2014.

[14] JAGGI, N.; MUKHERJEE, K.; KHANRA, M. Design and test of a controller for the Pluto 1.2 quadcopter. *IFAC-PapersOnLine*, v. 55, n. 1, p. 752–757, fevereiro 2022.

[15] LOZANO, Y.; GUTIéRREZ, O. Design and control of a four-rotary-wing aircraft. *IEEE Latin America Transactions*, v. 14, n. 11, p. 4433–4438, novembro 2016.

[16] AHMAD, F. et al. Simulation of the quadcopter dynamics with LQR based control. *Materials Today: Proceedings*, v. 24, p. 326–332, 2020.

[17] THU, K. M.; GAVRILOV, A. Designing and modeling of quadcopter control system using L1 adaptive control. *Procedia Computer Science*, v. 103, p. 528–535, março 2017.

[18] PéREZ, R. I. et al. Attitude control of a quadcopter using adaptive control technique. In: *Adaptive Robust Control Systems*. [S.l.]: IntechOpen, 2017. cap. 6. DOI: <10.5772/intechopen.71382>.

[19] ROTHE, J. et al. A modified model reference adaptive controller (m-mrac) using an updated mit-rule for the altitude of a uav. *Electronics*, v. 9, p. 1–15, julho 2020.

[20] LI, C.; WANG, Y.; YANG, X. Adaptive fuzzy control of a quadrotor using disturbance observer. *Aerospace Science and Technology*, v. 128, p. 1–11, julho 2022.

[21] JIN, X.-Z. et al. Robust adaptive neural network-based compensation control of a class of quadrotor aircrafts. *Journal of the Franklin Institute*, v. 357, n. 17, p. 12241–12263, novembro 2020.

[22] HUANG, J.-W. et al. Demonstration of a model-free backstepping control on a 2-DOF laboratory helicopter. *International Journal of Dynamics and Control*, v. 9, p. 97–108, junho 2020.

[23] ALMAKHLES, D. J. Robust backstepping sliding mode control for a quadrotor trajectory tracking application. *IEEE Access*, v. 8, p. 5515–5525, janeiro 2020.

[24] HE, D. et al. Real-time visual feedback control of multi-camera UAV. *Journal of Robotics and Mechatronics*, v. 33, n. 2, p. 263–273, abril 2021.

[25] URBAńSKI, K. Low altitude control for quadcopter using visual feedback. *Archives of Electrical Engineering*, v. 70, n. 4, p. 845–858, maio 2021.

[26] DESHPANDE, A. M.; MINAI, A. A.; KUMAR, M. Robust deep reinforcement learning for quadcopter control. *IFAC-PapersOnLine*, v. 54, n. 20, p. 90–95, outubro 2021.

[27] PI, C.-H. et al. Low-level autonomous control and tracking of quadrotor using reinforcement learning. *Control Engineering Practice*, v. 95, p. 1–11, fevereiro 2020.

[28] KOCH, W. et al. Reinforcement learning for UAV attitude control. *ACM Trans. Cyber-Phys. Syst.*, Association for Computing Machinery, v. 3, n. 2, p. 1–21, fevereiro 2019.

[29] STINGU, E.; LEWIS, F. L. An approximate dynamic programming based controller for an underactuated 6DoF quadrotor. *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, p. 271–278, 2011.

[30] DEEPAK, B.; SINGH, P. A survey on design and development of an unmanned aerial vehicle (quadcopter). *International Journal of Intelligent Unmanned Systems*, v. 4, p. 70–106, abril 2016.

[31] BERTSEKAS, D. P. *Dynamic Programming and Optimal Control*. 4. ed. Cambridge: Athena Scientific, 2012. II.

[32] WERBOS, P. Reinforcement learning and approximate dynamic programming (RLADP)-foundations, common misconceptions, and the challenges ahead. In: ____. [S.l.]: Wiley-IEEE Press, 2013. cap. 1, p. 1–30.

[33] LEWIS, F. L.; VRABIE, D.; VAMVOUDAKIS, K. G. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems Magazine*, v. 32, n. 6, p. 76–105, dezembro 2012.

[34] KIUMARSI, B. et al. Optimal tracking control of unknown discrete-time linear systems using input-output measured data. *IEEE Transactions on Cybernetics*, v. 45, n. 12, p. 2770–2779, dezembro 2015.

[35] VRABIE, D.; VAMVOUDAKIS, K. G.; LEWIS, F. L. *Optimal Adaptive Control and Differential Games by Reinforcement Learning Principles*. London: The Institution of Engineering and Technology, 2013.

[36] KHAN, S. G. et al. Reinforcement learning and optimal adaptive control: An overview and implementation examples. *Annual Reviews in Control*, v. 36, n. 1, p. 42–59, abril 2012.

[37] LIU, D. et al. Adaptive dynamic programming for control: A survey and recent advances. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, v. 51, n. 1, p. 142–160, janeiro 2021.

[38] FU, Z. et al. Online adaptive optimal control of vehicle active suspension systems using single-network approximate dynamic programming. *Mathematical Problems in Engineering*, v. 2017, p. 1–9, abril 2017.

[39] MAITI, R.; SHARMA, K. D.; SARKAR, G. PSO based parameter estimation and PID controller tuning for 2-DOF nonlinear twin rotor MIMO system. *International Journal of Automation and Control*, v. 12, p. 582–609, janeiro 2018.

[40] CHOUDHARY, S. K. Optimal feedback control of twin rotor MIMO system with a prescribed degree of stability. *International Journal of Intelligent Unmanned Systems*, v. 4, p. 226–238, outubro 2016.

[41] SáNCHEZ, O. S. et al. Optimized discrete control law for quadrotor stabilization: Experimental results. *Journal of Intelligent and Robotic Systems*, v. 84, p. 67–81, dezembro 2016.

[42] KALLIES, C.; IBRAHIM, M.; FINDEISEN, R. Approximated constrained optimal control subject to variable parameters. *IFAC-PapersOnLine*, v. 53, n. 2, p. 9310–9315, 2020.

[43] GOVINDHASAMY, J. et al. Reinforcement learning for online control and optimisation. *IEE Control Engineering Book Series*, Institution of Engineering and Technology, v. 70, n. 9, p. 293–326, 2005.

[44] AL-DABOONI, S.; WUNSCH, D. The boundedness conditions for model-free HDP(λ). *IEEE Transactions on Neural Networks and Learning Systems*, v. 30, n. 7, p. 1928–1942, julho 2019.

[45] GUO, W. et al. Online adaptation of controller parameters based on approximate dynamic programming. In: *2014 International Joint Conference on Neural Networks (IJCNN)*. Pequim, China: IEEE, 2014. p. 256–262.

[46] VARGAS, F. J. T.; PAGLIONE, P. *Ferramentas de Álgebra Computacional: Aplicações em Modelagem, Simulação e Controle para Engenharia*. 1. ed. São Paulo: LTC, 2015.

[47] STENGEL, R. F. *Flight Dynamics*. Princeton: Princeton University Press, 2004.

[48] BEARD, R. W.; BEARD, T. W. M. *Small unmanned aircraft: theory and practice*. 2. ed. Princeton: Princeton University Press, 2012.

[49] LEWIS, F. L.; VRABIE, D. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits and Systems Magazine*, v. 9, n. 3, p. 32–50, agosto 2009.

[50] BUSONIU, L. et al. *Reinforcement Learning and Dynamic Programming Using Function Approximators*. 1st. ed. Boca Raton, Florida, USA: CRC Press, Inc., 2010.

[51] LEWIS, F. L.; YESILDIRAK, A.; JAGANNATHAN, S. *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Philadelphia, USA: Taylor and amp; Francis, Inc., 1998.

[52] WANG, C. et al. Optimal critic learning for robot control in time-varying environments. *IEEE Transactions on Neural Networks and Learning Systems*, v. 26, n. 10, p. 2301–2310, outubro 2015.

[53] RêGO, P.; FONSECANETO, J.; FERREIRA, E. Convergence of the standard RLS method and UDU$^{\intercal}$ factorisation of covariance matrix for solving the algebraic riccati equation of the DLQR via heuristic approximate dynamic programming. *International Journal of Systems Science*, v. 46, p. 1–23, dezembro 2013.

[54] AL-TAMIMI, A.; ABU-KHALAF, M.; LEWIS, F. L. Adaptive critic designs for discrete-time zero-sum games with application to H_∞ control. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, v. 37, n. 1, p. 240–247, fevereiro 2007.

[55] IBARZ, J. et al. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, v. 40, n. 4-5, p. 698–721, janeiro 2021.

[56] ZHAO, W.; QUERALTA, J. P.; WESTERLUND, T. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In: *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. Canberra, ACT, Austrália: IEEE, 2020. p. 737–744.

[57] DORF, R.; BISHOP, R. *Modern Control Systems*. 13. ed. Hoboken, New Jersey, USA: Pearson, 2017.

[58] FERREIRA, E. F.; RêGO, P. H.; NETO, J. V. Numerical stability improvements of state-value function approximations based on RLS learning for online HDP-DLQR control system design. *Engineering Applications of Artificial Intelligence*, v. 63, p. 1–19, agosto 2017.

[59] ARAAR, O.; AOUF, N. Full linear control of a quadrotor UAV, LQ vs H_∞. In: *2014 UKACC International Conference on Control (CONTROL)*. Loughborough, Reino Unido: IEEE, 2014. p. 133–138.

[60] BRYSON, A.; HO, Y. *Applied Optimal Control: Optimization, Estimation, and Control*. New York: Taylor & Francis Group, 1975.

[61] NETO, J.; RêGO, P. QR-tuning and approximate-ls solutions of the HJB equation for online DLQR design via state and action-dependent heuristic dynamic programming. *International Journal of Innovative Computing, Information and Control*, v. 10, p. 1071–1094, janeiro 2014.

[62] JOELIANTO, E.; CHRISTIAN, D.; SAMSI, A. Swarm control of an unmanned quadrotor model with LQR weighting matrix optimization using genetic algorithm. *Journal of Mechatronics, Electrical Power, and Vehicular Technology*, v. 11, p. 1–10, julho 2020.

[63] ASSAHUBULKAHFI, M. et al. LQR tuning by particle swarm optimization of full car suspension system. *International Journal of Engineering and Technology*, v. 7, p. 328–331, maio 2018.

[64] KUANTAMA, E.; TARCA, I.; TARCA, R. Feedback linearization LQR control for quadcopter position tracking. In: *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*. Thessaloniki, Grécia: IEEE, 2018. p. 204–209.

[65] ÅSTRÖM, K.; WITTENMARK, B. *Adaptive Control*. 2. ed. Lund, Sweden: Addison-Wesley, 1995.

[66] SUN, X. et al. Adaptive forgetting factor recursive least square algorithm for online identification of equivalent circuit model parameters of a lithium-ion battery. *Energies*, v. 12, p. 1–15, junho 2019.

APÊNDICE A: Parâmetros da Simulação

Nesta seção estão descritos os valores dos parâmetros utilizados na simulação do projeto de controle.

Tabela 1. Table of Grades

Parâmetro	Valor	Unidade
Massa (m)	$0,8$	kg
Distância do centro do quadricóptero ao rotor (l)	$0,3$	m
Fator de arrasto (d)	7×10^{-5}	Nms^2
Fator de empuxo (b)	2×10^{-4}	Ns^2
Momento de Inércia no eixo x (I_{xx})	$5,17 \times 10^{-3}$	kgm^2
Momento de Inércia no eixo y (I_{yy})	$5,17 \times 10^{-3}$	kgm^2
Momento de Inércia no eixo z (I_{zz})	$1,7 \times 10^{-2}$	kgm^2
Força da gravidade (g)	$9,81$	m/s^2

APÊNDICE B: Modelo em Espaço de Estados

Considera-se o vetor de estado $x = [X\ Y\ Z\ \phi\ \theta\ \psi\ \dot{X}\ \dot{Y}\ \dot{Z}\ \dot{\phi}\ \dot{\theta}\ \dot{\psi}]^\intercal$. Assim, a partir da Equação (15), tem-se a seguinte representação em espaço de estado:

$$\dot{x} = Ax + Bu, \tag{B.1}$$

em que

$$A = \begin{bmatrix} 0_{3\times3} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} & I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times1} & 0_{3\times1} & 0_{3\times1} & 0_{3\times3} & I_{3\times3} \\ 0_{1\times3} & 0 & g & 0 & 0_{1\times3} & 0_{1\times3} \\ 0_{1\times3} & -g & 0 & 0 & 0_{1\times3} & 0_{1\times3} \\ 0_{4\times3} & 0_{4\times1} & 0_{4\times1} & 0_{4\times1} & 0_{4\times3} & 0_{4\times3} \end{bmatrix} \tag{B.2}$$

e

$$B = \begin{bmatrix} 0_{8\times1} & 0_{8\times1} & 0_{8\times1} & 0_{8\times1} \\ 1/m & 0 & 0 & 0 \\ 0 & 1/I_{XX} & 0 & 0 \\ 0 & 0 & 1/I_{YY} & 0 \\ 0 & 0 & 0 & 1/I_{ZZ} \end{bmatrix} \tag{B.3}$$

As matrizes do sistema discretizado são obtidas através do método segurador de ordem zero (*zero-order hold*) com o tempo de amostragem $T_s = 0,001\ s$.