BOCA
BOLETIM DE CONJUNTURA

UM CONTROLADOR ADAPTATIVO ÓTIMO BASEADO EM APRENDIZADO ONLINE ATOR-CRÍTICO PARA UM MANIPULADOR ROBÓTICO

AN OPTIMAL ADAPTIVE CONTROLLER BASED ON ONLINE ACTOR-CRITIC LEARNING FOR A ROBOT MANIPULATOR

UN CONTROLADOR ADAPTATIVO ÓPTIMO BASADO EN EL APRENDIZAJE ACTOR-CRÍTICO EN LÍNEA PARA UN MANIPULADOR ROBÓTICO

10.56238/bocav25n74-031

Patrícia Helena Moraes Rêgo
Doutora em Engenharia Elétrica
Instituição: Universidade Estadual do Maranhão (UEMA)
E-mail: phmrego@yahoo.com.br
Orcid: https://orcid.org/0000-0001-5899-0623
Lattes: http://lattes.cnpq.br/6535271381344851

Joelson Miller Bezerra de Sousa
Mestre em Engenharia da Computação
Instituição: Universidade Estadual do Maranhão (UEMA)
E-mail: joelsonmiller@hotmail.com
Lattes: http://lattes.cnpq.br/6867462270921352

RESUMO
As incertezas nos parâmetros de um manipulador robótico podem afetar, de forma significativa, o desempenho do manipulador, ocasionando erros de regime e de seguimento de trajetória. Controladores adaptativos apresentam-se como uma boa alternativa para esses sistemas, pois possuem como principal característica a capacidade de aprenderem online usando estimação de parâmetros em tempo real. No entanto, controladores adaptativos não são geralmente projetados com a qualidade de serem ótimos com respeito aos critérios de desempenho especificados e, desta forma, não são viáveis para aplicações onde o uso ótimo de recursos é altamente desejável, como por exemplo em robôs humanoides e robôs de serviços. Este artigo apresenta o projeto e investigação de desempenho de um controlador que combina características de controle adaptativo e controle ótimo para um manipulador robótico. Especificamente, o esquema de controle proposto é implementado como uma estrutura ator-crítico, a qual está inserida no contexto de aprendizado por reforço, caracterizando este projeto como uma abordagem independente do modelo da planta. Em contraste a outros sistemas ator-críticos em que são usadas duas redes neurais independentes, uma para aproximar a função valor, e a outra para aprender ações de controle, neste esquema, se define uma única rede neural, o que reduz o número de parâmetros a serem estimados. Os resultados de simulação demonstram o desempenho desejado do controlador proposto que atua em um manipulador de juntas rotativas com dois graus de liberdade.

Palavras-chave: Manipulador Robótico. Controle Adaptativo. Controle Ótimo. Aprendizado por Reforço. Esquema Ator-Crítico.

ABSTRACT
The uncertainties in the parameters of a robot manipulator can significantly affect the robot performance, causing steady-state and trajectory following errors. Adaptive controllers are a good alternative for these systems, since their main feature is the capability to learn online using real-time parameter estimation. Nevertheless, adaptive controllers are not usually designed to be optimal to a prescribed performance index, and thus, are not suitable to applications in which optimal use of resources is highly desirable, for instance humanoid and service robots. This paper presents the design and performance study of a controller that combine features of adaptive control and optimal control applied to a robot manipulator. Specifically, the proposed control scheme is implemented as an actor-critic structure, which is in the reinforcement learning context, characterizing this design as a model-free approach. In contrast to others actor-critic systems in which two independent neural networks are used, one for approximating the value function and another for learning the control actions, in this scheme, a single neural network is defined, reducing the number of parameters to be estimated. The simulation results validate the desired performance of the proposed controller applied in a two-link robot manipulator with revolute joints.

Keywords: Robot Manipulator. Adaptive Control. Optimal Control. Reinforcement Learning. Actor-Critic Scheme.

RESUMEN
Las incertidumbres en los parámetros de un manipulador robótico pueden afectar significativamente al rendimiento del manipulador, provocando errores de régimen y de seguimiento de la trayectoria. Los controladores adaptativos se presentan como una buena alternativa para estos sistemas, ya que su principal característica es la capacidad de aprender en línea utilizando la estimación de parámetros en tiempo real. Sin embargo, los controladores adaptativos no suelen diseñarse con la calidad de ser óptimos con respecto a los criterios de rendimiento especificados y, por lo tanto, no son viables para aplicaciones en las que es muy deseable el uso óptimo de los recursos, como por ejemplo en robots humanoides y robots de servicio. Este artículo presenta el diseño y la investigación del rendimiento de un controlador que combina características de control adaptativo y control óptimo para un manipulador robótico. En concreto, el esquema de control propuesto se implementa como una estructura actor-crítico, que se inserta en el contexto del aprendizaje por refuerzo, lo que caracteriza a este diseño como un enfoque independiente del modelo de la planta. A diferencia de otros sistemas actor-crítico en los que se utilizan dos redes neuronales independientes, una para aproximar la función de valor y otra para aprender acciones de control, en este esquema se define una única red neuronal, lo que reduce el número de parámetros que deben estimarse. Los resultados de la simulación demuestran el rendimiento deseado del controlador propuesto, que actúa en un manipulador de juntas rotativas con dos grados de libertad.

Palabras clave: Manipulador Robótico. Control Adaptativo. Control Óptimo. Aprendizaje por Refuerzo. Esquema Actor-Crítico.

1 INTRODUÇÃO

O desenvolvimento de estratégias de controle para manipuladores robóticos apresenta dificuldades decorrentes das próprias características do sistema, isto é, um robô articulador é um sistema dinâmico multivariável, com fortes não-linearidades devidas aos acoplamentos de suas juntas e movimentos, além de apresentar parâmetros incertos ou que variam no tempo, tais como a massa e inércia dos elos, atritos ou folgas nas engrenagens das juntas, variações nas cargas de trabalho, localização do centro de massa (que pode mudar quando o robô estiver com carga), entre outras (Fateh; Fateh, 2019). Estas imprecisões paramétricas resultam em perdas de exatidão e velocidade nos movimentos do manipulador, que em determinadas aplicações é altamente indesejável. Já a dinâmica não-linear pode levar o sistema à instabilidade em determinados pontos de operação (Craig, 2021).

Controladores convencionais de realimentação, tal como o PID (Proporcional-Integral-Derivativo), são vastamente utilizados na indústria por serem simples, fáceis de implementar e por apresentarem bom desempenho em diversas aplicações (Borase *et al*., 2021). Entretanto, este esquema de controle, por ser um tipo de controle com ganhos fixos, torna-se insuficiente quando aplicado a sistemas com não linearidades e/ou incertezas (parâmetros imprecisos, dinâmicas não-modeladas de alta frequência e perturbações), ou seja, sistemas que apresentam pontos de operação variáveis (Konstantopoulos; Baldivieso-Monasterios, 2020).

Dentre os controladores clássicos aplicados a manipuladores existem aqueles baseados em modelo (cinemático e/ou dinâmico para controle de posição, velocidade e força). Porém, estas abordagens necessitam do conhecimento completo das equações que descrevem o comportamento do sistema, sendo elas bastante complexas e com parâmetros que muitas vezes são incertos. A complexidade do modelo cresce também com o aumento de juntas e elos do manipulador, aumentando o custo computacional para solucionar estas equações (Moosavi; Zafar; Sanfilippo, 2022).

A teoria de controle adaptativo fornece meios para desenvolver soluções para sistemas dinâmicos que demandam controladores mais complexos. Esta abordagem permite compensar, de forma *online*¸ as variações e incertezas paramétricas do sistema garantindo que os critérios de desempenho desejados sejam alcançados (Sun *et al*., 2020). Tradicionalmente, os métodos de controle adaptativo podem ser divididos em duas abordagens: controle indireto e controle direto (Qi; Tao; Jiang, 2019). Em controle indireto, a estimação dos parâmetros do sistema precede a geração de uma entrada de controle. Em controle direto, os parâmetros do controlador são diretamente ajustados sem a necessidade das equações que regem o comportamento do sistema.

Na literatura de Controle Adaptativo encontram-se diversos estudos e métodos aplicados ao controle de trajetória de manipuladores robóticos. Dubowsky e Desforges (1979) são os pioneiros em empregar técnicas de controle adaptativo em robôs articulados. A abordagem usada por estes pesquisadores foi o Sistema Adaptativo por Modelo de Referência (*Model Reference Adaptive System*

- MRAS). Resultados práticos também mostraram os benefícios das abordagens baseadas nas técnicas *self-tuning* e *backstepping* em relação ao controle convencional com ganhos fixos (Clegg; Dunnigan; Lane, 2001) (Sasaki *et al*., 2009) (Hu; Xu; Zhang, 2012). Abordagens híbridas também foram exploradas (Maliotis, 1991) (Al-Olimat; Ghandakly, 2002) (Chen, 2005) (Alqaudi *et al*., 2016) (Zhang; Wei, 2017). Em (Wu; Yan; Cai, 2019) (Fateh; Fateh, 2019) (Yilmaz *et al*., 2022) (Freire; Rossomando; Soria, 2018) são propostos projetos de controle adaptativo baseados em técnicas de inteligência artificial, tais como redes neurais e lógica *fuzzy*, que são capazes de compensar as incertezas do modelo de um robô manipulador.

Apesar das técnicas de controle adaptativo terem alcançado sucesso em muitas aplicações, um aspecto que deve ser observado é que os projetos de controladores resultantes desses métodos, em geral, têm sido estruturados sem considerar a otimização da ação de controle e, desta forma, não são viáveis para aplicações onde o uso de estratégias ótimas de controle é requerida, como por exemplo em robôs humanoides/robôs de serviços (Khan *et al*., 2012). Nesse caso, uma abordagem conjunta das técnicas de controle adaptativo e controle ótimo é desejada. Controle ótimo consiste basicamente em determinar uma lei de controle de maneira a minimizar um critério de desempenho desejado. No contexto da robótica, critérios de desempenho podem envolver a energia ou força para a execução do movimento, ao mesmo tempo que devem ser satisfeitas as restrições físicas do sistema, tais como limites dos atuadores ou das juntas.

Muitos esforços na teoria de controle de sistemas estão atualmente concentrados em uma área do aprendizado de máquina baseada nos estudos do comportamento animal e psicologia cognitiva, chamada Aprendizado por Reforço (*Reinforcement Learning* - RL), que visa incorporar características de sistemas biológicos para o tratamento de sistemas com incertezas, introduzindo diversos termos, tais como adaptação, aprendizado, reconhecimento de padrões e auto-organização (Guo; Yan; Cui, 2020) (Yaghmaie; Gustafsson; Ljung, 2023) (Chen; Dai; Dong, 2024a) (Chen; Dong; Dai, 2024b) (Zhao *et al*., 2025) (Su *et al*., 2025) (Wang *et al*., 2025). O tema central na pesquisa de RL é o projeto de algoritmos que aprendem políticas de controle ótimas através do conhecimento apenas de amostras de transição dos estados ou trajetórias, que são coletadas antecipadamente ou pela interação em tempo real com o sistema.

Métodos Ator-Crítico constituem uma classe de técnicas de aprendizado por reforço que consistem essencialmente de duas estruturas paramétricas independentes (por exemplo, redes neurais), uma para representar a política de controle, denominada Ator, e a outra estrutura de rede é para representar a função valor, chamada Crítico (Sutton; Barto, 2018). O ator é um agente que interage com o ambiente, ou seja, o ator é o controlador que estabelece ações de controle, enquanto o crítico avalia o efeito das ações de controle e fornece diretrizes sobre como melhorar a lei de controle.

Aprendizado por reforço pode ser visto em (Kiumarsi *et al*, 2018) na perspectiva de um campo de pesquisa promissor para o projeto de uma classe de controladores adaptativos com estrutura ator-crítico que aprendem *online* soluções de controle ótimo sem fazer uso do modelo da dinâmica do sistema (planta). Esta abordagem resolve a equação de otimização (equação de Hamilton-Jacobi-Bellman - HJB) em uma maneira "para frente no tempo" usando métodos de diferenças temporais, aproximação de funções e melhorias de políticas. Tais controladores são inspirados em estruturas neurais biológicas que fornecem capacidades para lidar de forma eficaz com o grau de complexidade de sistemas não-lineares, incertos e parcialmente observáveis. Em (Kiumarsi *et al*, 2018), são apresentadas as principais ideias e algoritmos de aprendizado por reforço bem como suas aplicações em controle ótimo de sistemas dinâmicos.

1.1 OBJETIVOS

O presente artigo tem por objetivo avaliar o potencial de um algoritmo de aprendizado por reforço para resolver problemas de controle ótimo online da trajetória de um manipulador robótico com espaço de estado contínuo (espaço das juntas). Em contraste com a maioria dos algoritmos ator-crítico reportados na literatura (vide Seção 2), em que se utilizam duas redes neurais, uma para aproximar a função valor, e a outra para aprender ações de controle, o algoritmo proposto neste trabalho emprega uma arquitetura ator-crítico onde uma única rede neural é usada para aproximar a solução da equação HJB, o que reduz significativamente o número de parâmetros a serem estimados. Especificamente, neste esquema, ações de controle são calculadas de maneira exata por meio de um esquema de política gulosa com respeito à função valor, ao invés de se usar um aproximador paramétrico para representar a política de controle. Experimentos realizados em um braço robótico UR10 do simulador V-REP mostram que tal algoritmo aprende com sucesso a lei de controle ótimo para as tarefas de regulação e rastreamento para diferentes sinais de referência.

2 TRABALHOS CORRELATOS

Contribuições anteriores importantes para o projeto de controle fundamentado em RL incluem os trabalhos de Peters e Schaal (2008a) (2008b), que investigaram diversos métodos de aprendizado por reforço para robôs humanoides. Esses métodos foram classificados em três categorias: política gulosa, gradiente de política "*vanilla*" e gradiente de política natural. A abordagem Ator-Crítico natural, que explora a formulação do gradiente de política natural, foi destacada pelos autores por apresentar melhores propriedades de convergência. Uma extensão desse estudo é mostrada em (Bhatnagar *et al*., 2009).

Já (Shah; Gopal, 2009) apresentaram uma abordagem de controle baseada em Aprendizado Q para robôs manipuladores em ambientes incertos e forneceram um estudo comparativo de diferentes métodos de aproximação de função, tais como fuzzy, redes neurais, árvore de decisão e máquina de vetor de suporte.

Em (Khan *et al*., 2011, 2012), os autores enfatizaram aplicações de controladores RL em sistemas robóticos e propuseram um esquema de controle adaptativo ótimo fundamentado em Aprendizado Q (*Q-Learning*) e Programação Dinâmica Aproximada. A estratégia foi implementada no braço de um robô humanoide (Bristol Elumotion-Robotic-Torso II) considerando um caso sem restrições e outro com restrições de movimento.

Em (Pane; Nageshrao; Babuška, 2016), os autores forneceram validação experimental de um compensador baseado em aprendizado Ator-Crítico para melhorar o desempenho de um robô manipulador. O método proposto dispensa a necessidade de aprender o modelo do sistema e pode ser utilizado em qualquer controlador por realimentação (PID, LQR etc.). A validação do método foi demonstrada através de experimentos em um robô manipulador industrial com seis graus de liberdade para diferentes tipos de trajetórias de referência. Uma extensão desse trabalho é apresentada em (Pane *et al*., 2019).

A aplicação de controladores RL em manipuladores robóticos também é mostrada em (Hu; Si, 2018). Nesse trabalho, uma estratégia de Aprendizado Ator-Crítico com observador de estado via rede neural foi implementada para controlar um braço robótico com parâmetros desconhecidos e sujeito a zonas mortas desconhecidas.

Khan *et al*. (2019) propuseram um controle de complacência adaptativo ótimo para um dispositivo robótico de auxílio à locomoção. O esquema de controle sugerido é fundamentado em Aprendizado Q e programação dinâmica aproximada. Esse esquema é completamente independente de modelo dinâmico e emprega realimentação da posição e velocidade da junta, bem como o torque detectado da junta (aplicado pelo usuário durante a caminhada) para controle de complacência. A eficiência do controlador é testada em simulação em um modelo de dispositivo robótico de auxílio à locomoção.

Kamboj *et al*. (2020) apresentaram uma estratégia de controle cinemático ótimo em tempo discreto para um manipulador usando a estrutura Ator-Crítico. A metodologia exposta foi aplicada em um modelo 3D de um manipulador com seis graus de liberdade em experimentos realizados em um software de simulação. Em seguida, implementou-se a estratégia em um robô real do mesmo modelo do simulado.

Em (He *et al*., 2021), os autores discutiram o projeto de controle e a validação de experimentos de um sistema de manipulador flexível de dois elos. Uma estratégia de controle de aprendizado por reforço é desenvolvida com base na estrutura ator-crítico para atenuar vibrações enquanto mantém o rastreamento da trajetória.

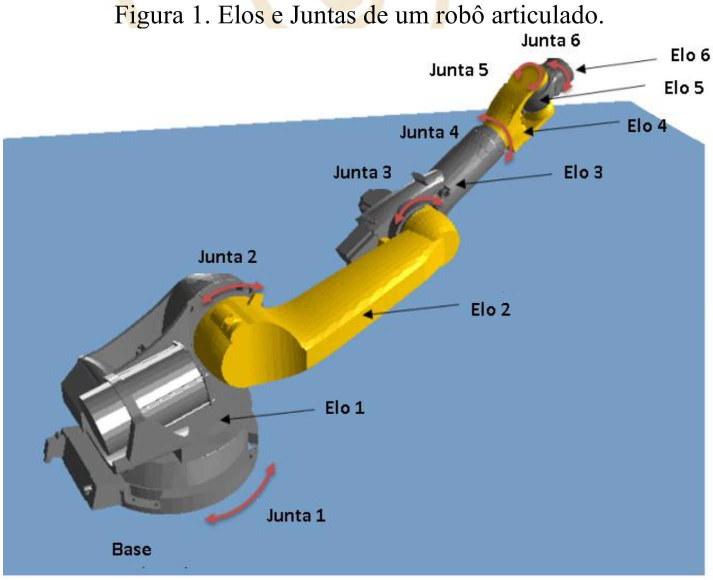
Um controlador de rastreamento baseado em Aprendizado Ator-Crítico para um manipulador também foi estudado por (Cao *et al*., 2023). Nesse trabalho, a técnica de modos deslizantes é utilizada para que a ação obtida pelo esquema Ator-Crítico garanta a convergência do erro de rastreamento em um tempo fixo. Além disso, um compensador *antiwindup* foi projetado para lidar com os efeitos da saturação do atuador da junta.

Na literatura acima, a maioria dos algoritmos RL ator-crítico são implementados utilizando duas redes neurais, uma para aproximar a função valor, e a outra para aprender ações de controle. Para reduzir a complexidade computacional associada com métodos ator-críticos, propõe-se, no presente artigo, uma arquitetura onde uma única rede neural é usada para aproximar a solução de controle ótimo, o que reduz significativamente o número de parâmetros a serem estimados. Especificamente, ações de controle são calculadas de maneira exata por meio de um esquema de política gulosa com respeito à função valor, ao invés de se usar uma aproximação paramétrica para representar a política de controle.

3 DESCRIÇÃO DO SISTEMA MANIPULADOR ROBÓTICO

Um manipulador robótico, ou robô articulado, é formado por um conjunto de corpos individuais conectados entre si formando uma cadeia cinemática capaz de realizar tarefas através da interação com o ambiente (Craig, 2021). As duas partes fundamentais que compõem um robô articulado são os elos, ou articulações, e as juntas. Os elos são as estruturas físicas (rígidas ou flexíveis) que compõem o robô. Já as juntas são responsáveis por promover o movimento relativo entre as articulações por meio de acionadores e são comumente classificadas de acordo com mobilidade que estas viabilizam. Os tipos mais comuns encontrados na indústria são as juntas rotacionais e as prismáticas.

Figura 1. Elos e Juntas de um robô articulado.



Fonte: Abbas, 2018.

A Figura 1 ilustra uma sequência de elos e juntas de um braço robótico. As extremidades do robô articulador são denominadas de base e efetuador. A base fica ligada ao primeiro elo e fixa o mecanismo em algum ponto no espaço de tarefas. O efetuador é uma ferramenta conectada ao último elo do articulador e é por este ponto que há a interação com o ambiente. O tipo de atuador instalado dependerá da tarefa a ser executada.

3.1 EQUAÇÕES DINÂMICAS DE UM MANIPULADOR ROBÓTICO

A dinâmica dos manipuladores estuda a relação entre as forças aplicadas nos atuadores das juntas e o movimento do mecanismo. A formulação de Lagrange permite modelar o comportamento dinâmico de um corpo em termos das energias cinéticas e potenciais ao invés de considerar os momentos e forças aplicadas individualmente em cada junta. A equação de Lagrange é expressa por

$$\tau = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} \qquad (1)$$

$$L(\dot{q}, q) = K(\dot{q}, q) - U(q)\,, \qquad (2)$$

em que $K(\cdot)$ é a energia cinética e $U(\cdot)$ é a energia potencial armazenada no mecanismo. Essa equação é escrita em termos das coordenadas generalizadas q do articulador e sua derivada $\dot{q}$ no tempo. O termo τ, por sua vez, representa o vetor generalizado de forças, incluindo as forças e os torques aplicados no sistema.

Para um robô manipulador com n elos rígidos, a energia cinética pode ser escrita na forma

$$K(\dot{q}, q) = \sum_{i=0}^{n} k_i \qquad (3)$$

$$k_i = \frac{1}{2} m_i v_{C_i}^T v_{C_i} + \frac{1}{2} \omega_i^T \, {}^{C_i}I_i \omega_i\,, \qquad (4)$$

em que k_i é a energia cinética para o i-ésimo elo. Para cada elo, tem-se duas componentes, uma relacionada a velocidade linear v_{C_i}, e a outra, a velocidade angular ω_i, relativas ao centro de massa da respectiva articulação, com m_i a massa do elo i, e ${}^{C_i}I_i$ é a matriz de inércia.

A energia potencial pode ser expressa como

$$U(q) = \sum_{i=0}^{n} u_i \tag{5}$$

$$u_i = m_i g^T P_{C_i} \tag{6}$$

em que u_i é a energia potencial para o i-ésimo elo, definida em termos da massa m_i, do vetor de gravidade g e da localização P_{C_i} do centro de massa relativo à base.

Aplicando-se o lagrangeano $L(\cdot)$ na equação (1), pode-se reordenar os termos da expressão resultante de modo a obter

$$\tau = M(q)\ddot{q} + N(q, \dot{q}) + G(q), \tag{7}$$

em que $M(q)$ é a matriz $n \times n$ de massa do manipulador, $N(q, \dot{q})$ é um vetor de dimensão $n \times 1$ relacionado as forças de Coriolis e centrípeta, e $G(q)$ é um vetor $n \times 1$ com os termos que envolvem a gravidade.

Desse modo, o modelo de um manipulador pode ser escrito na forma de Espaço de Estados por

$$\begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ -M^{-1}(N + G) \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix} \tau. \tag{8}$$

4 METODOLOGIA

No contexto de controle ótimo e aprendizado por reforço, a noção de maximizar recompensas futuras ponderadas é modificada para minimizar o custo de controle. Desta forma, o objetivo é determinar uma lei de controle ou política de controle $h^*(x_k, d_k) = u_k^*$ que minimize o índice de desempenho (função valor)

$$V(x_k, d_k) = \sum_{i=k}^{\infty} \gamma^{i-k} r(x_i, u_i, d_i), \tag{9}$$

onde $x_k \in \mathbb{R}^n$ é o vetor de estado, $u_k \in \mathbb{R}^m$ é o vetor de entrada de controle, d_k é o vetor de trajetória desejada, $0 < \gamma \leq 1$ é o fator de desconto, e $r(\cdot)$ é a função de utilidade que retorna o custo de controle em um passo de tempo. Uma função de utilidade razoavelmente geral em problemas de minimização de energia é dada por:

$$r(x_i, u_i, d_i) = \tilde{r}(x_i, d_i) + u_i^T R u_i, \qquad (10)$$

onde R é uma matriz definida positiva. O vetor d_i pode ser descrito como uma demanda de projeto, fazendo com que $\tilde{r}(\cdot)$ represente o custo para executar a tarefa desejada, como por exemplo, o custo de rastreamento.

Usando o princípio da otimalidade de Bellman (Vrabie; Vamvoudakis; Lewis, 2013), o índice de desempenho ótimo pode ser escrito como

$$V^*(x_k, d_k) = \min_{u_k}\big(r(x_k, u_k, d_k) + \gamma V^*(x_{k+1}, d_{k+1})\big). \qquad (11)$$

Em aprendizado por reforço, uma variante da função valor $V(\cdot)$, chamada função Q (ou função valor ação), é usada. Tal função tem uma aplicação apropriada nos projetos de controle em que o modelo da planta não está disponível. A função Q associada à uma política de controle h é definida por

$$Q^h(x_k,\, u_k,\, d_k) = r(x_k,\, u_k,\, d_k) + \gamma V^h(x_{k+1},\, d_{k+1}), \qquad (12)$$

e a função Q ótima satisfaz a seguinte equação

$$Q^*(x_k,\, u_k,\, d_k) = r(x_k,\, u_k,\, d_k) + \gamma V^*(x_{k+1},\, d_{k+1}). \qquad (13)$$

Combinando as equações (11) e (13), a equação da otimalidade de Belmann em termos da função Q é dada por

$$V^*(x_k,\, d_k) = \min_{u_k}\big(Q^*(x_k,\, u_k, d_k)\big) \qquad (14)$$

e a política de controle ótima é obtida por

$$h^*(x_k,\, d_k) = \arg\min_{u_k} Q^*(x_k, u_k, d_k). \qquad (15)$$

Supondo Q^* suficientemente suave (diferenciável), o sinal de controle pode ser obtido como solução da equação

$$\frac{\partial Q^*(x_k, u_k, d_k)}{\partial u_k} = 0. \tag{16}$$

4.1 ESTRATÉGIA DE APRENDIZADO *ONLINE* ATOR-CRÍTICO

O esquema ator-crítico descrito a seguir considera um sistema manipulador com dois graus de liberdade, podendo ser estendido para manipuladores com n graus de liberdade. A lei de controle é sintetizada no problema de rastreamento ótimo da posição das juntas do manipulador. Em particular, $x_k = [x_{k1} \quad x_{k2} \quad x_{k3} \quad x_{k4}]^T$ é definido como o vetor de estado no instante de tempo k, onde $x_{k1} = q_1$ e $x_{k2} = q_2$ são, respectivamente, a posição angular da junta 1 e da junta 2, e $x_{k3} = \dot{q_1}$ e $x_{k4} = \dot{q_2}$ são, na ordem devida, a velocidade angular da junta 1 e da junta 2. O sinal de controle, naturalmente, é um vetor 2×1 onde $u_k = \tau$ é a força aplicada nas juntas. Para o problema de rastreamento ótimo considerado, a função de utilidade reduz-se a

$$r(x_k,\, e_k,\, u_k) = e_k^T Q_c e_k + (u_{k+1} - u_k)^T S(u_{k+1} - u_k) + u_k^T R u_k \equiv r_k, \tag{17}$$

em que $Q_c \in \mathbb{R}^{4\times 4}$, $R \in \mathbb{R}^{2\times 2}$ e $S \in \mathbb{R}^{2\times 2}$ são matrizes definidas positivas e diagonais.

No presente estudo, a estrutura paramétrica para aproximar a função Q assume a forma dada por

$$\hat{Q}_i(x_k,\, u_k,\, d_k,\, w_i) = w_i^T \phi(x_k,\, u_k,\, d_k), \tag{18}$$

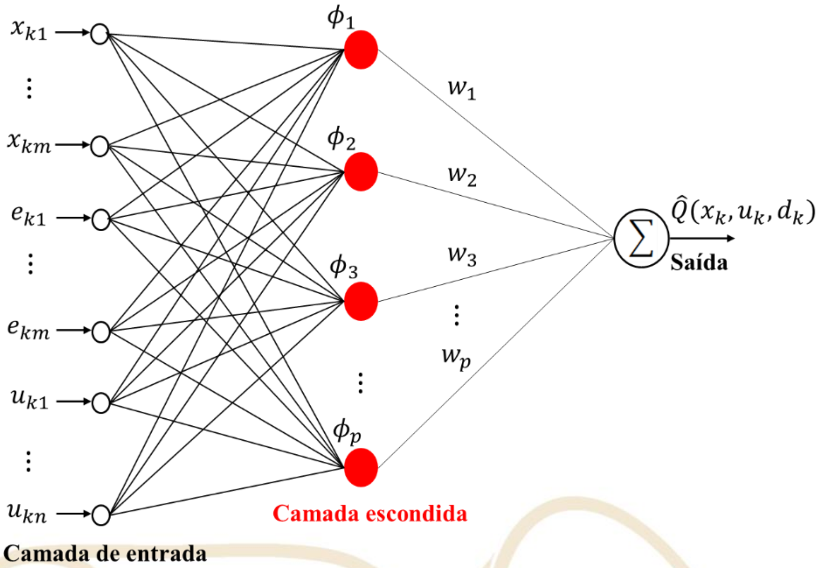
em que w_i é a i-ésima estimação do vetor de pesos da rede neural e $\phi(\cdot)$ é o vetor de funções de ativação ou funções de base. Considera-se que o valor desejado para estimação do parâmetro w_i é dado por

$$\Delta_{objetivo} = r(x_k,\, e_k,\, u_k) + \gamma \hat{Q}_i(x_{k+1}, u_{k+1}, d_{k+1}) \tag{19}$$

A Figura 2 ilustra a arquitetura da rede neural utilizada para estimar a função Q, em que $m = 4$, $n = 2$ e $p = 105$. As funções ϕ_j, $j = 1, \dots, p$, são as componentes do vetor de funções de ativação resultantes do produto de Kronecker dado na equação (21).

O vetor de pesos w_i é calculado pela minimização, em um sentido dos mínimos quadrados, do erro de diferencial temporal, que é definido por

Figura 2. Arquitetura da rede neural utilizada para estimar a função Q.



Fonte: Elaborado pelos autores.

$$\delta_k = r_k + \gamma \hat{Q}_i(x_{k+1}, u_{k+1}, d_{k+1}, w_i) - \hat{Q}_i(x_k, u_k, d_k, w_i). \tag{20}$$

O vetor de funções de ativação é construído por polinômios de ordem superior. Por simplificação, $\phi(\cdot)$ será representado utilizando o produto de Kronecker $\otimes$ com a exclusão dos termos redundantes (Vrabie; Vamvoudakis; Lewis, 2013). Esta exclusão é necessária para que os elementos que compõem o vetor de funções de base $\phi(\cdot)$ tornam-se linearmente independentes. O objetivo é inserir alguns elementos quadráticos e termos de até quarta ordem dos erros de rastreamento, dos estados e dos sinais de controle, de modo que a rede neural possa aprender as não-linearidades do manipulador. Portanto,

$$\phi(z_k) = z_k \otimes z_k \,, \tag{21}$$

em que

$$z_k = [u_k^T \quad e_k^T \quad e_{k1}^2 \quad e_{k2}^2 \quad e_{k3}^2 \quad e_{k4}^2 \quad x_{k1}^2 \quad x_{k2}^2 \quad x_{k3}^2 \quad x_{k4}^2]^T \tag{22}$$

de modo que $e_k = [e_{k1} \quad e_{k2} \quad e_{k3} \quad e_{k4}]^T = x_k - d_k$ é o erro de rastreamento. Desta maneira, a Rede Neural Artificial (RNA) a ser implementada possui 105 neurônios.

A função $\hat{Q}$ toma a forma

$$\hat{Q}_i(x_k, u_k, d_k, w_i) = w_{i,1}^T \varphi_1(z_k) + w_{i,21}^T \varphi_2(z_k) u_{k1} + w_{i,22}^T \varphi_2(z_k) u_{k2} + w_{i,31} u_{k1}^2 + w_{i,32} u_{k2}^2, \tag{23}$$

onde $\phi(z_k) = [\varphi_1^T(z_k) \quad \varphi_2^T(z_k)u_{k1} \quad \varphi_2^T(z_k)u_{k2} \quad u_{k1}^2 \quad u_{k2}^2]^T$ é decorrente da equação (21). Especificamente, os elementos que compõem $\varphi_1(\cdot)$ e $\varphi_2(\cdot)$ são independentes de u_{k1} e u_{k2}.

Aplicando a equação (16) para determinar a política de controle, temos

$$\frac{\partial \hat{Q}_i(x_k, u_k, d_k, w_i)}{\partial u_{k1}} = w_{i,21}^T \varphi_2(z_k) + 2w_{i,31}u_{k1} = 0$$

$$u_{k1} = -\frac{1}{2w_{i,31}} w_{i,21}^T \varphi_2(z_k) \tag{24}$$

$$\frac{\partial \hat{Q}_i(x_k, u_k, d_k, w_i)}{\partial u_{k2}} = w_{i,22}^T \varphi_2(z_k) + 2w_{i,32}u_{k2} = 0$$

$$u_{k2} = -\frac{1}{2w_{i,32}} w_{i,22}^T \varphi_2(z_k). \tag{25}$$

Reorganizando na forma matricial, a política de controle pode ser escrita como

$$h_i(x_k, d_k) = -\frac{1}{2}\begin{bmatrix} w_{i,31} & 0 \\ 0 & w_{i,32} \end{bmatrix}^{-1} \begin{bmatrix} \varphi_2^T(z_k) & \mathbf{0}_{1\times 12} \\ \mathbf{0}_{1\times 12} & \varphi_2^T(z_k) \end{bmatrix} w_{i,2}, \tag{26}$$

em que $w_{i,2} = \begin{bmatrix} w_{i,21}^T & w_{i,22}^T \end{bmatrix}^T$.

Em aprendizado por reforço, o ator é o agente que gera a política de controle, ou seja, o ator é descrito matematicamente pela equação (26). Já o crítico, é descrito pela equação (23).

4.2 ALGORITMO DE APRENDIZADO *ONLINE* ATOR-CRÍTICO

Um aspecto relacionado à abordagem ator-crítico é que as estimativas da função Q de uma dada política de controle são atualizadas a cada passo de tempo k usando dados observados do sistema (estados do manipulador). Para tanto, será utilizado o algoritmo iterativo dos mínimos quadrados recursivos (*Recursive Least-Squares* - RLS) para a estimação do vetor de pesos w_i. A eficiência do método RLS no aprendizado *online* é principalmente devido à sua robustez para lidar com variações nos parâmetros de regressão e a rápida convergência (Ferreira; Rêgo; Neto, 2017).

Portanto, aplicando o algoritmo RLS, a estimativa dos pesos da RNA, a cada passo de tempo k, é dada por

$$w_{k+1} = w_k + K_k \delta_k \tag{27}$$

$$K_k = \frac{P_k \phi(z_k)}{\lambda + \phi(z_k)^T P_k \phi(z_k)} \tag{28}$$

$$P_{k+1} = \frac{1}{\lambda}\left[P_k - \frac{P_k \phi(z_k)\phi(z_k)^T P_k}{\lambda + \phi(z_k)^T P_k \phi(z_k)}\right], \tag{29}$$

sendo λ, $0 < \lambda \leq 1$, o fator de esquecimento e P_k é a matriz de correlação inversa.

O esquema de aprendizado por reforço empregado neste trabalho exige uma política de controle inicial estável. A finalidade é manter o controlador estável durante os instantes iniciais até que o agente adquira experiência suficiente (observando o ambiente) para que uma nova política possa ser calculada. Por simplificação, os ganhos da rede neural devem ser inicializados de modo a resultar em um controlador PD (Proporcional-Derivativo) discreto. Este pode ser implementado modificando os pesos da equação (26), onde observa-se que:

$$\varphi_2(z_k) = [e_{k1} \quad e_{k2} \quad e_{k3} \quad e_{k4} \quad e_{k1}^2 \quad e_{k2}^2 \quad e_{k3}^2 \quad e_{k4}^2 \quad x_{k1}^2 \quad x_{k2}^2 \quad x_{k3}^2 \quad x_{k4}^2]^T, \tag{30}$$

ou seja, $h(x_k, d_k)$ depende diretamente dos erros de posição e velocidade dos elos. Posto isto, constata-se que facilmente pode-se obter um controle PD estabelecendo, por exemplo,

$$\begin{aligned} w_{i,31} &= w_{i,32} = \frac{1}{2}, \\ w_{i,21} &= [K_{P_1} \quad 0 \quad K_{D_1} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T, \\ w_{i,22} &= [0 \quad K_{P_2} \quad 0 \quad K_{D_2} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T, \end{aligned} \tag{31}$$

onde K_{P_1} e K_{D_1}, e K_{P_2} e K_{D_2} são os ganhos proporcional e derivativo, respectivamente, das juntas 1 e 2. O ajuste desses parâmetros será realizado por tentativa e erro.

Um resumo do algoritmo de aprendizado por reforço ator-crítico implementado neste estudo é apresentado a seguir.

Algoritmo RL Ator-Crítico

Entrada: fator de desconto γ, fator de aprendizado α, valor inicial da matriz de covariância β e o fator de esquecimento λ.

Inicialize os pesos da rede neural de forma a garantir um controlador PD estável. Meça os estados x_0 e os erros de trajetória e_0 iniciais. Inicialize as matrizes $P_0 = \beta I$, Q_c, R e S arbitrariamente e $i = 0$.

Repita para cada amostra dos estados $k = 0, 1, 2, \ldots$

▷ **Sinal de ruído como componente de exploração**

$\xi = [\,]$

▷ **Sinal de controle**

$u_k = h_i(x_k, d_k) + \xi$

Aplique u_k e meça os estados x_{k+1}

$u_{k+1} = h_i(x_{k+1}, d_{k+1})$

$e_{k+1} = x_{k+1} - d_{k+1}$

$r_k = e_k^T Q_c e_k + (u_{k+1} - u_k)^T S(u_{k+1} - u_k) + u_k^T R u_k$

▷ **Mínimos quadrados recursivos - RLS**

$\Delta_{objetivo} = r_k + \gamma \hat{Q}(x_{k+1}, u_{k+1}, e_{k+1})$

$\widehat{W}_k = w_k^T \phi_k$

$K_k = \frac{P_k \phi_k}{\lambda + \phi_k^T P_k \phi_k}$

$w_{k+1} = w_k + K_k\left(\Delta_{objetivo} - \widehat{W}_k\right)$

$P_{k+1} = \frac{1}{\lambda}\left(P_k - \frac{P_k \phi_k \phi_k^T P_k}{\lambda + \phi_k^T P_k \phi_k}\right)$

Se fim de um período de aprendizado:

$w_{i+1(ctrl)} = \alpha w_{k+1} + (1 - \alpha) w_{i(ctrl)}$

▷**Atualização da política de controle**

$h_{i+1} \leftarrow -\frac{1}{2}\begin{bmatrix} w_{i+1(ctrl),31} & 0 \\ 0 & w_{i+1(ctrl),32} \end{bmatrix}^{-1} \begin{bmatrix} \varphi_2^T(z_k) & \mathbf{0}_{1\times 10} \\ \mathbf{0}_{1\times 10} & \varphi_2^T(z_k) \end{bmatrix} w_{i(ctrl),2}$

$P_{k+1} = \beta I$

$i = i + 1$

fim-se

até satisfazer o critério de parada

O algoritmo inicia-se com os ganhos da RNA definidos arbitrariamente para produzir o efeito de um controle PD. Considerou-se a condição inicial da matriz de correlação inversa do RLS dada na forma $P_0 = \beta I$, em que β é uma constante com valor suficientemente grande e I é a matriz identidade. Durante os primeiros instantes, não há atualização na política de controle para garantir a estabilidade durante o aprendizado inicial, entretanto o vetor de pesos w_k é calculado a cada passo aplicando as equações (27) a (29). O sinal de controle é obtido em cada instante de tempo k usando (26). Um sinal de ruído ξ, conhecido como ruído de exploração, é adicionado na entrada de controle com o propósito de aprendizado online (Jiang; Jiang, 2017). Ao fim desse período, os pesos do controlador são atualizados, iniciando-se um novo período de aprendizagem. Para fornecer robustez ao algoritmo, a atualização dos parâmetros da política é obtida por

$$w_{i+1(ctrl)} = \alpha w_{k+1} + (1 - \alpha) w_{i(ctrl)}, \qquad (32)$$

onde $w_{i(ctrl)}$ são os parâmetros do controlador implementado durante o i-ésimo ciclo, $0 < \alpha \leq 1$ é o fator de aprendizado. Nesse instante, a matriz P é redefinida. Os pesos do controlador são novamente mantidos inalterados até que o ciclo em curso tenha se concluído. O processo é repetido até a convergência dos parâmetros da rede. Alcançado este objetivo, o controlador opera com pesos constantes.

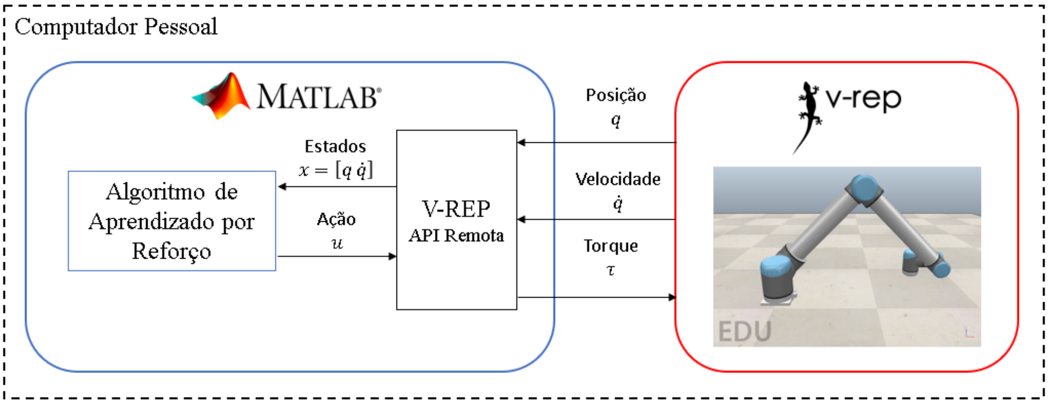
5 ESTRUTURA DE SIMULAÇÃO

De modo a fornecer uma estrutura de simulação que permita desenvolver os algoritmos e realizar os experimentos foi utilizado o *software* V-REP (*Virtual Robotics Experimentation Plataform*) em conjunto com o MATLAB (*Matrix Laboratory*). O V-REP é um simulador para robôs de propósito geral que fornece vários motores de física para as simulações, diversos modelos robóticos e múltiplas configurações do ambiente. Desta forma, é possível personalizar todos os objetos da cena, incluindo os parâmetros dos sensores e atuadores, permitindo assim atingir resultados mais fiéis (Rohmer; Singh; Freese, 2013).

No V-REP são disponibilizados diferentes meios de controlar os objetos/modelos na cena, seja através de rotinas embarcadas, nós do ROS (*Robot Operating System*) (Quigley; Gerkey; Smart, 2015), API (*Application Programming Interface*) remota, um plugin ou alguma solução personalizada. Os controladores podem ser escritos em C/C++, Python, Java, Lua e MATLAB (Shamshiri *et al*., 2018). Neste estudo, o modelo robótico usado no simulador é controlado por uma rotina externa desenvolvida na plataforma MATLAB fazendo uso da API remota. A Figura 3 ilustra a comunicação entre o controlador e o ambiente de simulação.

As configurações a serem seguidas para o funcionamento adequado das simulações usando as plataformas descritas acima e dentro do contexto de aprendizado por reforço podem vistas em detalhes em (Pluškoski; Ciganović; Jovanović, 2019).

Figura 3. Esquema de controle do V-REP por API remota via MATLAB.

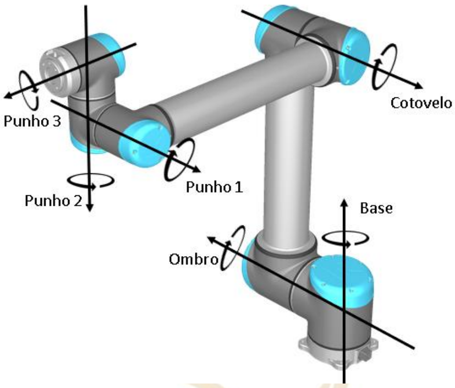


Fonte: Elaborado pelos autores.

6 RESULTADOS DE SIMULAÇÃO

Nesta seção, os resultados das simulações do esquema de controle proposto neste trabalho são apresentados e discutidos. Para execução desses experimentos computacionais foi utilizado o modelo do braço robótico UR10 disponível no simulador V-REP. Visto que este articulador possui seis graus de liberdade, nestes ensaios o torque gerado pela lei de controle será aplicado apenas nas juntas do ombro e do cotovelo (Figura 4) enquanto as demais juntas são desativadas e bloqueadas em suas respectivas posições de equilíbrio (0º). O controle foi realizado utilizando a API remota através de rotinas implementadas na plataforma MATLAB.

Figura 4. Juntas do manipulador UR10.

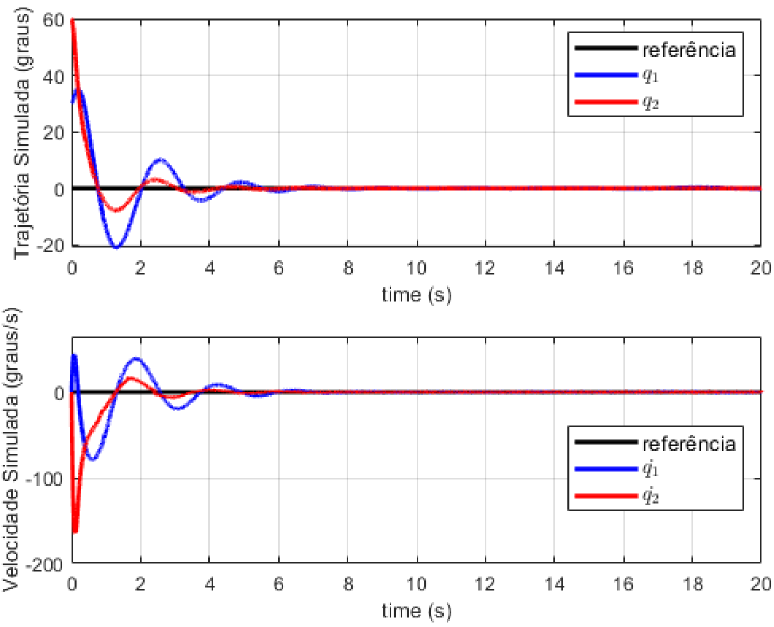


Fonte: Elaborado pelos autores.

A avaliação do esquema de controle via aprendizado por reforço será feita pela análise dos resultados de simulações de três tarefas: regulação, seguimento de trajetória de um sinal de múltiplos degraus e um sinal senoidal.

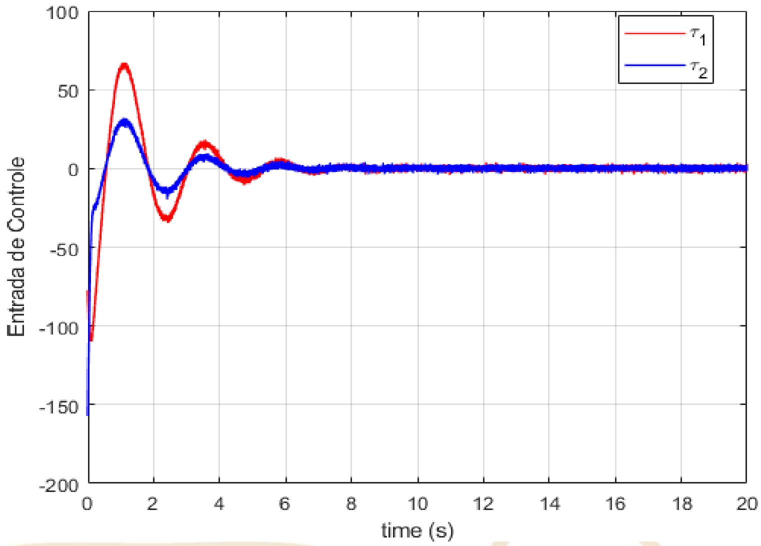
O comportamento dos estados para o caso de regulação é apresentado na Figura 5. A configuração inicial das juntas foi definida como $x_0 = [\pi/6 \quad \pi/3 \quad 0 \quad 0]^T$ e os parâmetros do controlador foram ajustados para os seguintes valores $K_{P_1} = K_{P_2} = 150$, $K_{D_1} = K_{D_2} = 30$, $\gamma = 0{,}98$, $Q_c = diag(250,\ 250,\ 0{,}001,\ 0{,}001)$, $R = diag(0{,}0001,\ 0{,}0001)$, $\alpha = 0{,}2$ e $P_0 = 10^4 I_{78\times78}$. O ciclo de aprendizado para esta simulação foi de 0,8 s. O esforço de controle aplicado nas juntas é apresentado na Figura 6 e a atualização dos pesos do ator é exibida na Figura 7.

Figura 5. Trajetória dos estados.



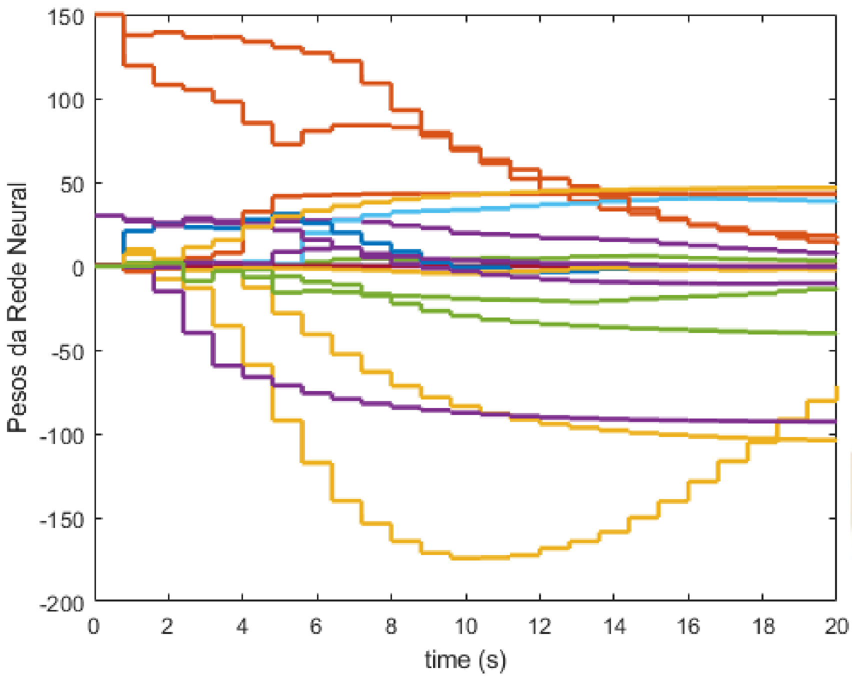
Fonte: Elaborado pelos autores.

Figura 6. Sinal de controle.



Fonte: Elaborado pelos autores.

Figura 7. Atualização dos pesos da rede do ator.

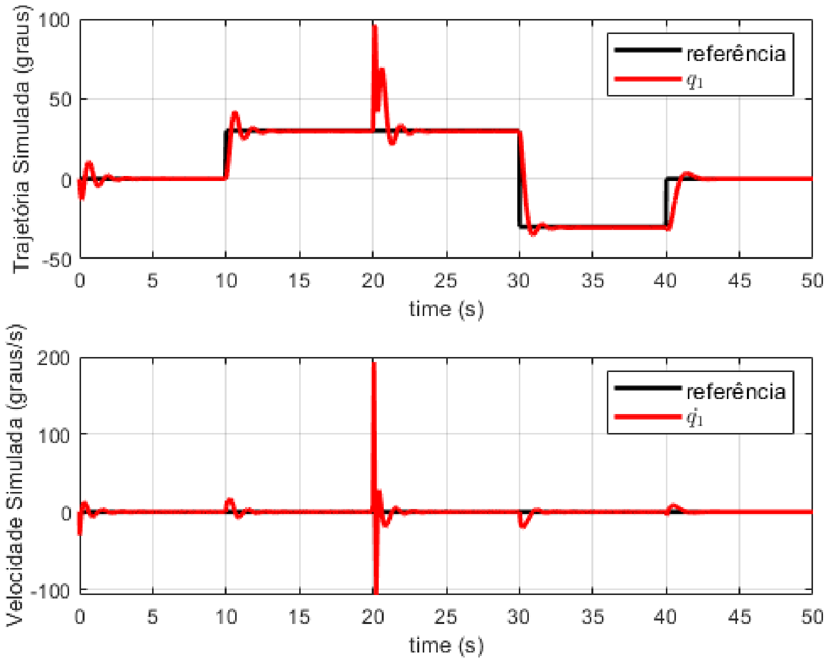


Fonte: Elaborado pelos autores.

Na segunda experiência sugerida para validar o controlador implementado, foi utilizado um sinal de referência de múltiplos degraus, de modo a simular a tarefa de pegar e colocar (*pick and place*), comumente realizada por manipuladores. Para este experimento o estado inicial foi configurado em $x_0 = [0 \quad 0 \quad 0 \quad 0]^T$. Os parâmetros de controle foram os mesmos utilizados para o caso de regulação exceto para os valores seguintes $Q_c = diag(100, 100, 0{,}001, 0{,}001)$, $\alpha = 0{,}1$, $K_{P_1} = K_{P_2} = 500$, $K_{D_1} = K_{D_2} = 50$ e ciclo de aprendizado alterado para 2 s. Sob estes ajustes, a resposta de rastreamento, o torque aplicado nas juntas e a atualização dos pesos da rede do ator são apresentados nas Figuras 8 a 11.

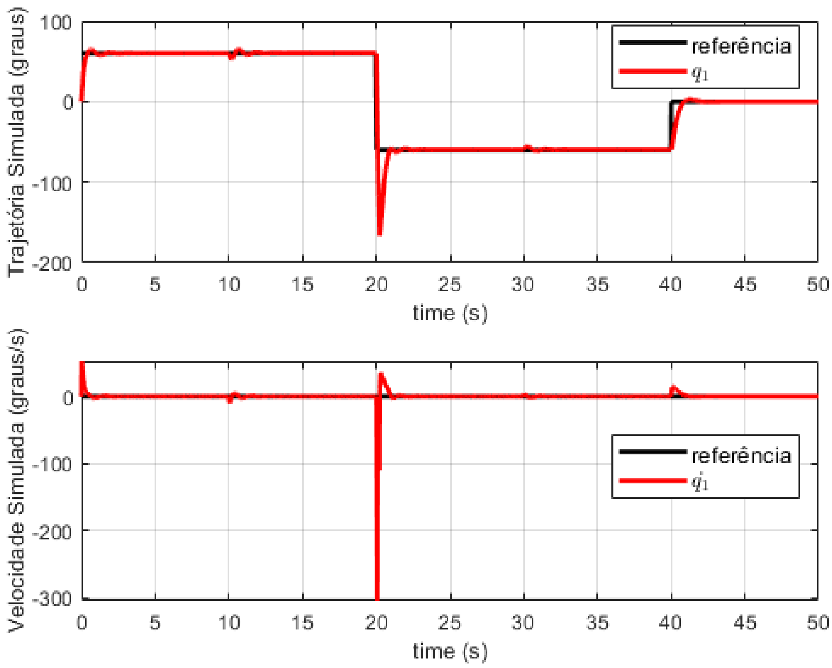
Como visto, as juntas são capazes de alcançar o sinal de referência com erros dentro dos limites aceitáveis e a estabilidade do sistema é mantida durante todo o tempo de simulação. É mostrado também que no instante de tempo de 20 s houve um aumento no sinal de controle causando um sobressinal indesejado, porém nos instantes seguintes, a partir da 15ª atualização da política (30 s), observou-se um aprimoramento no rastreamento em relação a política inicial (primeiros 2 s), consequência do aprendizado adquirido.

Figura 8. Seguimento de trajetória da junta do ombro.



Fonte: Elaborado pelos autores.

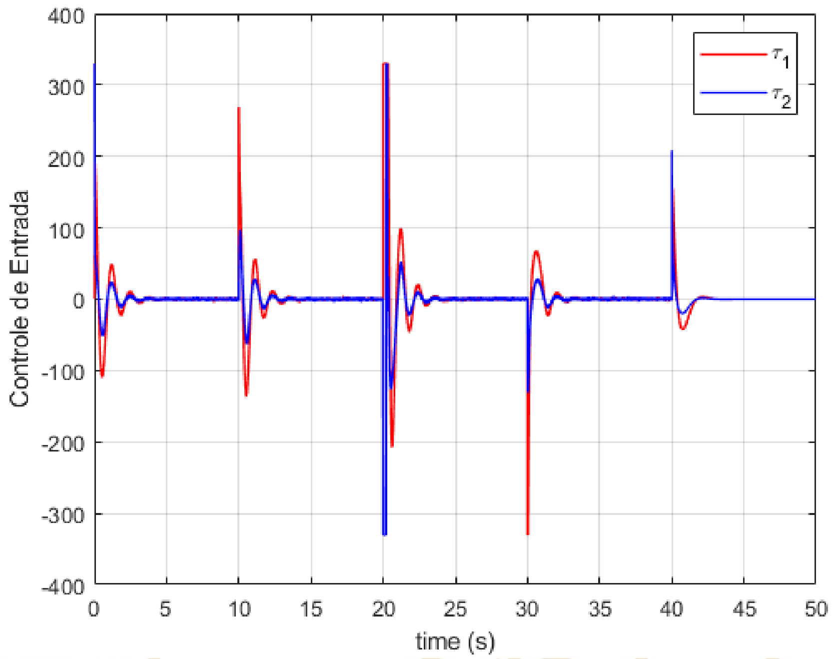
Figura 9. Seguimento de trajetória da junta do cotovelo.



Fonte: Elaborado pelos autores.

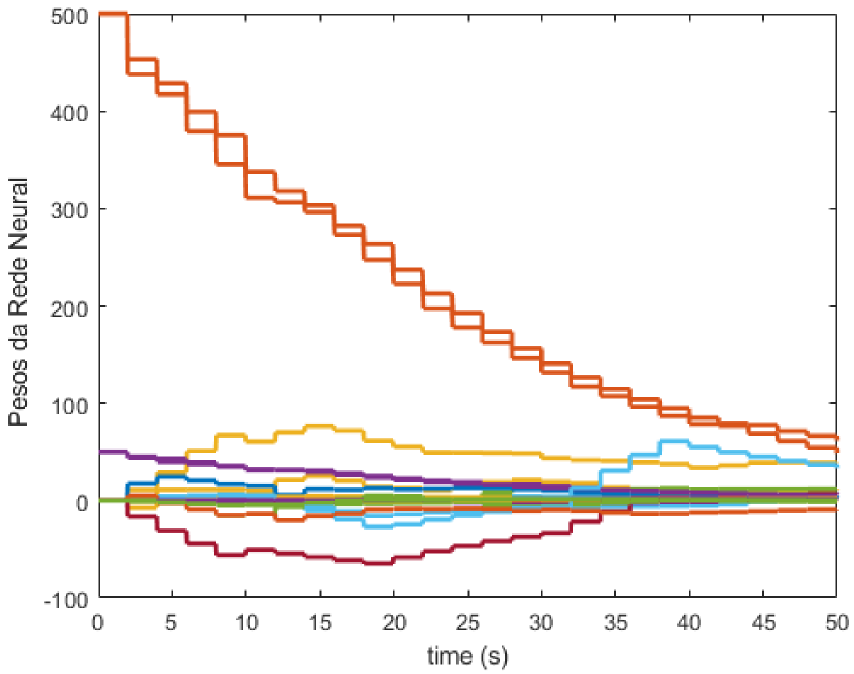
BOCA

Figura 10. Sinal de controle.



Fonte: Elaborado pelos autores.

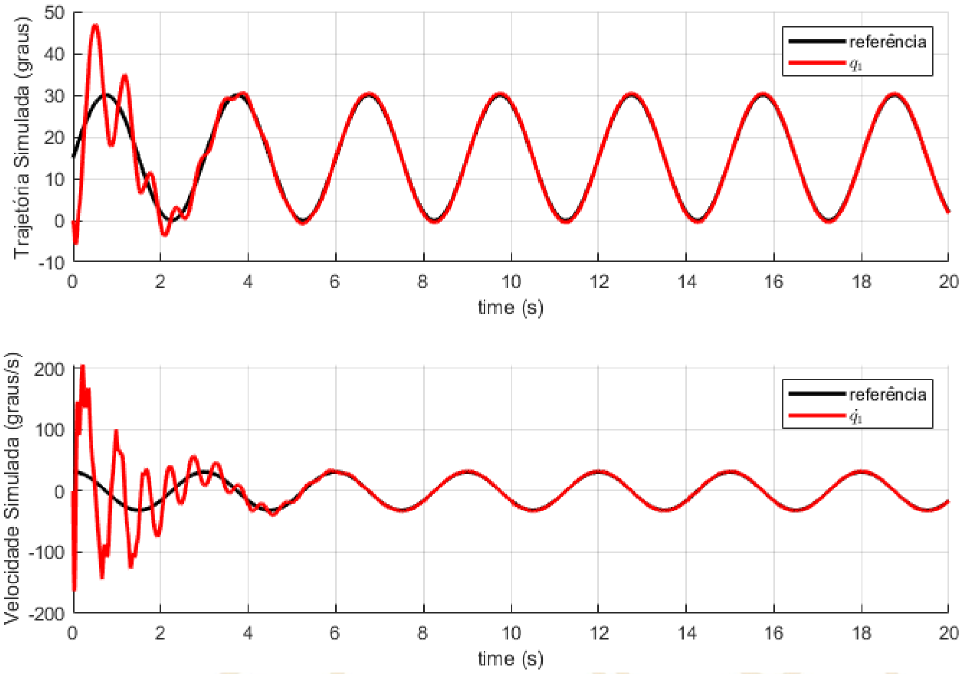
Figura 11. Atualização dos pesos da rede do ator.



Fonte: Elaborado pelos autores.

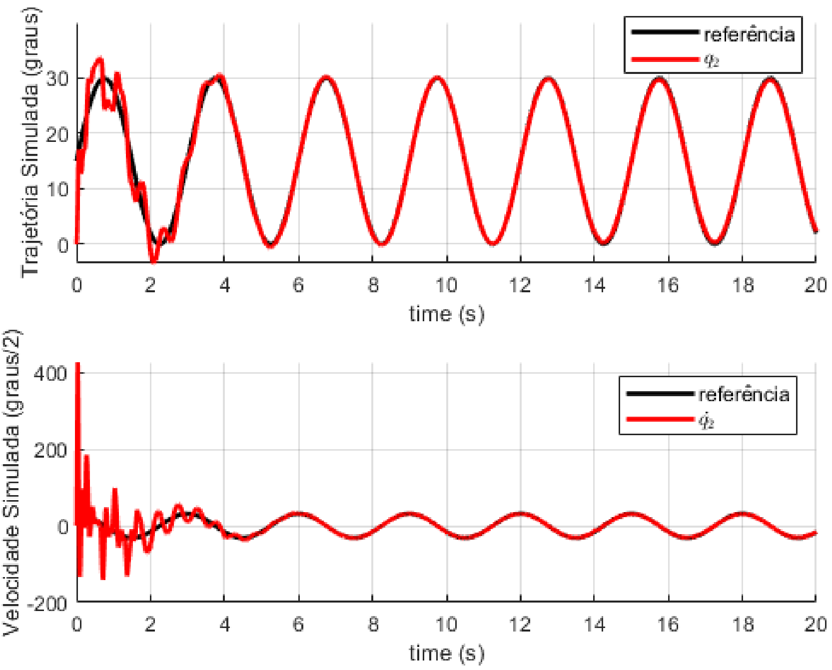
No último experimento proposto, um sinal senoidal foi estabelecido como referência para as juntas do articulador sob os seguintes ajustes $Q_c = diag(200, 200, 0{,}001, 0{,}001)$, $K_{P_1} = 4000$, $K_{P_2} = 2000$, $K_{D_1} = 50$, $K_{D_2} = 20$ e $\alpha = 0{,}4$. Os demais parâmetros foram configurados nos mesmos valores do experimento 2. Os resultados da simulação são observados nas Figuras 12 a 15. De acordo com as Figuras 12 e 13, onde é mostrado o desempenho de rastreamento, observa-se o aprimoramento do seguimento de trajetória ao fim de cada ciclo de aprendizado (intervalos de 2 s). A partir do terceiro ciclo os erros de rastreamento se estabilizam dentro de limites toleráveis.

Figura 12. Seguimento de trajetória da junta do ombro.



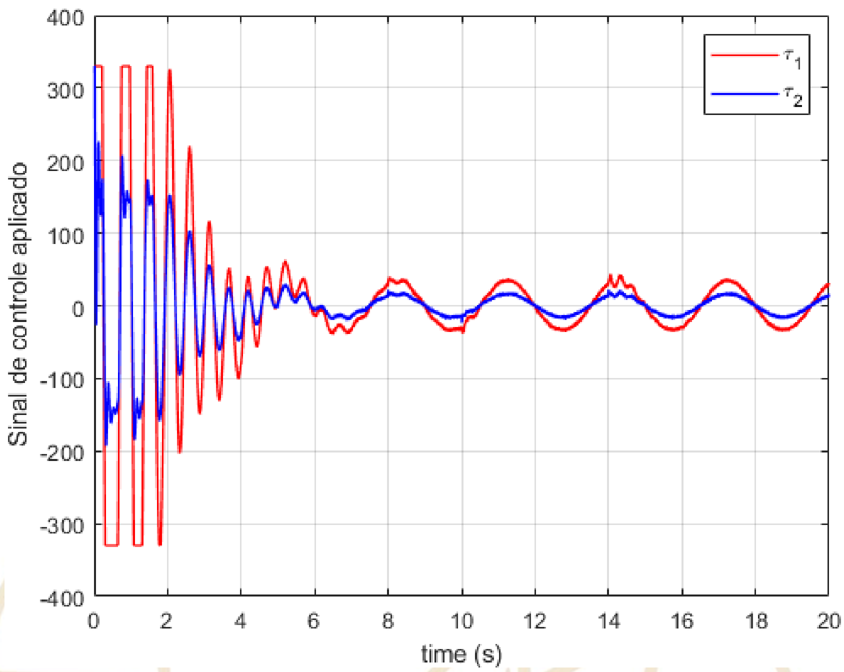
Fonte: Elaborado pelos autores.

Figura 13. Seguimento de trajetória da junta do cotovelo.



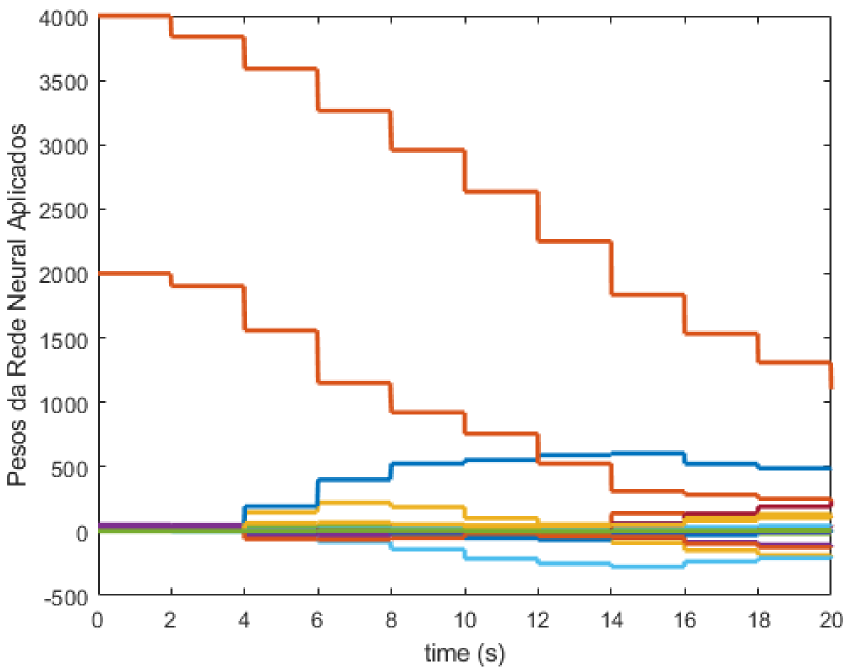
Fonte: Elaborado pelos autores.

Figura 14. Sinal de controle.



Fonte: Elaborado pelos autores.

Figura 15. Atualização dos pesos da rede do ator.



Fonte: Elaborado pelos autores.

7 CONCLUSÃO

Neste trabalho foi proposto um esquema de controle baseado em aprendizado por reforço aplicado em um manipulador robótico, usando uma abordagem ator-crítico. Neste projeto, apenas uma rede neural foi treinada para aproximar a função Q usando apenas as medidas reais do sistema via o estimador RLS. A fim de fornecer robustez ao esquema, a atualização da política de controle, obtida pela minimização da função Q, ocorre ao fim de um número fixo de iterações (ciclo de aprendizado), mantendo-se constante durante este intervalo. Para aproximar a função valor ação, uma rede neural polinomial foi utilizada, mostrando-se adequada para aprender as não linearidades do manipulador. Experimentos computacionais com o controlador apresentado foram realizados utilizando o modelo do robô UR10 no simulador V-REP. As simulações incluíram a realização da tarefa de regulação e seguimento de trajetória dos sinais senoidal e de múltiplos degraus. Nos resultados simulados, observou-se a estabilidade das variáveis de estado durante todo o tempo de simulação e a capacidade de rastreamento dos sinais de referência, mesmo sem o conhecimento explícito da dinâmica do manipulador.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Pró-Reitoria de Pesquisa e Pós-Graduação - PPG da Universidade Estadual do Maranhão - UEMA.

REFERÊNCIAS

ABBAS, Z. **Motion control of robotic arm manipulator using PID and sliding mode technique**. 2018. Tese (Doutorado em Engenharia Elétrica) – Capital University of Science and Technology, Islamabad, 2018.

AL-OLIMAT, K. S.; GHANDAKLY, A. A. Multiple model reference adaptive control algorithm using on-line fuzzy logic adjustment and its application to robotic manipulators. *In*: **Conference Record of the 2002 IEEE Industry Applications Conference**. 37th IAS Annual Meeting, Pittsburgh, PA, USA, 2002, p. 1463-1466.

ALQAUDI, B. *et al*. Model reference adaptive impedance control for physical human-robot interaction. **Control Theory and Technology**. v. 14, p. 68-82, fev. 2016.

BHATNAGAR, S. *et al*. Natural actor-critic algorithms. **Automatica**, v. 45, n. 11, p. 2471-2482, nov. 2009.

BORASE, R. P. *et al*. A review of PID control, tuning methods and applications. **International Journal of Dynamics and Control**. v. 9, p. 818-827, 2021.

CAO, S. *et al*. Reinforcement learning-based fixed-time trajectory tracking control for uncertain robotic manipulators with input saturation. **IEEE Transactions on Neural Networks and Learning Systems**, v. 34, n. 8, p. 4584-4595, ago. 2023.

CHEN, L.; DAI, S.-L.; DONG, C. Adaptive optimal tracking control of an underactuated surface vessel using actor–critic reinforcement learning. **IEEE Transactions on Neural Networks and Learning Systems**, v. 35, n. 6, p. 7520-7533, jun. 2024a.

CHEN, L.; DONG, C.; DAI, S.-L. Adaptive optimal consensus control of multiagent systems with unknown dynamics and disturbances via reinforcement learning. **IEEE Transactions on Artificial Intelligence**, v. 5, n. 5, p. 2193-2203, maio 2024b.

CHEN, W.-D. Experimental study of robot manipulators based on robust adaptive control. *In*: **International Conference on Machine Learning and Cybernetics**, Guangzhou, China, 2005, p. 18-21.

CLEGG, A. C.; DUNNIGAN, M. W.; LANE, D. M. Self-tuning position and force control of an underwater hydraulic manipulator. *In*: **Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation**, Seoul, Korea (South), maio. 2001, p. 3226-3231.

CRAIG, J. J. **Introduction to robotics: mechanics and control**, Ed. 4. Global Edition. São Paulo: Pearson, 2021.

DUBOWSKY, S.; DESFORGES, D. T. The application of model-referenced adaptive control to robotic manipulators. **Journal of Dynamic Systems, Measurement, and Control**, v. 101, n. 3, p. 193-200, set. 1979.

FATEH, S.; FATEH, M. M. Adaptive fuzzy control of robot manipulators with asymptotic tracking performance. **Journal of Control, Automation and Electrical Systems**, v. 31, p. 52-61, out. 2019.

FERREIRA, E. F. M.; RÊGO, P. H. M.; NETO, J. V. F. 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, ago. 2017.

FREIRE, E. O.; ROSSOMANDO, F. G; SORIA, C. M. Self-tuning of a neuro-adaptive PID controller for a SCARA robot based on neural network. **IEEE Latin America Transactions**, v. 16, n. 5, p. 1364-1374, jul. 2018.

GUO, X.; YAN, W.; CUI, R. Reinforcement learning-based nearly optimal control for constrained-input partially unknown systems using differentiator. **IEEE Transactions on Neural Networks and Learning Systems**, v. 31, n. 11, p. 4713-4725, nov. 2020.

HE, W. *et al*. Reinforcement learning control of a flexible two-link manipulator: an experimental investigation. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, v. 51, n. 12, p. 7326-7336, dez. 2021.

HU, Q.; XU, L.; ZHANG, A. Adaptive backstepping trajectory tracking control of robot manipulator. **Journal of the Franklin Institute**. v. 349, n. 3, p. 1087-1105, 2012.

HU, Y.; SI, B. A reinforcement learning neural network for robotic manipulator control. **Neural Computation**. v. 30, n. 7, p. 1983-2004, jul. 2018.

JIANG, Y.; JIANG, Z.-P. **Robust adaptive dynamic programming**. Hoboken, New Jersey: John Wiley & Sons, Inc., 2017.

KAMBOJ, A. *et al*. L. Discrete-time Lyapunov based kinematic control of robot manipulator using actor-critic framework. *In*: **2020 International Joint Conference on Neural Networks (IJCNN)**, Glasgow, United Kingdom, 2020, p. 1-7.

KHAN, S. G. *et al*. Reinforcement learning based compliance control of a robotic walk assist device, **Advanced Robotics**, v.33, n. 24, p. 1281-1292, nov. 2019.

KHAN, S. G. *et al*. A Q-learning based Cartesian model reference compliance controller implementation for a humanoid robot arm. *In*: **2011 IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM)**, Qingdao, China, set. 2011, p. 214-219.

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, 2012.

KIUMARSI, B. *et al*. Optimal and autonomous control using reinforcement learning: a survey. **IEEE Transactions on Neural Networks and Learning Systems**, v. 29, n. 6, p. 2042-2062, jun. 2018.

KONSTANTOPOULOS, G. C.; BALDIVIESO-MONASTERIOS, P. R. State-limiting PID controller for a class of nonlinear systems with constant uncertainties. **International Journal of Robust and Nonlinear Control**, v. 30, p. 1770-1787, 2020.

MALIOTIS, G. A. Hybrid model reference adaptive control/computed torque control scheme for robotic manipulators. **Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering**, v. 205, n. 3, p. 215-21, 1991.

MOOSAVI, S. K. R.; ZAFAR, M. H.; SANFILIPPO, F. Forward kinematic modelling with radial basis function neural network tuned with a novel meta-heuristic algorithm for robotic manipulators. **Robotics**, v. 11, n. 2, p. 1-17, abr. 2022.

PANE, Y. P. *et al*. Reinforcement learning based compensation methods for robot manipulators. **Engineering Applications of Artificial Intelligence**, v. 78, p. 236-247, fev. 2019.

PANE, Y. P.; NAGESHRAO, S. P.; BABUŠKA, R. Actor-critic reinforcement learning for tracking control in robotics. *In*: **2016 IEEE 55th Conference on Decision and Control (CDC)**, Las Vegas, NV, USA, dez. 2016, p. 5819-5826.

PETERS, J.; SCHAAL, S. Learning to control in operational space. **International Journal of Robotics Research**, v. 27, n. 2, p. 197-212, fev. 2008a.

PETERS, J.; SCHAAL, S. Natural actor-critic. **Neurocomputing**, v. 71, n. 7-9, p. 1180-1190, marc. 2008b.

PLUŠKOSKI, A.; CIGANOVIĆ, I.; JOVANOVIĆ, M. D. Benefits of Residual Networks in Reinforcement Learning using V-Rep Simulator. *In*: **6th International Conference IcETRAN**, Silver Lake, Serbia, jun. 2019, p. 1-6.

QI, R.; TAO, G.; JIANG, B. **Adaptive control: a tutorial introduction**. In book: Fuzzy system identification and adaptive control. Communications and Control Engineering. Springer, Cham., 2019, p. 55-74.

QUIGLEY, M.; GERKEY, B.; SMART, W. D. **Programming robots with ROS:** a practical introduction to the robot operating system. Ed. 1. O'Reilly Media, Inc. 2015.

ROHMER, E.; SINGH, S. P. N.; FREESE, M. V-REP: A versatile and scalable robot simulation framework. *In*: **IEEE/RSJ International Conference on Intelligent Robots and Systems**, Tokyo, Japan, nov. 2013.

SASAKI, M. *et al*. Self-tuning control of a two-link flexible manipulator using neural networks. *In*: **2009 ICCAS-SICE**, Fukuoka, Japan, ago. 2009, p. 2468-2473.

SHAH, H.; GOPAL, M. Reinforcement learning control of robot manipulators in uncertain environments. *In*: **IEEE International Conference on Industrial Technology**, Churchill, VIC, Australia, fev. 2009, p. 1-6.

SHAMSHIRI, R. R. *et al*. **Robotic harvesting of fruiting vegetables:** A simulation approach in V-REP, ROS and MATLAB. In (Ed.), Automation in Agriculture - Securing Food Supplies for Future Generations. IntechOpen. mar. 2018.

SU, Y. *et al*. Fixed-time optimal trajectory tracking control for an unmanned surface vehicle via reinforcement learning. **IEEE/ASME Transactions on Mechatronics**, p. 1-12, set. 2025.

SUN, N. *et al*. Adaptive control for pneumatic artificial muscle systems with parametric uncertainties and unidirectional input constraints. **IEEE Transactions on Industrial Informatics**, v. 16, n. 2, p. 969-979, fev. 2020.

SUTTON, R. S.; BARTO, A. G. **Reinforcement learning: An Introduction**. Ed. 2. Cambridge, Massachusetts: MIT Press, 2018.

VRABIE, D.; VAMVOUDAKIS, K. G.; LEWIS, F. L. **Optimal adaptive control and differential games by reinforcement learning principles**. London, United Kingdom: The Institution of Engineering and Technology, 2013.

WANG, Z. *et al*. Adaptive altitude control for underwater vehicles based on deep reinforcement learning. *In*: **2025 8th International Conference on Transportation Information and Safety (ICTIS)**, Granada, Spain, 2025, p. 79-84.

WU, L.; YAN, Q; CAI, J. Neural network-based adaptive learning control for robot manipulators with arbitrary initial errors. **IEEE Access**, v. 7, p. 180194-180204, dez. 2019.

YAGHMAIE, F. A.; GUSTAFSSON, F.; LJUNG , L. Linear quadratic control using model-free reinforcement learning. **IEEE Transactions on Automatic Control**, v. 68, n. 2, p. 737-752, fev. 2023.

YILMAZ, B. M. *et al*. Self-adjusting fuzzy logic based control of robot manipulators in task space. **IEEE Transactions on Industrial Electronics**, v. 69, n. 2, p. 1620-1629, fev. 2022.

ZHANG, D.; WEI, B. Design, analysis and modelling of a hybrid controller for serial robotic manipulators. **Robotica**, v. 35, n. 9, p. 1888-1905, set. 2017.

ZHAO, D. *et al*. Linear quadratic control of unknown nonlinear systems using model-free reinforcement learning. **IEEE Transactions on Industrial Electronics**, v. 72, n. 12, p. 13751-13762, dez. 2025.