Considerações sobre o controle ótimo LQR online com solução da equação algébrica de Riccati baseada em algoritmos de filtragem adaptativa ⋆

Williams Jesús López Yánez *, **Francisco das Chagas de Souza** *

* *Laboratório de Sistemas Adaptativos e Processamento de Sinais-LSAPS, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Maranhão-UFMA, São Luís-MA, Brasil (e-mail:williams.lopez@discente.ufma.br; francisco.souza@ufma.br).*

Abstract: To solve the infinite-horizon optimal control LQR problem, it is necessary to obtain the solution of the discrete algebraic Riccati (DARE) equation. However, the DARE is a nonlinear matrix equation and its solution is usually difficult to find it analytically. In this paper, a discrete-time LQR control design is presented, using adaptive filtering algorithms to approximate the solution of the Bellman equation, which is equivalent to the Lyapunov equation, thus obtaining a sequence of matrices that converges to the DARE solution. Simulations in Matlab® and Simulink® show the performance of the control design when the normalized least-mean-square (NLMS) and recursive least-square (RLS) algorithms are used.

Resumo: Para resolver o problema de controle ótimo LQR com horizonte infinito, é necessário obter a solução da equação algébrica de Riccati discreta (DARE - *discrete algebraic Riccati equation*). No entanto, a DARE é uma equação matricial não linear e sua solução é geralmente difícil de encontrar analiticamente. Neste artigo, é apresentado um projeto de controle LQR em tempo discreto, usando algoritmos de filtragem adaptativa para aproximar a solução da equação de Bellman, que é equivalente à equação de Lyapunov, obtendo assim uma sequência de matrizes que converge para a solução da DARE. Simulações no Matlab® e Simulink® mostram o desempenho do projeto de controle quando os algoritmos NLMS (*normalized least-mean-square*) e RLS (*recursive least-square*) são usados.

Keywords: Adaptive filtering algorithms, Bellman equation, LQR, Lyapunov equation.

Palavras-chaves: Algoritmos de filtragem adaptativa, equação de Bellman, LQR, equação de Lyapunov.

1. INTRODUÇÃO

Neste artigo é apresentado um projeto de controle ótimo discreto LQR *online*, usando algoritmos de filtragem adaptativa na solução da equação algébrica de Riccati discreta (DARE - *discrete algebraic Riccati equation*) (Lewis et al., 2012a, p. 69). O esquema apresentado baseia-se no método de Hewer (1971), onde uma sequência de equações de Lyapunov são resolvidas até convergir para a solução da DARE. Para obter a solução da equação de Lyapunov, é resolvida uma equação equivalente, chamada equação de Bellman (Lewis and Vrabie, 2009).

Para resolver a equação de Bellman, é usado neste trabalho um modelo de regressão linear múltipla (Haykin, 2014, p. 123), onde o regressor (vetor de entrada) depende dos estados do sistema dinâmico, enquanto que a resposta desejada (dados observados) é uma função chamada utilidade, a qual depende dos estados e das ações de controle (Lewis and Vrabie, 2009). Como resultado, obtém-se um método de projeto de controle ótimo LQR *online*, onde não é necessário conhecer o modelo do sistema para resolver a equação de Bellman.

As técnicas de controle ótimo adaptativo, livre do modelo do sistema, pertencem à área da programação dinâmica adaptativa (ADP - *adaptive dynamic programming*) e aprendizagem por reforço (RL - *reinforcement learning*) (Sutton et al., 1992; Khan et al., 2012; Busoniu et al., 2010, 2018; Sutton and Barto, 2018; Rizvi and Lin, 2019; Pang et al., 2019; Pang and Jiang, 2020). Em geral, nas áreas de ADP e RL, a equação de Bellman é resolvida usando redes neurais, métodos do tipo gradiente ou dos mínimos quadrados recursivo (RLS - *recursive least-square*) (Lewis and Vrabie, 2009; Lewis et al., 2012b).

Um trabalho prévio relacionado ao problema do LQR baseado em técnicas de filtragem adaptativa é encontrado em (Silva et al., 2014), onde são usados os algoritmos de filtragem LMS (*least-mean-square*) e PNLMS (*proportional normalized least-mean-square*) junto com a técnica de RL conhecida como HDP (*heuristic dynamic programming*).

⋆ O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), da Fundação de Amparo à Pesquisa e ao Desenvolvimento Científico e Tecnológico do Maranhão (FAPEMA) e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Independentemente das técnicas usadas em RL, a principal contribuição deste trabalho é mostrar como resolver a DARE, *online*, usando técnicas de filtragem adaptativa, e assim resolver o problema de controle ótimo LQR discreto em tempo real. Para conseguir isso, o conjunto dos números inteiros $\mathbb{Z} = \{0, 1, 2, 3, \ldots\}$ é dividido em subconjuntos $[t_j, t_{j+1})$, disjuntos dois a dois, com $j = 0, 1, 2, \ldots$, tal que para todo instante do tempo k no conjunto $[t_j, t_{j+1})$, a solução $\mathbf{S}_j$ de uma equação de Bellman é obtida usando esquemas de filtragem adaptativa, de maneira que $\mathbf{S}_j \to \mathbf{S}^*$ quando $j \to \infty$, sendo $\mathbf{S}^*$ a solução da DARE.

Os algoritmos de adaptação usados neste artigo são o NLMS (*normalized least-mean-square*) e o RLS. Um requesito para obter convergência nos algoritmos NLMS e RLS é a condição de persistência de exitação (Goodwin and Sin, 2009, p. 72). A persistência de exitação é uma condição imposta sobre o sinal de entrada do filtro.

Neste trabalho, para obter a condição de persistência de excitação, é usada a técnica de revitalização de estados (Murray et al., 2002; Al-Tamimi et al., 2007), onde o vetor de estado é revitalizado periodicamente impedindo-o de convergir para o vetor nulo; como consequência, o sinal de entrada do filtro adaptativo não converge para zero.

Para os resultados experimentais, será usado a norma vetorial $\|\mathbf{x}\| = \sqrt{\mathbf{x}^T\mathbf{x}}$, onde $\mathbf{x}$ é um vetor coluna. Se $\mathbf{X}$ é uma matriz, então $\|\mathbf{X}\|$ denota a norma espectral, ou seja, o maior valor singular da matriz $\mathbf{X}$.

2. FORMULAÇÃO DO PROBLEMA

Considere o modelo linear

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \tag{1}$$

onde $\mathbf{x}_k \in \mathbb{R}^n$ é o estado, $\mathbf{u}_k \in \mathbb{R}^m$ é a entrada de controle, $\mathbf{A} \in \mathbb{R}^{n\times n}$ e $\mathbf{B} \in \mathbb{R}^{n\times m}$.

Seja o índice de desempenho associado ao sistema (1)

$$V(\mathbf{x}_k) = \sum_{i=k}^{\infty} \left(\mathbf{x}_i^T\mathbf{Q}\mathbf{x}_i + \mathbf{u}_i^T\mathbf{R}\mathbf{u}_i\right), \tag{2}$$

onde $\mathbf{Q} = \mathbf{Q}^T \geq 0$ e $\mathbf{R} = \mathbf{R}^T > 0$. É assumido que o sistema (1) é estabilizável, ou seja, existe uma lei de controle

$$\mathbf{u}_k = -\mathbf{K}\mathbf{x}_k, \tag{3}$$

com $\mathbf{K} \in \mathbb{R}^{m\times n}$ tal que os autovalores da matriz $\mathbf{A} - \mathbf{B}\mathbf{K}$ pertencem ao círculo unitário (Chen, 1999, p. 131). Nesse caso, diz-se que a matriz $\mathbf{K}$ é admissível. Além disso, é assumido que $(\mathbf{A}, \sqrt{\mathbf{Q}})$ é observável (Chen, 1999, p. 171).

O problema do LQR (Lewis et al., 2012a, p. 32) consiste em encontrar a lei de controle (3), com $\mathbf{K}$ admissível, tal que o índice de desempenho (2) sujeito a (1) seja minimizado.

Para uma lei de controle da forma (3), com $\mathbf{K}$ admissível, o índice (2) é quadrático (Lewis and Vrabie, 2009), isto é,

$$V^{\mathbf{K}}(\mathbf{x}_k) = \mathbf{x}_k^T\mathbf{S}\mathbf{x}_k, \tag{4}$$

onde $\mathbf{S} = \mathbf{S}^T > 0$ é a solução da equação de Lyapunov

$$(\mathbf{A} - \mathbf{B}\mathbf{K})^T\mathbf{S}(\mathbf{A} - \mathbf{B}\mathbf{K}) - \mathbf{S} + \mathbf{Q} + \mathbf{K}^T\mathbf{R}\mathbf{K} = 0. \tag{5}$$

A matriz de ganho ótimo $\mathbf{K}^*$ que minimiza (2) é da forma

$$\mathbf{K}^* = (\mathbf{R} + \mathbf{B}^T\mathbf{S}^*\mathbf{B})^{-1}\mathbf{B}^T\mathbf{S}^*\mathbf{A}, \tag{6}$$

sendo $\mathbf{S}^* = (\mathbf{S}^*)^T > 0$ a solução única da DARE

$$\mathbf{A}^T\mathbf{S}\mathbf{A} - \mathbf{S} + \mathbf{Q} - \mathbf{A}^T\mathbf{S}\mathbf{B}(\mathbf{R} + \mathbf{B}^T\mathbf{S}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{S}\mathbf{A} = 0. \tag{7}$$

Portanto, para resolver o problema do LQR, primeiro é resolvida a DARE e logo com a sua solução $\mathbf{S}^*$ obtém-se a matriz de ganho ótimo (6).

2.1 Controle online

No controle ótimo *online*, a matriz de ganho deve ser calculada em tempo real. Assim, seja J um número inteiro positivo fixo. Para cada $j = 0, 1, 2, \ldots$, considere

$$t_j = jJ. \tag{8}$$

Seja o conjunto $[t_j, t_{j+1})$, com J elementos, definido por

$$[t_j, t_{j+1}) = \{t_j, t_j + 1, t_j + 2, \ldots, t_{j+1} - 1\}. \tag{9}$$

Os conjuntos $[t_j, t_{j+1})$ são disjuntos dois a dois, ou seja

$$[t_j, t_{j+1}) \cap [t_i, t_{i+1}) = \emptyset, \tag{10}$$

para todo $j \neq i$, sendo $\emptyset$ o conjunto vazio. Além disso,

$$\bigcup_{j=0}^{\infty} [t_j, t_{j+1}) = \mathbb{Z}. \tag{11}$$

Neste artigo, é desenvolvido uma lei de controle

$$\mathbf{u}_k = -\mathbf{K}_j\mathbf{x}_k, \tag{12}$$

onde a matriz $\mathbf{K}_j$ é obtida em tempo real, para $k \in [t_j, t_{j+1})$, tal que $\mathbf{K}_j \to \mathbf{K}^*$, quando $j \to \infty$, sendo $\mathbf{K}^*$ a matriz de ganho ótimo (6).

Seja a matriz

$$\mathbf{K}_j = (\mathbf{R} + \mathbf{B}^T\mathbf{S}_{j-1}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{S}_{j-1}\mathbf{A}, \tag{13}$$

com $j \geq 1$, sendo $\mathbf{S}_j$ a solução da equação de Lyapunov

$$\mathbf{S}_j = (\mathbf{A} - \mathbf{B}\mathbf{K}_j)^T\mathbf{S}_j(\mathbf{A} - \mathbf{B}\mathbf{K}_j) + \mathbf{Q} + \mathbf{K}_j^T\mathbf{R}\mathbf{K}_j, \tag{14}$$

com $j \geq 0$ e $\mathbf{K}_0$ admissível. De (4), segue-se que

$$V^{\mathbf{K}_j}(\mathbf{x}_k) = \mathbf{x}_k^T\mathbf{S}_j\mathbf{x}_k. \tag{15}$$

A sequência $\mathbf{S}_j$ (14) converge para a solução da DARE, quando $j \to \infty$, e portanto a sequência de ganho (13) converge para a matriz de ganho ótimo (6) (Hewer, 1971).

Neste trabalho, é mostrado como resolver a equação de Lyapunov (14) *online*, para $k \in [t_j, t_{j+1})$, usando esquemas de filtragem adaptativa, obtendo assim uma sequência de matrizes $\mathbf{S}_j$ em tempo real que converge para a solução da DARE.

3. EQUAÇÃO DE BELLMAN COMO UM PROBLEMA DE FILTRAGEM

Seja a lei de controle (12), onde $\mathbf{K}_j$ é definida em (13). Para $\mathbf{K}_j$, o índice (2) pode ser escrito da forma

$$V^{\mathbf{K}_j}(\mathbf{x}_k) = \mathbf{x}_k^T\mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^T\mathbf{R}\mathbf{u}_k + V^{\mathbf{K}_j}(\mathbf{x}_{k+1}). \tag{16}$$

A equação em (16) é conhecida como equação de Bellman (Lewis and Vrabie, 2009).

A seguir, será mostrado um esquema de filtragem adaptativa para aproximar a solução da equação de Bellman, a qual é equivalente à equação de Lyapunov (14).

Segue-se de (15) que a equação de Bellman (16) para o problema do LQR é

$$\mathbf{x}_k^T\mathbf{S}_j\mathbf{x}_k = \mathbf{x}_k^T\mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^T\mathbf{R}\mathbf{u}_k + \mathbf{x}_{k+1}^T\mathbf{S}_j\mathbf{x}_{k+1}. \tag{17}$$

Seja $\mathbf{A}_{\mathbf{K}_j} = \mathbf{A} - \mathbf{B}\mathbf{K}_j$. Então, (17) é equivalente a

$$\mathbf{x}_k^T \mathbf{S}_j \mathbf{x}_k = \mathbf{x}_k^T \left[\mathbf{Q} + \mathbf{K}_j^T \mathbf{R} \mathbf{K}_j + \mathbf{A}_{\mathbf{K}_j}^T \mathbf{S}_j \mathbf{A}_{\mathbf{K}_j} \right] \mathbf{x}_k. \quad (18)$$

Considerando que (18) deve ser satisfeita para todos os instantes de tempo k, então a equação de Lyapunov (14) é equivalente à equação de Bellman (17); entretanto, a equação de Bellman não depende das matrizes do modelo do sistema e pode ser resolvida *online* usando dados obtidos ao longo das trajetória do sistema.

Observe que a função em (15) pode ser reescrita como

$$V^{\mathbf{K}_j}(\mathbf{x}_k) = \mathbf{s}_j^T \widehat{\mathbf{x}}_k, \quad (19)$$

onde $\widehat{\mathbf{x}}_k$ é o produto de Kronecker (Golub and Van Loan, 2013)

$$\widehat{\mathbf{x}}_k = \mathbf{x}_k \otimes \mathbf{x}_k \quad (20)$$

e $\mathbf{s}_j$ é um vetor formado pelas colunas da matriz $\mathbf{S}_j$, ou seja,

$$\mathbf{s}_j = [\mathbf{S}_{j1}^T \ \mathbf{S}_{j2}^T \ \cdots \ \mathbf{S}_{jn}^T]^T, \quad (21)$$

sendo $\mathbf{S}_{ji}$ a i-ésima coluna da matriz $\mathbf{S}_j$.

Como $\mathbf{S}_j \in \mathbb{R}^{n \times n}$ é simétrica e tem somente $n(n+1)/2$ elementos independentes, então os termos redundantes em $\widehat{\mathbf{x}}_k$ são removidos para definir uma base quadrática $\bar{\mathbf{x}}_k$ com $n(n+1)/2$ elementos definidos como segue.

Para um vetor $\mathbf{x}$ em $\mathbb{R}^n$

$$\mathbf{x} = [x_1 \ \ x_2 \ \cdots \ \ x_n]^T, \quad (22)$$

define-se aqui

$$\bar{\mathbf{x}} = \left[x_1^2 \ \ 2x_1x_2 \cdots 2x_1x_n \ \ x_2^2 \ \ 2x_2x_3 \cdots 2x_{n-1}x_n \ \ x_n^2 \right]^T. \quad (23)$$

Ou seja, $\bar{\mathbf{x}}$ é o resultado da remoção dos termos redundantes em $\widehat{\mathbf{x}} = \mathbf{x} \otimes \mathbf{x}$. Logo, removendo os termos correspondentes, redundantes em $\mathbf{s}_j$, obtém-se $\bar{\mathbf{s}}_j \in \mathbb{R}^{n(n+1)/2}$, onde os elementos em $\bar{\mathbf{s}}_j$ são os elementos da matriz $\mathbf{S}_j$ em (15). A matriz $\mathbf{S}_j$ é chamada reconstrução do vetor $\bar{\mathbf{s}}_j$, enquanto que $\bar{\mathbf{s}}_j$ é chamado vetorização da matriz $\mathbf{S}_j$.

Então, de (19) obtém-se

$$V^{\mathbf{K}_j}(\mathbf{x}_k) = \bar{\mathbf{s}}_j^T \bar{\mathbf{x}}_k. \quad (24)$$

Logo, substituindo (24) em (17), a equação de Bellman torna-se

$$(\bar{\mathbf{x}}_k - \bar{\mathbf{x}}_{k+1})^T \bar{\mathbf{s}}_j = \mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k. \quad (25)$$

Seja o esquema de filtragem adaptativa (Figura 1), onde:

$$\varphi_k = \bar{\mathbf{x}}_k - \bar{\mathbf{x}}_{k+1} \quad \text{(sinal de entrada)} \quad (26a)$$
$$y_k = \varphi_k^T \theta_k \quad \text{(sinal de saída)} \quad (26b)$$
$$d_k = \mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \quad \text{(sinal desejada)} \quad (26c)$$
$$e_k = d_k - y_k \quad \text{(sinal de erro)} \quad (26d)$$

onde θ_k é uma aproximação do vetor $\bar{\mathbf{s}}_j$ no tempo $k \in [t_j, t_{j+1})$.

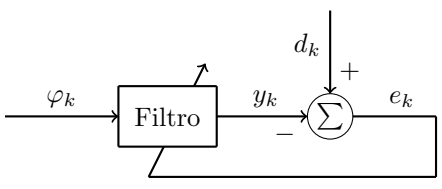


Figura 1. Esquema de filtragem adaptativa para a identificação de $\bar{\mathbf{s}}_j$ em (25).

Como $k \in [t_j, t_{j+1})$, então (25) forma um conjunto de J equações com $n(n+1)/2$ incógnitas, cuja solução pode ser determinada via método de mínimos quadrados. Portanto, para obter uma solução única de mínimos quadrados $\bar{\mathbf{s}}_j$, é necessário e suficiente que a matriz

$$\Phi(t_j) = \sum_{k=t_j}^{t_{j+1}-1} \varphi_k \varphi_k^T \quad (27)$$

seja não singular. Esta condição em (27) de não singularidade é denominada condição de excitação (Åström and Wittenmark, 2008, p. 44).

Uma condição necessária para que a matriz em (27) seja não singular é

$$J \geq n(n+1)/2. \quad (28)$$

3.1 Algoritmo NLMS

A estrutura algorítmica fundamental dos filtros adaptativos para produzir estimativas recursivas θ_k do vetor ótimo $\bar{\mathbf{s}}_j$ é (Bitmead, 1984)

$$\theta_{k+1} = \theta_k + \mu_k f(\theta_k, e_k, \varphi_k) \quad (29)$$

com $\mu_k > 0$. Uma versão de (29) é o algoritmo NLMS (Haykin, 2014, p. 337), chamado também algoritmo de projeção (Goodwin and Sin, 2009, p. 52), (Åström and Wittenmark, 2008, p. 54)

$$\theta_{k+1} = \theta_k + \frac{\mu e_k \varphi_k}{\varphi_k^T \varphi_k + \psi} \quad (30)$$

com $\psi \geq 0$ e $0 < \mu < 2$.

3.2 Persistência de excitação

Para convergência exponencial do algoritmo NLMS (30), é requerido que o sinal φ_k satisfaça a condição de persistência de excitação (Bitmead, 1984): isto é, exista um T fixo, tal que para todo t

$$\infty > \alpha_1 \mathbf{I} > \sum_{k=t}^{t+T} \varphi_k \varphi_k^T > \alpha_2 \mathbf{I} > 0, \quad (31)$$

sendo α_1, α_2 números reais positivos e $\mathbf{I}$ a matriz identidade.

A Tabela 1 mostra o algoritmo para a solução do problema do LQR *online* usando filtragem adaptativa.

Tabela 1: Algoritmo *online* para o problema do LQR.

Entradas: $\mathbf{K}_0$ admissível; $\mathbf{x}_0$; θ_0:

1. **Para** $j = 0, 1, 2, \ldots, N$
2. **Para** $k = t_j, t_j + 1, \ldots, t_{j+1} - 1$
3. $\mathbf{u}_k \leftarrow -\mathbf{K}_j \mathbf{x}_k$
4. $\mathbf{x}_{k+1} \leftarrow \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k$
5. $\varphi_k \leftarrow \bar{\mathbf{x}}_k - \bar{\mathbf{x}}_{k+1}$
6. $d_k \leftarrow \mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k$
7. $\theta_{k+1} \leftarrow \theta_k + f(e_k, \varphi_k, \theta_k)$ como em (30)
8. **Fim**
9. $\mathbf{S}_j \leftarrow$ reconstrução de $\theta_{t_{j+1}}$
10. $\mathbf{K}_{j+1} \leftarrow (\mathbf{R} + \mathbf{B}^T \mathbf{S}_j \mathbf{B})^{-1} \mathbf{B}^T \mathbf{S}_j \mathbf{A}$
11. **Fim**

4. CONVERGÊNCIA DO ALGORITMO ADAPTATIVO PARA O PROBLEMA DO LQR

No algoritmo da Tabela 1, para obter a solução da equação de Bellman, o sinal φ_k tem que satisfazer (31). Essa

condição garante que a adaptação (30), para $k \geq 0$, converge para a solução de mínimos quadrados $\bar{\mathbf{s}}_j$. Nesse caso, para cada $j = 0, 1, 2, \ldots, N$, o vetor de parâmetros $\theta_{t_{j+1}}$, na Tabela 1, é uma aproximação de $\bar{\mathbf{s}}_j$, sendo $\bar{\mathbf{s}}_j$ a vetorização da solução da equação de Lyapunov (14). Portanto, o método em Hewer (1971) garante que a sequência $\mathbf{S}_j$ na Tabela 1 converge para a solução da DARE quando $N \to \infty$.

Neste artigo, para garantir a condição de persistência de excitação (31), será utilizada a técnica conhecida como revitalização de estados (Murray et al., 2002; Al-Tamimi et al., 2007).

5. EXPERIMENTO NUMÉRICO

Nesta seção, serão realizadas simulações do algoritmo apresentado na Tabela 1 para as matrizes em Ten Hagen and Kröse (1998), onde

$$\mathbf{A} = \begin{bmatrix} -0.6 & -0.4 \\ 1.0 & 0.0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix}. \tag{32}$$

Sem perda de generalidade, são escolhidas as matrizes

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{R} = [1]. \tag{33}$$

Nesse caso, a solução $\mathbf{S}^*$ da DARE (7) e a matriz de ganho ótimo $\mathbf{K}^*$ (6) são, respectivamente,

$$\mathbf{S}^* = \begin{bmatrix} 2.0791 & 0.4108 \\ 0.4108 & 1.3210 \end{bmatrix}, \quad \mathbf{K}^* = [0.4630 \;\; -0.0708]. \tag{34}$$

Uma escolha diferente de matrizes identidades em (33), somente afeta o sinal desejada (26c) e o sinal de erro (26d), obtendo assim soluções diferentes das matrizes $\mathbf{S}^*$ e $\mathbf{K}^*$.

O estado inicial do sistema é $\mathbf{x}_0 = [1 \;\; 1]^T$. Seja a matriz inicial admissível $\mathbf{K}_0 = [0.2 \;\; -0.2]$ e o vetor de parâmetros inicial $\theta_0 = [0 \;\; 0 \;\; 0]^T$ na Tabela 1.

5.1 Exemplo 1

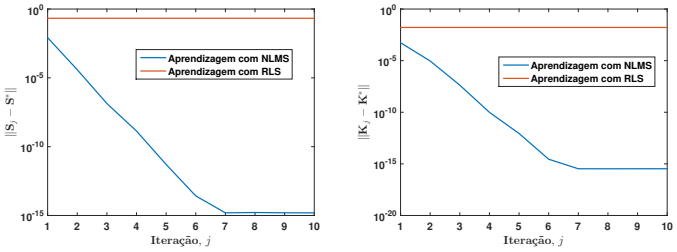
Dois cenários são testados:

- Caso (i). $J = 20, N = 10$, sem revitalização de estados.
- Caso (ii). $J = 20, N = 10$, com revitalização de estados.

Como $n = 2$ (número de estados da planta), então $J = 20$ satisfaz a condição (28). Portanto, em 20 iterações do algoritmo adaptativo, obtém-se uma aproximação da solução da equação de Bellman. Enquanto que $N = 10$ é uma quantidade de iterações suficiente para obter convergência das matrizes $\mathbf{S}_j$ e $\mathbf{K}_j$ na Tabela 1.

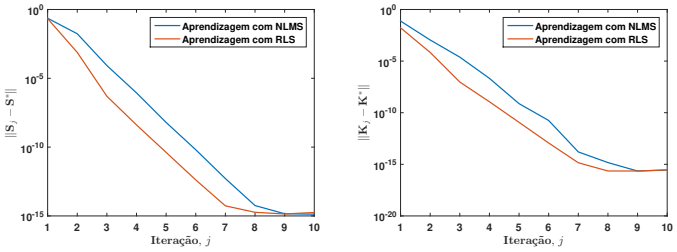
Para os cenários (i) e (ii), simulações serão realizadas usando o algoritmo NLMS e o algoritmo padrão RLS (Åström and Wittenmark, 2008, p. 51) na etapa 7 da Tabela 1.

Caso (i). A Figura 2 (a) e Figura 2 (b) mostram que as sequências $\mathbf{S}_j$ e $\mathbf{K}_j$ convergem para a solução da DARE (7) e para a matriz de ganho ótimo (6), respectivamente, quando é usado o algoritmo NLMS. Enquanto que o algoritmo RLS não consegue adaptar os parâmetros de maneira adequada. A Figura 4 (a) mostra o comportamento dos



(a) (b)

Figura 2. Caso (i). (a) Comportamento de $\|\mathbf{S}_j - \mathbf{S}^*\|$ e (b) comportamento de $\|\mathbf{K}_j - \mathbf{K}^*\|$ verificando covergência das matrizes $\mathbf{S}_j$ e $\mathbf{K}_j$ durante o proceso de aprendizagem. Nesse caso, o algoritmo RLS não consegue adaptar os parâmetros de maneira adequada.



(a) (b)

Figura 3. Caso (ii). (a) Comportamento de $\|\mathbf{S}_j - \mathbf{S}^*\|$ e (b) comportamento de $\|\mathbf{K}_j - \mathbf{K}^*\|$ verificando covergência das matrizes $\mathbf{S}_j$ e $\mathbf{K}_j$ durante o proceso de aprendizagem quando os estados são redefinidos a cada cinco passos da trajetória.

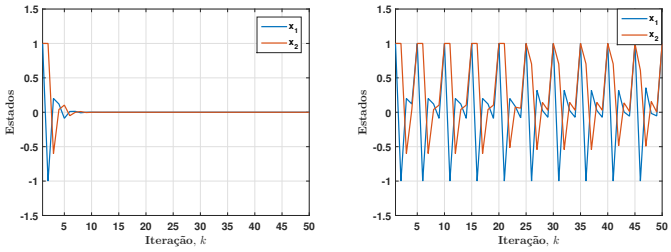
estados durante o processo de aprendizagem usando o algoritmo NLMS.

Caso (ii). Visando evitar convergência de φ_k em (26a) para o vetor nulo, os estados são revitalizados a cada cinco passos da trajetória, ou seja $\mathbf{x}_k = \mathbf{x}_0$ para cada k múltiplo de 5. A justificativa para isso é que a partir do estado $\mathbf{x}_5$, os estados já estão perto do vetor nulo (Figura 4 (a)). A Figura 4 (b) mostra o comportamento dos estados quando são revitalizados. A Figura 3 (a) e a Figura 3 (b) mostram que as sequências $\mathbf{S}_j$ e $\mathbf{K}_j$, as quais convergem para a solução da DARE (7) e a matriz de ganho ótimo (6), respectivamente, quando são usados ambos os algoritmos NLMS e RLS.

5.2 Exemplo 2

Visando ilustrar como o controle ótimo LQR *online* desenvolvido neste artigo pode ser implementado na plataforma de simulação de Simulink®, nesta seção, os blocos principais do projeto de controle são mostrados.

O cenário apresentado é $J = 1$ e $N = 200$. Para $J = 1$, a matriz em (27) tem posto 1 e portanto é singular. Então, somente em uma iteração do algoritmo adaptativo, obtém-se uma aproximação da solução da equação de Bellman. Ou seja, a matriz de ganho está sendo atualizada em cada instante do tempo.



(a) (b)

Figura 4. (a) Trajetória dos estados sem revitalização de estados e aprendizagem NLMS no Caso (i). (b) Trajetória dos estados com revitalização de estados e aprendizagem NLMS no Caso (ii).

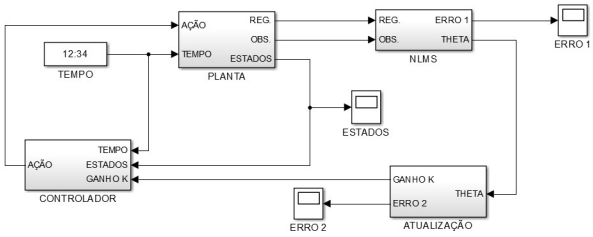


Figura 5. Modelo em Simulink® para o problema do LQR em tempo real.

A Figura 5 mostra o bloco principal do projeto. No tempo $k+1$, a **planta** gera o vetor REG e o escalar OBS como em (26a) e (26c), respectivamente. Logo, o bloco **NLMS** usa os dados REG e OBS para produzir um vetor THETA (que é uma aproximação dos parâmetros da solução da DARE), onde **erro 1** é a norma da diferença do vetor THETA com a solução da DARE. Em seguida, o bloco **atualização**, atualiza a matriz de ganho como em (13) que é enviada ao **controlador**, onde **erro 2** é a norma da diferença da matriz de ganho atual com a matriz de ganho ótimo. O bloco **tempo** é usado pela **planta** para reinicializar os estados periodicamente (revitalização de estados), enquanto que o **controlador** usa o **tempo** para atualizar a matriz de ganho.

A Figura 6 mostra o bloco **planta** do modelo. As entradas são a ação de controle (AÇÃO) e o **tempo**. Os blocos **base quadrática** e **base quadrática 1** calculam $\bar{\mathbf{x}}_k$ e $\bar{\mathbf{x}}_{k+1}$ como em (23), respectivamente. Logo, o vetor de saída REG é calculado como em (26a). A saída OBS é calculada como em (26c). O diagrama de blocos da **revitalização** é mostrado na Figura 7, onde o bloco **revita** revitaliza os estados periodicamente, evitando convergência do vetor REG para o vetor nulo. A saída ESTADOS é o estado atual da planta.

O diagrama de blocos **NLMS** é mostrado na Figura 8, a qual tem como entradas o vetor REG e o escalar OBS. A saída THETA é o parâmetro atual θ_k gerado como em (30). A saída ERRO 1 é igual a $\|\theta_k - \theta^*\|$ que é calculado pelo bloco **norma**, sendo θ^* os parâmetros da solução da DARE.

O diagrama de blocos da **atualização** é mostrado na Figura 9. O bloco **reconstrução** reconstrói a matriz $\mathbf{S}_k$ a partir dos parâmetros THETA. Logo, o bloco **K** atualiza a matriz

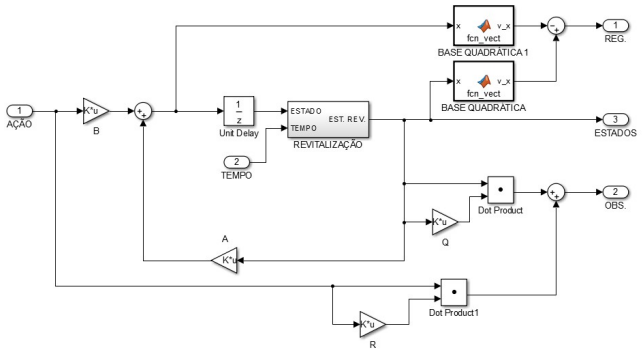


Figura 6. Diagrama de blocos da **planta**

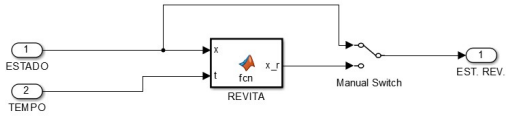


Figura 7. Diagrama de blocos da **revitalização**.

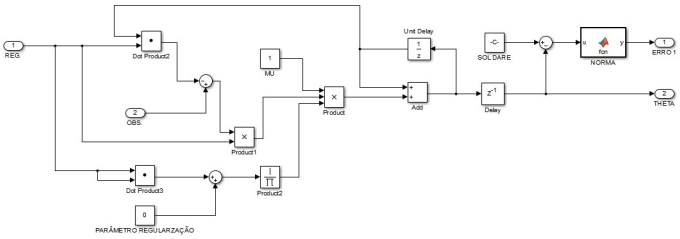


Figura 8. Diagrama de blocos do algoritmo **NLMS**.

de ganho $\mathbf{K}_k$ como em (13) no instante do tempo k. Em seguida, a bloco **norma** calcula $\|\mathbf{K}_k - \mathbf{K}^*\|$.

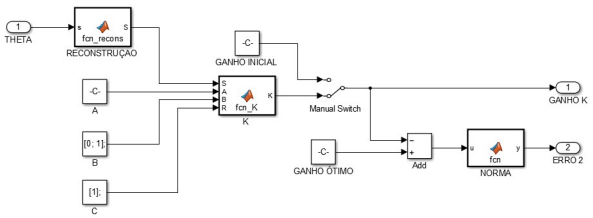


Figura 9. Diagrama de blocos de **atualização**.

O diagrama de blocos do **controlador** é mostrado na Figura 10. Em cada instante do tempo, o **controlador** muda sua matriz de ganho que é recebida desde o bloco **atualização**. A saída do **controlador** é a ação de controle que será aplicada na **planta**.

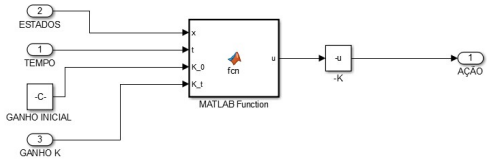
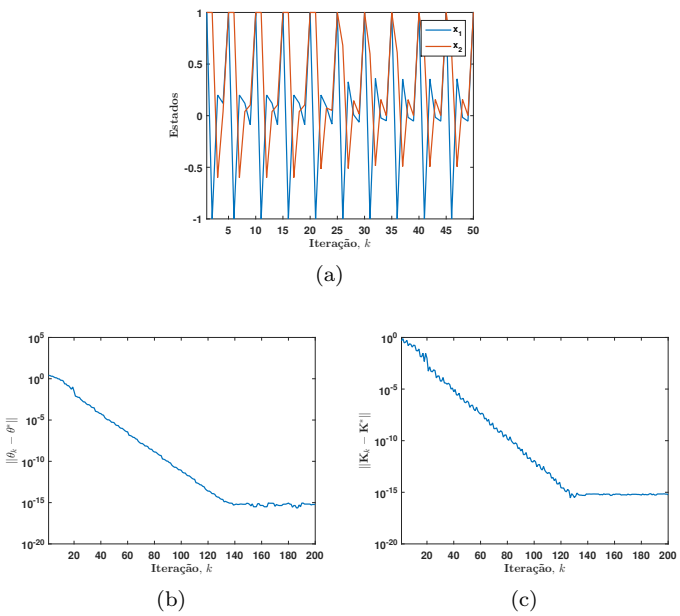


Figura 10. Diagrama de blocos do **controlador**.

O número total de iterações é 200, usando revitalização de estados. O controlador atualiza a matriz de ganho $\mathbf{K}_k$ em cada instante do tempo k até convergir para a matriz de ganho ótimo $\mathbf{K}^*$ (Figura 11 (c)). A Figura 11



(a)

(b) (c)

Figura 11. Simulação usando o modelo do Simulink®. (a) Trajetória dos estados sendo revitalizados. (b) Comportamento de $\|\theta_k - \theta^*\|$ em cada instante do tempo k verificando covergência dos parâmetros θ_k. (c) comportamento de $\|\mathbf{K}_k - \mathbf{K}^*\|$ em cada instante do tempo k verificando covergência das matrizes $\mathbf{K}_k$.

(b) mostra o comportamento de $\|\theta_k - \theta^*\|$, sendo θ^* os parâmetros da solução da DARE. A Figura 11 (a) mostra o comportamento dos estados sendo revitalizados quando k é um múltiplo de 5.

6. CONCLUSÃO

Neste artigo, foi apresentada uma técnica de filtragem adaptativa para resolver o problema de cotrole ótimo discreto LQR *online*. O esquema adaptativo apresentado resolve uma sequência de equações de Bellman em tempo real, até obter a solução da equação algébrica de Riccati discreta. Os algoritmos de filtragem adaptativa usados foram o algoritmo NLMS e o algoritmo RLS.

REFERÊNCIAS

Al-Tamimi, A., Abu-Khalaf, M., and Lewis, F.L. (2007). Adaptive critic designs for discrete-time zero-sum games with application to H_∞ control. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(1), 240–247.

Åström, K.J. and Wittenmark, B. (2008). *Adaptive control*. Dover Publications, second edition.

Bitmead, R. (1984). Persistence of excitation conditions and the convergence of adaptive schemes. *IEEE Trans. Inform. Theory*, 30(2), 183–191.

Busoniu, L., de Bruin, T., Tolic, D., Kober, J., and Palunko, I. (2018). Reinforcement learning for control: Performance, stability, and approximators. *Annual Reviews in Control*, 46, 8–28.

Busoniu, L., Babuska, R., De Schutter, B., and Ernst, D. (2010). *Reinforcement learning and dynamic programming using function approximators*. Taylor & Francis Group.

Chen, C.T. (1999). *Linear System Theory and Design*. Oxford University Press, Inc.

Golub, G.H. and Van Loan, C.F. (2013). *Matrix computations*, volume 3. JHU press.

Goodwin, G.C. and Sin, K.S. (2009). *Adaptive filtering prediction and control*. Dover Publications.

Haykin, S. (2014). *Adaptive filter theory*. Pearson, fifth edition.

Hewer, G. (1971). An iterative technique for the computation of the steady state gains for the discrete optimal regulator. *IEEE Trans. Autom. Control*, 16 (4), 382–384.

Khan, S.G., Herrmann, G., Lewis, F.L., Pipe, T., and Melhuish, C. (2012). Reinforcement learning and optimal adaptive control: An overview and implementation examples. *Annual reviews in control*, 36(1), 42–59.

Lewis, F.L. and Vrabie, D. (2009). Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits Syst. Mag.*, 9 (3), 32–50.

Lewis, F.L., Vrabie, D., and Syrmos, V. (2012a). *Optimal Control*. John Wiley & Sons, Inc., third edition.

Lewis, F.L., Vrabie, D., and Vamvoudakis, K.G. (2012b). Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Syst. Mag.*, 32(6), 76–105.

Murray, J.J., Cox, C.J., Lendaris, G.G., and Saeks, R. (2002). Adaptive dynamic programming. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 32(2), 140–153.

Pang, B., Bian, T., and Jiang, Z.P. (2019). Adaptive dynamic programming for finite-horizon optimal control of linear time-varying discrete-time systems. *Control Theory and Technology*, 17(1), 73–84.

Pang, B. and Jiang, Z.P. (2020). Robust reinforcement learning: A case study in linear quadratic regulation. *arXiv preprint arXiv:2008.11592*.

Rizvi, S.A.A. and Lin, Z. (2019). Output feedback reinforcement learning control for the continuous-time linear quadratic regulator problem. *IEEE Transactions on Neural Networks and Learning Systems*, 30(5), 1523–1536.

Silva, M.E.G., da Fonseca Neto, J.V., and de Souza, F.d.C. (2014). PNLMS-based algorithm for online approximated solution of HJB equation in the context of discrete mimo optimal control and reinforcement learning. In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, 69–76. IEEE.

Sutton, R.S. and Barto, A.G. (2018). *Reinforcement Learning, An Introduction*. Cambridge, MA: MIT Press, second edition edition.

Sutton, R.S., Barto, A.G., and Williams, R.J. (1992). Reinforcement learning is direct adaptive optimal control. *IEEE Control Systems Magazine*, 12(2), 19–22.

Ten Hagen, S. and Kröse, B. (1998). Linear quadratic regulation using reinforcement learning. In *BENELEARN-98, 8th Belgian-Dutch Conference on Machine Learning*, 39–46.