Projeto de Controle Ótimo RLQD em Sistemas MIMO com Abordagens PDH e PDHDA no Controle de Níveis de Líquido em Tanques ⋆

Evandro Martins Araújo Filho * **João Viana da Fonseca Neto** ** **José Pinheiro de Moura** ***

* *Programa de Pós-graduação em Engenharia Elétrica-PPGEE, Universidade Federal do Maranhão São Luís, Brasil, (e-mail: evando.martins@dee.ufma.br*
** *Universidade Federal do Maranhão, Departamento de Engenharia Elétrica, Programa de Pós-graduação em Engenharia Elétrica-PPGEE São Luís, Brasil, (e-mail: joao.fonseca@ufma.br)*
*** *Universidade Estadual do Maranhão, Departamento de Engenharia da Computação, Programa de Pós-Graduação em Engenharia da Computação e Sistemas (PECS), São Luís, Brasil, (e-mail: josepinheiro@professor.uema.br)*

Abstract: In this article, a methodology for the implementation of online projects of adaptive-optimal control systems in a multivariable tank level system is presented. Adaptive-optimal control approaches are explored, including the use of heuristic dynamic programming (HDP) and action-dependent heuristic dynamic programming (AD-HDP). These approaches are compared with the Schur method, which utilizes a complete model of the system. Evaluation is carried out by applying a discrete linear quadratic regulator (DLQR) in a small-scale plant representing a water supply system.

Resumo: Neste artigo, apresenta-se uma metodologia para a implementação de projetos *online* de sistemas de controle ótimo adaptativo em um sistema multivariável de nível de tanque. São exploradas abordagens de controle adaptativo-ótimo, incluindo o uso de programação dinâmica heurística (PDH) e programação dinâmica heurística dependente de ação (PDHDA). Essas abordagens são comparadas com o método de *Schur*, que utiliza um modelo completo do sistema. A avaliação é feita aplicando um Regulador Linear Quadratico Discreto (RLQD) em uma planta de pequena escala que representa um sistema de abastecimento de água.

Keywords: Adaptive optimal control; Multivariable system; Heuristic dynamic programming; Action-dependent heuristic dynamic programming; Discrete regulator.

Palavras-chaves: Controle ótimo adaptativo; Sistema multivariável; Programação dinâmica heurística; Programação dinâmica heurística dependente de ação; regulador discreto.

1. INTRODUÇÃO

Controladores ótimos podem ser obtidos por meio de vários métodos, incluindo a programação dinâmica. A equação de Hamilton-Jacobi-Bellman (HJB) é o ponto central para a implementação de algoritmos de programação dinâmica (Bellman, 1958) e (Bertsekas, 2012). No entanto, a principal desvantagem da programação dinâmica clássica é que os métodos não são adequados para aplicações em tempo real devido à complexidade computacional associada à resolução da equação HJB, o que é caro para problemas de controle ótimo em larga escala com vários estágios (Lendaris, 2009).

Na prática, um dos principais problemas na aplicação de técnicas de controle é a variação de parâmetros e perturbações durante a operação de um sistema. Para obter a saída desejada, é necessário implementar um controlador que se adapte a essas mudanças. Em (Uematsu et al., 2022), os autores abordam alguns métodos de identificação em malha fechada, sendo um deles baseado na avaliação da variância mínima generalizada (GMV, do inglês Generalized Minimum Variance).

O método de controle baseado em modelo (CBM) requer conhecimento do modelo da planta. Neste trabalho aborda-se o projeto e a análise de controle ótimo por meio de Regulador Linear Quadrático (RLQ) e Regulador Linear Quadrático Discreto (RLQD) usando técnicas de Programação Dinâmica Heurística (PDH) e Programação Dinâmica Heurística Dependente de Ação (PDHDA) para controle de níveis de líquidos em tanques.

Apresentação do trabalho: Na Seção 2, descreve-se o modelo da planta, incluindo espaço de estado em tempo contínuo e discreto, bem como a abordagem PDH e PDHDA para o projeto de controlador RLQD. Em seguida, na

⋆ Abordagens de Programação Dinâmica Heurística (PDH) e Programação Dinâmica Heurística Dependente de Ação (PDHDA).

Seção 3, realiza-se simulações computacionais para validar e analisar os resultados obtidos com as técnicas propostas. Por fim, na Seção 4, são apresentadas as conclusões do trabalho, destacando os principais *insights*, contribuições e possíveis direções para trabalhos futuros. Este estudo visa contribuir para o avanço do conhecimento em controle ótimo adaptativo, fornecendo uma compreensão mais aprofundada das técnicas de PDH e PDHDA e sua aplicabilidade em sistemas práticos.

2. A DESCRIÇÃO E FORMULAÇÃO MATEMÁTICA DA PLANTA

Os modelos matemáticos podem assumir formas diferentes e, dependendo do sistema de interesse e das circunstâncias envolvidas no processo, um modelo matemático pode ser mais adequado do que outro.

Neste projeto, o modelo matemático é usado para controlar os níveis de água na planta, conforme mostrado nas Figuras 1 a 2. A planta consiste em um sistema de três tanques, onde o tanque primário (01) alimenta dois tanques secundários (02 e 03). O sistema apresentado neste trabalho é classificado como Múltiplas Entradas - Múltiplas Saídas (MIMO) com três entradas representadas por meio das vazões q_1, q_2, q_3 e duas saídas representadas por meio das alturas h_2 e h_3.

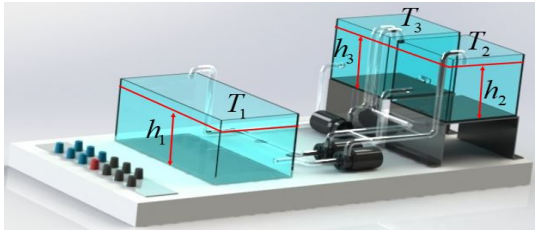

Figura 1. Modelo de uma planta compacta. Nesta figura, as variáveis T_1, T_2 e T_3 representam os tanques 1, 2 e 3. As variáveis h_1, h_2 e h_3 representam as alturas dos tanques 1, 2 e 3, respectivamente.

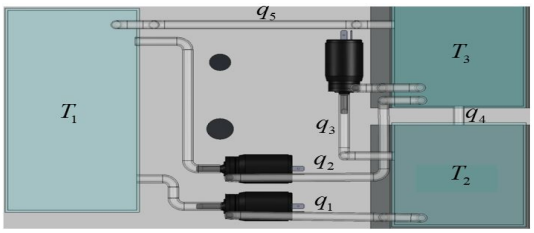

Figura 2. O esquema da planta compacta mostrada na Figura 1 ilustra, além das variáveis já apresentadas, as variáveis que representam as taxas de fluxo dos tanques, que são: q_1, q_2, q_3, q_4 e q_5.

É observado que os tanques 2 e 3 estão no mesmo nível com interação, então q_1 pode ser escrito da seguinte forma:

$$q_4 = \frac{h_2 - h_3}{R_4} \ e \ q_5 = \frac{h_3}{R_5}. \tag{1}$$

A entrada são as taxas de fluxo q_1, q_2 e q_3, e as saídas são as alturas h_1 e h_2. Assim, o sistema pode ser reescrito da seguinte forma:

$$\frac{dh_2}{dt} = \frac{1}{C_2}q_1 - \frac{1}{C_2}q_3 - \frac{1}{C_2R_4}h_2 + \frac{1}{C2R_4}h_3, \tag{2}$$

$$\frac{dh_3}{dt} = \frac{1}{C_3}q_2 - \frac{1}{C_3}q_3 - \left(\frac{R_5 + R_4}{C_3R_5R_4}\right)h_2. \tag{3}$$

A representação matemática em espaço de estado das Eqs. (2) and (3) é dado por

$$\begin{bmatrix}\dot{h}_2\\ \dot{h}_3\end{bmatrix} = \begin{bmatrix}-\frac{1}{C_2R_4} & \frac{1}{C_2R4}\\ \frac{1}{C_3R_4} & \frac{-C_3R_5-C_3R_4}{C_3R_4C_3R_5}\end{bmatrix}\begin{bmatrix}h_2\\ h_3\end{bmatrix} + \begin{bmatrix}\frac{1}{C_2} & 0 & -\frac{1}{C_2}\\ 0 & \frac{1}{C_3} & \frac{1}{C_3}\end{bmatrix}\begin{bmatrix}q_1\\ q_2\\ q_3\end{bmatrix}, \tag{4}$$

sendo R_1, R_2 e R_3 são as resistências dos solenoides das válvulas 1, 2 e 3, respectivamente, C_2 e C_3 são as capacitâncias dos tanques 2 e 3, respectivamente. Em forma reduzida, a equação de espaço de estados (4) e a equação de saída são dadas por

$$\dot{x} = Ax + Bx, \tag{5}$$

$$y = Cx, \tag{6}$$

sendo $y = [h_1, h_2]$, C é a matriz de saída e $x = [h_1, h_2]$ são as variáveis de espaço de estados, A, B, C e D são as matrizes de estado, controle e saída no tempo contínuo.

2.1 Modelo da planta em espaço de estados em tempo contínuo

Para a modelagem em espaço de estados da planta, foram utilizadas as resistências elétricas dos solenoides das válvulas dos tanques, conforme mostrado nas Figuras 1-2. Os parâmetros físicos da planta são definidos como: $R_1 = 1, s/cm^2$ - Resistência da válvula solenóide 1; $R_2 = 0.4, s/cm^2$ - Resistência da válvula solenóide 2; $R_3 = 0.3, s/cm^2$ - Resistência da válvula solenóide 3; $C_2 = 144, cm^2$ - Capacitância ou área de seção transversal do Tanque 2 e $C_3 = 144, cm^2$ - Capacitância ou área de seção transversal do Tanque 3.

Substituindo os parâmetros de resistências e capacitâncias dos tanques na Eq. (4), as matrizes de espaço de estados em tempo contínuo do modelo da planta são dadas por:

$$A = \begin{bmatrix}-0.0174 & 0.0174\\ 0.0174 & -0.0405\end{bmatrix}, \tag{7}$$

$$B = \begin{bmatrix}-0.0069 & 0 & -0.0069\\ 0.0174 & 0.0069 & 0.0069\end{bmatrix}, \tag{8}$$

e

$$C = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \tag{9}$$

2.2 Modelo da planta em espaço de estados em tempo discreto

As matrizes fornecidas nas Eqs. (7)-(9) que representam o modelo de espaço de estados da planta em tempo contínuo são discretizadas pelo método *Zero-Order Hold* (ZOH)

com tempo de amostragem (Ts) igual a 1 segundo e são apresentadas da seguinte forma:

$$A_d = \begin{bmatrix} 0.9351 & 0.0619 \\ 0.0619 & 0.8525 \end{bmatrix}, \tag{10}$$

$$B_d = \begin{bmatrix} -0.0269 & 0.0009 & -0.0260 \\ -0.0009 & -0.0009 & 0.0248 \end{bmatrix}, \tag{11}$$

e

$$C_d = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{12}$$

2.3 Abordagens PDH para o Projeto de Controlador RLQD

Os algoritmos PDH são desenvolvidos com base na abordagem *Recursive Least Squares* (RLS) Astrom and Witternmark (1995) para aproximar a função de valor do estado. Nesta seção, a estimativa RLS e suas variantes são formuladas para aproximar a política de controle DLQR ótima. O método dos mínimos quadrados (RLS) foi formulado por Gauss em 1795 e é atualmente aplicado na solução de problemas de engenharia e ciência.

A formulação do modelo de regressão para o problema de aproximação da equação HJB usando a forma canônica da equação de regressão é dada por:

$$\theta_{i+1} = \theta_i + K^i_{RLS}(d_i - \varphi_i^T)\theta_i, \tag{13}$$

$$K^{i+1}_{RLS} = P^{i+1}_{RLS}\varphi_i(\lambda + \theta_i^T P^{i+1}_{RLS}\varphi_i), \tag{14}$$

$$P^{i+1}_{RLS} = \frac{1}{\lambda}(I - K^{i+1}_{RLS}\varphi_i^T)P^i_{RLS}, \tag{15}$$

sendo θ_i é a variável que representa os parâmetros desconhecidos, ou seja, as soluções da equação HJB. A variável P^i_{RLS} representa a matriz de covariância do processo de busca dos parâmetros θ_i, e os valores de P^{i+1}_{RLS} à direita da igualdade na Eq. (14) indicam que eles são atualizados no lado direito da igualdade na Eq. (15), e a atualização de K^i_{RLS} ocorre de forma semelhante. Os valores observados representam os componentes do vetor de gradiente da função de custo $V(x)$ em relação ao estado.

A estimativa da função de custo V^{K_j} para uma determinada política K_j requer a amostragem da recompensa instantânea r_k e dos estados, assim, modelos do ambiente e da função de utilidade são necessários para calcular a função de custo correspondente à política ótima. Nesse contexto, o método RLS visa encontrar um vetor de parâmetros θ_j para estimar V^{K_j} (Ferreira et al., 2015). Portanto, a estrutura paramétrica da aproximação linear é dada por:

$$V^{K_j}(x_k) = \phi^T(x_k), \tag{16}$$

e deve satisfazer a seguinte condição dada por:

$$V^{K_j}(x_k) = r(x_k,\, h_j(x_k)) + \lambda V^{k_j}(f(x_k,\, h_j(x_k))). \tag{17}$$

A forma quadrática da função de custo é representada como o produto de Kronecker dos estados como uma função da vetorização da matriz P, que é a solução da equação HJB (Santos et al., 2014), Eq. (17). A solução ótima em sua forma vetorizada é dada por:

$$V^{k_j}(x_k) = x_k^T P(x_k) = \bar{x}_k^T vec(P_j), \tag{18}$$

sendo $\bar{x}_k^T \in R^{n(n+1)/2}$ é um vetor que é definido a partir do produto de Kronecker, dado por:

$$\bar{x}_k^T \otimes x_k^T = [x^2_{1,k} \cdots x_{1,k}x_{n,k}\, x^2_{2,k} \cdots x_{n-1,k}x^2_{n,k}], \tag{19}$$

sendo $x_{l,k}$ é o l-ésimo termo do vetor de estado x no tempo k, e $vec_{P_j} \in R^{n(n+1)/2}$ é o resultado da vetorização. Este vetor tem os n elementos de P_j e $n(n+1)/2 - n$ somas não repetidas $p_{rs} + p_{sr}$. Os elementos de x_k e $vecP_j$ são ordenados para satisfazer a forma quadrática da Eq. (18).

O algoritimo PDH-RLS-RLQD é dado por:

Algoritmo 1: PDH-RLS-RLQD Algoritmo

I. Parâmetros de Inicialização
1. A_d, B_d, C_d e D_d;
2. N;
3. Q and R;
4. $\theta_0(0)$;
5. $x(0)$.

II. Processo Iterativo
1. Para $N = i$, passo 1:
2. Esforço de Controle:
   2.1 $u_k = K_{HDP} \times x_k$.
3. Estados:
   3.1 $x_{k+1} = A_d x_k + B_d u_k$.
4. Atualização do Fator de Desconto γ:
   4.1 $\gamma = \gamma^k$.
5. Montagem do vetor desejado:
   5.1 $V^{k_j}(xk) = x_k^T Q x_k + u_k^T R u_k + \gamma(x_{k+1}^T P x_{k+1})$.
6. Produto $Kronecker$:
   6.1 $\varphi^{k+1} = x_k^T \otimes x_k^T$.
7. Erro RLS:
   7.1 $\varepsilon = V^{k_j}(x_k) - \varphi^T\theta_k$.
8. Ganho RLS:
   8.1 $K^{k+1}_{RLS} = P^{k+1}_{RLS}\varphi(\lambda + \varphi_i^T P^{k+1}_{RLS}\varphi_i)^T$.
9. Vetor de parâmetros RLS:
   9.1 $\theta_{k+1} = \theta_k + K^{k+1}_{RLS}\varepsilon$.
10. Matriz de covariância RLS:
   10.1 $P^{k+1}_{RLS} = (P^k_{RLS} - (K^{k+1}_{RLS}\varphi^{k+1}P^k_{RLS}))/\lambda$.
11. P_{PDH} e K_{PDH} matrices:
   11.1 $P^{k+1}_{PDH} = (f(\varphi_{k+1}))$;
   11.2 $K^{k+1}_{RLS} = (R + B_d^T P^{k+1}_{PDH})^{-1}(B_d^T P^{k+1}_{PDH} A_d)$.
12. Fim do Processo Iterativo.

2.4 Abordagens PDH e PDHDA para o Projeto de Controlador RLQD

A PDHDA é uma técnica muito útil de Programação Dinâmica Aproximada (ADP) para o controle de sistemas (Bellman, 1958), (Lendaris, 2009) e (Liu et al., 2014). Também conhecido como Q-learning, é usado para estimar a função de qualidade, comumente conhecida como função Q ou função valor de ação, para qualquer política, ótima ou não.

A função Q é a soma dos custos instantâneos sujeitos à ação u a partir de um estado x, multiplicados pelo custo total que se acumularia se a política fixa fosse seguida a partir desse estado (Guo et al., 2004), conforme a Eq. (20).

$$Q_f(x_k, u_k) = r(x_k, u_k) + \lambda V(x_k + 1). \tag{20}$$

O *feedback* Kx_k é inicialmente calculado para que a função Q possa ser atualizada (Jang et al., 2019). A partir da Eq. (20), para um sistema linear com custo quadrático, a função Q também será quadrática (Gupta et al., 2015), conforme apresentada na equação a seguir:

$$Q(x_k, u_k) = r(x_k, u_k) + V(x_{k+1}) =$$
$$\begin{bmatrix} x_k^T & u_k^T \end{bmatrix} \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} x_k^T \\ u_k^T \end{bmatrix} + x_{k+1}^T P x_{k+1} =$$
$$\begin{bmatrix} x_k^T & u_k^T \end{bmatrix} G \begin{bmatrix} x_k^T \\ u_k^T \end{bmatrix} + (Ax_k + Bu_k)^T P (Ax_k + Bu_k) =$$
$$\begin{bmatrix} x_k^T & u_k^T \end{bmatrix} G + \begin{bmatrix} A^T \\ B^T \end{bmatrix} P \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix}.$$

Resolvendo as expressões algébricas acima, obtém-se a seguinte equação:

$$Q(x_k, u_k) = \begin{bmatrix} x_k^T & u_k^T \end{bmatrix} H \begin{bmatrix} x_k & u_k \end{bmatrix}, \tag{21}$$

sendo $G = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}$.

G é uma matriz diagonal por blocos com blocos Q e R, que são matrizes de ponderação de estado e de entrada, respectivamente, e a matriz H é uma matriz que combina os vetores x_k e u_k (que representam o estado e a entrada, respectivamente).

A função Q pode ser calculada diretamente pela equação DLQR, que é quadrática em x_k e u_k, e é dada por

$$Q_f = (x_k, u_k) = \begin{bmatrix} x_k^T & u_k^T \end{bmatrix} \begin{bmatrix} H_{xx} & H_{xu} \\ H_{ux} & H_{uu} \end{bmatrix} \Rightarrow$$
$$Q_f = \begin{bmatrix} x_k^T & u_k^T \end{bmatrix} H_u \begin{bmatrix} x_k \\ u_k \end{bmatrix}, \tag{22}$$

sendo $G = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}$.

G é uma matriz diagonal com blocos Q e R, que são matrizes de ponderação de estado e de entrada, respectivamente, e a matriz H é uma matriz de núcleo. A matriz H é dada por:

$$H = \left[G + \begin{bmatrix} A^T \\ B^T \end{bmatrix} P \begin{bmatrix} A & B \end{bmatrix} \right], \tag{23}$$

sendo H_u uma matriz simétrica positiva, com quatro blocos que são: $H_{xx} = Q + A^T PA$, $H_{ux} = A^T PB$, $H_{xu} = B^T PA$ e $H_{uu} = R + B^T PB$.

O bloco H_{uu} resulta em uma matriz simétrica definida positiva. Dada uma função de custo e uma política para qualquer intervalo de tempo, uma política melhorada é determinada pela seguinte equação:

$$K_{x+1} = \arg\min_u Q - f(x_k, u_k). \tag{24}$$

Em seguida, um valor mínimo de u_k pode ser obtido tomando a derivada parcial de $Q_f(x_k, u_k)$ em relação a u_k, igualando-a a zero, conforme equação abaixo:

$$\frac{\delta}{\delta u_k} Q_f(x_k, u_k) = 0. \tag{25}$$

Substituindo o valor de $Q_f(x_k, u_k)$ da Eq. (22) em (25), a ação de controle pode ser determinada como:

$$0 = H_{ux} x_k + H_{uu} u_k, \tag{26}$$

$$u_k = (-H_{uu})^{-1} H_{ux} x_k. \tag{27}$$

De acordo com a Eq. (24), $K_{k+1} = (-H_{uu})^{-1} H_{ux}$ é a política de minimização para todos os x_k.

Algoritimo 2: PDHDA-RLS-RLQD Algoritimo

I. parâmetros de Inicialização
1. A_d, B_d, C_d and D_d;
2. N;
3. Q and R;
4. $\theta_0(0)$;
5. $x(0)$ and x_{rev};
6. G and H_{AD};
7. $H_{ux}0$, $H_{uu}0$ and $K_{AD}(0)$.

II. Processo Iterativo
1. Para $N = i$, passo 1.
2. Perturbações:
   2.1 w_k.
3. Esforços de Controle
   3.1 $u_k = K_{PDHDA} \times x_k$.
4. Estados:
   4.1 $x_{k+1} = A_d x_k + B_d u_k$.
5. Atualização do Fator de Desconto γ:
   5.1 $\gamma = \gamma^k$.
6. Montagem do vetor desejado:
   6.1 $$V(x_k) = x_k^T Q x_k + u_k^T R u_k + y + \left(\begin{bmatrix} x \\ -K_{PDHDA} \times x \end{bmatrix}^T \begin{bmatrix} x \\ -K_{PDHDA} \times x \end{bmatrix} \right).$$
7. Produto *Kronecker*:
   7.1 $\varphi^{k+1} = [\lambda \ u] \otimes [x \ u]$.
8. Erro RLS:
   8.1 $\xi = V(x_k) - \varphi^T \theta_k$.
9. Ganho RLS:
   9.1 $K_{RLS}^{k+1} = P_{RLS}^{k+1} \varphi_i (\lambda + \varphi_i^T P_{RLS}^{k+1} \varphi_i)^T$.
10. Vetor de Parâmetros RLS:
    10.1 $\theta_{k+1} = \theta_k + K_{RLS}^{k+1} \varepsilon$.
11. Matrizes P_{PDHDA} e K_{PDHDA}:
    11.1 $P_{PDHDA}^{k+1} = (f(\varphi_{k+1})$;
    1.12.2 $K_{RLS}^{k+1} = (R + B_d^T P_{PDHDA}^{k+1})^{-1} B_d^T P_{PDHDA}^{k+1} A_d$.
12. Fim do Processo Iterativo.

3. SIMULAÇÕES COMPUTACIONAIS E ANÁLISE DOS RESULTADOS

Nesta seção, apresenta-se os resultados e análises simulados usando o *software MATLAB* para os métodos propostos neste artigo.

3.1 Avaliação dos objetivos de controle

Os resultados são apresentados e analisados a partir de uma perspectiva de controle, levando em consideração o objetivo alcançado, sem considerar perturbações aleatórias nos estados da planta. Após a implementação do algoritmo PDH e PDHDA, usando a aproximação da função de valor por meio do método RLS, a convergência das soluções DLQR é avaliada. As matrizes de ganho K_{HDP} e K_{Schur} são dadas por

$$K_{PDH} = K_{Schur} = \begin{bmatrix} 0.0035 & 0.0014 \\ -0.0014 & -0.0015 \\ 0.0021 & 0.0001 \end{bmatrix}. \qquad (28)$$

Os valores dos parâmetros da matriz fornecidos na Eq. (28) são os mesmos tanto para KPDH quanto para K_{Schur}, garantindo convergência para todos os parâmetros da matriz. Vale destacar que o K_{Schur} é o ganho de realimentação de estados calculado pelo método Schur.

As matrizes de ganho K_{PDHDA} e K_{Schur} são dadas por:

$$K_{PDHDA} = K_{Schur} = \begin{bmatrix} -0.0037 & -0.0016 \\ 0.0017 & 0.0015 \\ -0.0019 & -0.0002 \end{bmatrix} \qquad (29)$$

Também pode ser observado a partir da matriz dada em (29) que K_{PDHDA} é igual a K_{Schur}, o que mostra que há convergência em todos os parâmetros de K_{PDHDA} com H_{Schur}, que é a solução ótima.

Na Figura 3 mostra as convergências dos estados x_1 e x_2 para PDH e $PDHDA$, onde a similaridade entre os estados x_1 e x_2 de PDH com $PDHDA$ pode ser observada, no entanto, em $PDHDA$ os estados exibem pequenas oscilações.

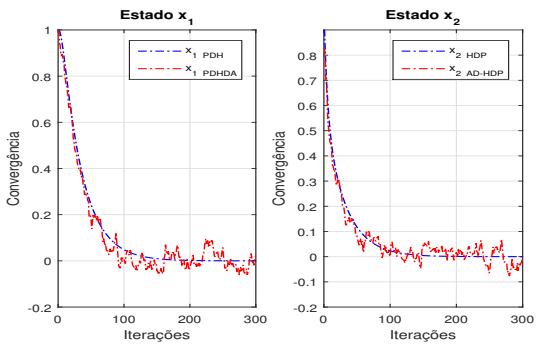

Figura 3. Comportamento dos estados x_1 e x_2 para os métodos PDH e PDHDA.

Nas Figuras 4 e 5 são ilustrados o comportamento das ações de controle para os métodos HDP e PDHDA, respectivamente. Observa-se que as três ações u_1, u_2 e u_3 alcançam o objetivo de controle e permanecem na trajetória sem oscilações significativas em ambos os métodos. No entanto, o método PDH demonstra um desempenho superior. Isso se deve ao fato de que este método é baseado no modelo da planta, permitindo uma adaptação mais precisa do controle às características do sistema. Por outro lado, o método PDHDA opera sem um modelo explícito, o que pode resultar em um ajuste menos eficiente às condições dinâmicas do sistema, dependendo apenas do sinal medido.

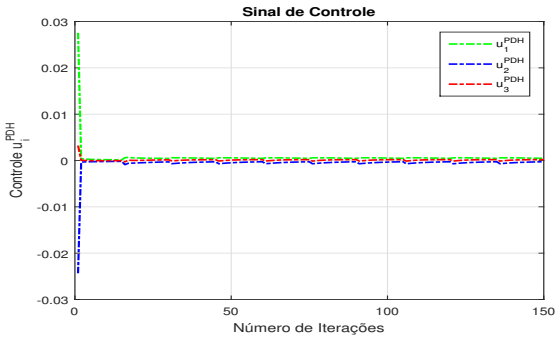

Figura 4. Sinal de Controle u_{PDH}.

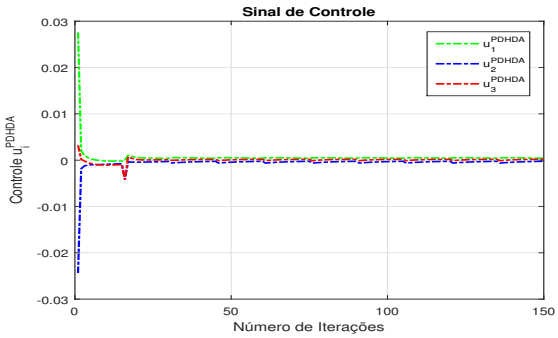

Figura 5. Sinal de Controle u_{PDHDA}.

Na Figura 6 é ilustrado o comportamento do ruído do sinal de controle para os métodos HDP e PDHDA.

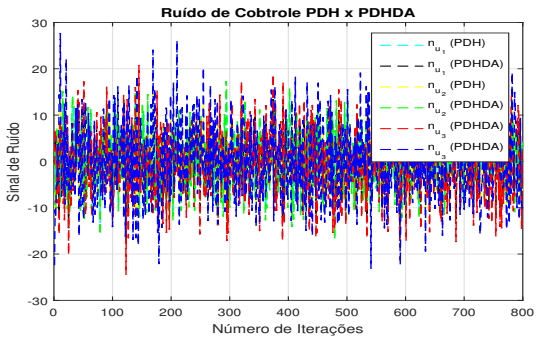

Figura 6. Ruído do Sinal de Controle u_{PDHDA}.

Nas Figuras 7 e 8, apresenta-se o comportamento dos ganhos K usando o RLQD aplicado a PDH e PDHDA em comparação com a solução ótima (Método de Schur). Observa-se que os ganhos K convergiram para a solução ótima em ambas as técnicas utilizadas, garantindo resultados satisfatórios para a planta avaliada.

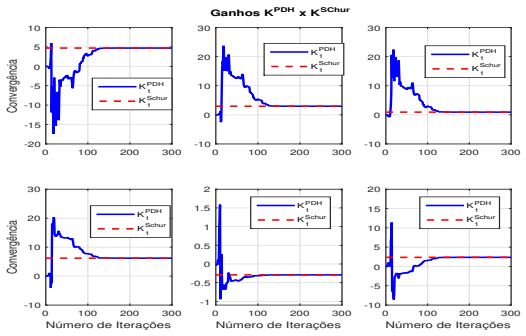

Figura 7. Ganhos K_{Schur}^{PDH}.

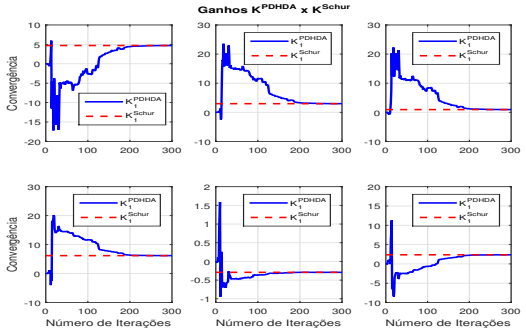

Figura 8. Ganhos K_{Schur}^{PDHDA}.

Por fim, apresenta-se na Figura 9 o comportamento das saídas dos tanques 2 e 3 com as técnicas PDH e PDHDA em comparação com a referência, observa-se, que tanto para PDH quanto para PDHDA as saídas medidas tentam acompanhar o valor de referência, mostrando eficiência das técnicas apresentadas.

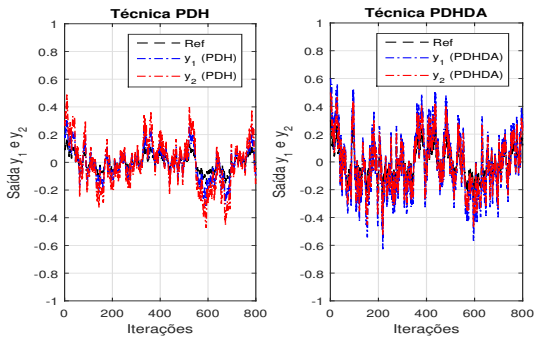

Figura 9. Saídas y_1^{PDH} e y_2^{PDH} x y_1^{PDHDA} e y_2^{PDHDA}.

4. CONCLUSÃO

A conclusão deste trabalho destaca a eficiência das técnicas de PDH e PDHDA no projeto de controladores ótimos adaptativos para sistemas de controle. Ao longo das simulações computacionais, observou-se que ambos os métodos mostraram capacidade de convergir para soluções ótimas, garantindo resultados satisfatórios para a planta avaliada. Os resultados alcançados foram satisfatórios. Portanto, PDH e PDHDA podem ser alternativas para sistemas de controles MIMO. Uma extensão interessante para futuras pesquisas no contexto de PDH e PDHDA é a incorporação de técnicas de aprendizado por reforço, como $Q-learning$, para determinar os ganhos dos controladores PDH e PDHDA.

AGRADECIMENTOS

Agradecemos à Universidade Federal do Maranhão (UFMA) e ao programa de Pós-Graduação de Engenharia Elétrica (PPGEE)da UFMA por viabilizar esta pesquisa. Agradecemos também à Universidade Estadual do Maranhão (UEMA) pela contribuição na realização deste trabalho. Agradecemos. Estamos especialmente gratos à FAPEMA por incentivar a pesquisa de alto nível no Estado do Maranhão. Por fim, agradecemos à CAPES por promover e apoiar pesquisas avançadas que contribuíram para este trabalho.

REFERÊNCIAS

Astrom, K. and Witternmark, B. (1995). Adaptive control, lund institute of technology, sweeden.

Bellman, R. (1958). Dynamic programming and stochastic control processes. *Information and control*, 1(3), 228–239.

Bertsekas, D. (2012). *Dynamic programming and optimal control: Volume I*, volume 1. Athena scientific.

Guo, M., Liu, Y., and Malec, J. (2004). A new q-learning algorithm based on the metropolis criterion. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(5), 2140–2143.

Gupta, N., Kumar, M., Sharma, A., Gaur, M.S., Laxmi, V., Daneshtalab, M., and Ebrahimi, M. (2015). Improved route selection approaches using q-learning framework for 2d nocs. In *Proceedings of the 3rd International Workshop on Many-core Embedded Systems*, 33–40.

Hou, Z. and Jin, S. (2013). *Model free adaptive control*. CRC press Boca Raton, FL, USA:.

Jang, B., Kim, M., Harerimana, G., and Kim, J.W. (2019). Q-learning algorithms: A comprehensive classification and applications. *IEEE access*, 7, 133653–133667.

Lendaris, G.G. (2009). A retrospective on adaptive dynamic programming for control. In *2009 International Joint Conference on Neural Networks*, 1750–1757. IEEE.

Liu, Y., Luo, Y., and Zhang, H. (2014). Adaptive dynamic programming for discrete-time lqr optimal tracking control problems with unknown dynamics. In *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, 1–6. IEEE.

McClement, D.G., Lawrence, N.P., Backström, J.U., Loewen, P.D., Forbes, M.G., and Gopaluni, R.B. (2022). Meta-reinforcement learning for the tuning of pi controllers: An offline approach. *Journal of Process Control*, 118, 139–152.

Uematsu, R., Masuda, S., and Kano, M. (2022). Closed-loop identification of plant and disturbance models based on data-driven generalized minimum variance regulatory control. *Journal of Process Control*, 115, 197–208.

Zhonsheng, H. and Jin, S. (2014). Model free adaptive control theory and application.