Universidade Federal do Maranhão - UFMA
Programa de Pós-Graduação em Engenharia Elétrica - PPGEE

Tese de Doutorado

Efeito do Ruído de Prova no Controle Ótimo LQR via Q-Learning Baseado em Filtragem Adaptativa

Autor: Williams Jesús López Yánez
Orientador: Prof. Dr. Francisco das Chagas de Souza

São Luís - MA
2022

Williams Jesús López Yánez

Efeito do Ruído de Prova no Controle Ótimo LQR via Q-Learning Baseado em Filtragem Adaptativa

Tese submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Maranhão, como parte dos requisitos necessários à obtenção do grau de Doutor em Engenharia Elétrica na Área de Concentração de Automação e Controle.

Orientador: Prof. Dr. Francisco das Chagas de Souza

São Luís - MA
2022

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Yánez, Williams Jesús López.
Efeito do ruído de prova no controle ótimo LQR via Q-Learning baseado em filtragem adaptativa / Williams Jesús López Yánez. - 2022.
127 p.

Orientador(a): Francisco das Chagas de Souza.
Tese (Doutorado) - Programa de Pós-graduação em Engenharia Elétrica/ccet, Universidade Federal do Maranhão, Universidade Federal do Maranhão, São Luís-MA, 2022.

1. Aprendizagem por reforço. 2. Controle ótimo discreto LQR. 3. Persistência de excitação. 4. Q-learning. 5. Ruído de prova. I. de Souza, Francisco das Chagas. II. Título.

Tesis dedicada a Daniela "La Chiquitota"

Agradecimentos

À Organização dos Estados Americanos (OEA), ao Grupo Coimbra de Universidades Brasileiras (GCUB) e à Universidade Federal do Maranhão (UFMA) pela vaga e bolsa outorgada para fazer esse doutorado.

Agradeço, em especial, ao Professor Vicente Leonardo Paucar Casas pelo importante apoio pessoal desde o tempo de minha chegada a São Luís.

Ao meu orientador, Professor Francisco das Chagas de Souza, por aceitar orientar esta tese e por sua "infinita" paciência no acompanhamento da minha pesquisa.

"Tuve que convencerme a mí mismo de que yo era el mejor antes de que pudiera convencer a los demás"

(Muhammad Ali)

Williams Jesús López Yánez

Efeito do Ruído de Prova no Controle Ótimo LQR via Q-Learning Baseado em Filtragem Adaptativa

Tese aprovada em 13 de maio de 2022.

Prof. Dr. Francisco das Chagas de Souza - UFMA
Orientador

Prof. Dr. João Viana da Fonseca Neto - UFMA
Examinador Interno

Prof. Dr. Ginalber Luiz de Oliveira Serra - IFMA
Examinador Interno

Prof. Dra. Patrícia Helena Moraes Rêgo - UEMA
Examinador Externo

Prof. Dr. Omar Andres Carmona Cortes - IFMA
Examinador Externo

Resumo

Q-learning é um método de aprendizagem por reforço (RL - *reinforcement learning*), livre de modelo, que é usado para resolver o problema de controle ótimo baseado na aprendizagem da função valor de ação (ou função Q). A maneira usual de aprender a função valor de ação é resolver uma equação de Bellman. Nesta tese, para resolver a equação de Bellman no problema de controle ótimo LQR, um algoritmo de filtragem adaptativa baseado no algoritmo de mínimo quadrado médio normalizado (NLMS - *normalized least-mean-squares*) é usado ao invés do algoritmo de mínimos quadrados recursivos (RLS - *recursive least-squares*). Um requerimento geral para obter convergência em algoritmos de filtragem adaptativa é a condição de persistência de excitação. A persistência de excitação é uma condição imposta de maneira que a matriz formada pelos vetores de regressores tenha todas as colunas linearmente independentes. No contexto de controle ótimo via Q-learning, a persistência de excitação é obtida adicionando um ruído de prova na ação de controle. O ruído de prova afeta os estados do sistema real e pode afetar o desempenho do filtro adaptativo na solução da equação de Bellman. Neste trabalho, realiza-se um estudo sobre o efeito do ruído de prova baseado nas matrizes de covariância dos estados e entradas de controle do sistema, onde uma fórmula fechada e propriedades de convergência de ditas matrizes são obtidas. Além disso, verifica-se através de experimentos numéricos que o algoritmo NLMS apresenta desempenho superior quando comparado ao algoritmo RLS, nos casos em que o ruído de prova tem pequena variância. O uso do algoritmo NLMS em nossa abordagem apresenta duas vantagens: a primeira, o algoritmo NLMS apresenta menor complexidade computacional quando comparado ao algoritmo RLS; a segunda, para obter a persistência da condição de excitação, pode-se usar ruídos de prova com baixa variância, o que é desejável em aplicações do mundo real.

Palavras-chaves: Aprendizagem por reforço. Controle ótimo discreto LQR. Persistência de excitação. Q-learning. Ruído de prova.

Abstract

Q-learning is a reinforcement learning (RL) method, model-free, that is used to solve the optimal control problem based on learning the action value function (or function Q). The usual way to learn the action value function is to solve a Bellman equation. In this thesis, to solve the Bellman equation in the LQR optimal control problem, an adaptive filtering algorithm based on the normalized least-mean-square (NLMS) algorithm is used instead of the recursive least-squares (RLS). A general requirement for achieving convergence in adaptive filtering algorithms is the excitation persistence condition. The persistence of excitation is a condition imposed so that the matrix formed by the regressor vectors has all columns linearly independent. In the context of optimal control via Q-learning, persistence of excitation is obtained by adding a probe noise to the control action. The probe noise affects real system states and may affect the performance of the adaptive filter in solving the Bellman equation. In this work, a study is carried out on the effect of probe noise based on the covariance matrices of the states and control inputs of the system, where a closed formula and convergence properties of such matrices are obtained. Furthermore, it is verified through numerical experiments that the NLMS algorithm presents superior performance when compared to the RLS algorithm, in cases where the probe noise has small variance. The use of the NLMS algorithm in our approach has two advantages: first, the NLMS algorithm presents lower computational complexity when compared to the RLS algorithm; the second, to obtain the persistence of the excitation condition, one can use probe noises with low variance, which is desirable in real-world applications.

Keywords: Reinforcement learning. Optimal discrete LQR control. Persistence of excitation. Q-learning. Probe noise.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

Segundo Kiumarsi et al. (2018), os dois princípios para resolver problemas de controle ótimo são o princípio do máximo de Pontryagin (PONTRYAGIN et al., 1986, p. 17) e o princípio da programação dinâmica (DP - *dynamic programming*) (BELLMAN, 1957, p. 83). Sussmann e Willems (1997) afirmam que a teoria de controle ótimo nasceu na ex-União Soviética com o trabalho do princípio do máximo de Pontryagin por L.S. Pontryagin e seu grupo de trabalho. O princípio do máximo de Pontryagin fornece uma condição necessária de otimalidade, enquanto a programação dinâmica fornece uma condição suficiente de otimalidade resolvendo uma equação diferencial parcial, para sistemas em tempo contínuo, e uma equação à diferença, para sistemas em tempo discreto, conhecida como equação de Hamilton-Jacobi-Bellman (HJB).

O método clássico de controle ótimo LQR (*Linear Quadratic Regulator*) resolve uma equação de Riccati "para trás no tempo" no qual é necessário o conhecimento do modelo do sistema (LEWIS; VRABIE; SYRMOS, 2012, p. 32). No contexto de controle ótimo LQR, com horizonte infinito, é necessário resolver uma equação algébrica de Riccati dependente das matrizes do modelo do sistema (LEWIS; VRABIE; SYRMOS, 2012, p. 69). Portanto, os métodos baseados na resolução da equação de Riccati (ou algébrica de Riccati), não podem ser implementado em aplicações onde não é conhecido um modelo do sistema e nem podem ser projetados de uma maneira *online*.

Uma maneira de resolver o problema do controle ótimo *online* é o uso do conceito de ator-crítico (ou crítico adaptativo) (AC - *actor/adaptive-critic*) (SUTTON; BARTO, 2018, p. 331), (KHAN et al., 2012; LEWIS; VRABIE, 2009). A abordagem de crítico adaptativo para resolver o problema de controle ótimo também é conhecida como programação dinâmica adaptativa/aproximada (ADP-*adaptive/approximate dynamic programming*) (LEWIS; VRABIE, 2009; LENDARIS, 2009). Uma família de estruturas ADP foi proposta por Werbos (WERBOS, 1974; WERBOS, 1989; WERBOS, 1991; WERBOS, 1992), os quais são métodos de diferenças temporais de aprendizagem por reforço (SUTTON, 1984; SUTTON, 1988), na qual a função valor é atualizada em cada instante do tempo. A família de estruturas ADP de Werbos tem sido usada por outros autores (HUANG; BALAKRISHNAN, 2001; HUANG; BALAKRISHNAN, 2000; KRISHNAKUMAR; NEIDHOEFER, 1997; LENDARIS et al., 2001; LIU; BALAKRISHNAN, 2000; PADHI; BALAKRISHNAN, 1999; PADHI; BALAKISHNAN, 2001).

O conceito de crítico adaptativo é uma combinação de ideias da programação dinâmica e aprendizagem por reforço (RL-*reinforcement learning*) (LENDARIS, 2009; KHAN et al., 2012), que usa uma aproximação da função valor ótima para realizar seu

projeto de controlador. A aprendizagem por reforço, na sua abordagem clássica, é uma forma de aprendizagem de máquina onde um agente interage com o sistema, tentando obter a maior recompensa possível recebida pelo sistema como consequência de suas ações (veja Figura 1) (HAYKIN, 2009, p. 36), (SUTTON; BARTO, 2018, p. 6).

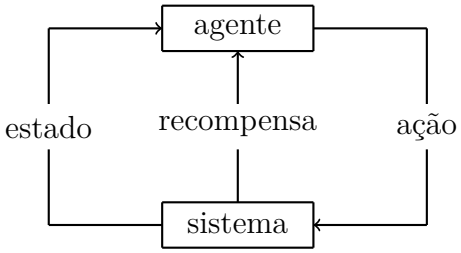


Figura 1 – O agente interage com o sistema tentando obter a maior recompensa possível.

A aprendizagem por reforço, na sua abordagem moderna, baseia-se em programação dinâmica para decidir sobre um curso de ação considerando possíveis estágios futuros sem realmente experimentá-los (HAYKIN, 2009, p. 628).

Busoniu et al. (2010) clasificam os algoritmos de aprendizagem por reforço em três categorias, baseado em como é obtida a política ótima, isto é:

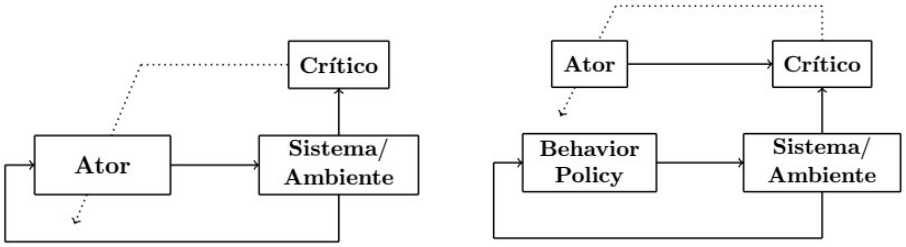
1. **Iteração de política**: os algoritmos de iteração de política avaliam políticas de controle para obter suas funções valor. Logo, tais algoritmos usam estas funções valor para obter uma nova política melhorada.

2. **Iteração de valor**: os algoritmos de iteração de valor aprendem a função valor ótima. Logo, a política ótima de controle é obtida da função valor ótima.

3. **Busca de política**: os algoritmos de busca de política usam diretamente técnicas de otimização para obter a política ótima.

Existem dois tipos de funções valor: a função valor de estado e a função valor de ação (ou função Q). Para obter uma política melhorada da função valor de estado é requerido o modelo do sistema, enquanto que para obter uma política melhorada da função valor de ação não é necessário o modelo do sistema (LEWIS; VRABIE, 2009; KHAN et al., 2012).

A família de estruturas de programação dinâmica adaptativa de Werbos inclui: programação dinâmica heurística (HDP-*heuristic dynamic programming*), programação dinâmica heurística dual (DHP*dual heuristic programming*), programação dinâmica heurística dependente de ação (ADHDP-*action dependent heuristic dynamic programming*) e programação heurística dual dependente de ação (ADDHP-*action dependente dual heuristic programming*) (WERBOS, 1974; WERBOS, 1989; WERBOS, 1991; WERBOS, 1992):



(a) *On-policy* (b) *Off-policy*

Figura 2 – Duas classes de métodos de aprendizagem por reforço.

1. **Estrutura HDP**: o objetivo do crítico é estimar a função valor de estado. Para o treinamento do controlador, é necessário o conhecimento do modelo, enquanto que o treinamento do crítico não precisa do modelo.

2. **Estrutura DHP**: o crítico estima o gradiente da função valor de estado. Nesse caso, para o treinamento do crítico e do controlador, é necessário o modelo do sistema.

3. **Estrutura ADHDP**: o treinamento do controlador é obtido das derivadas, em relação à ação de controle, da função Q. Portanto, a ADHDP não usa o modelo da planta no processo de aprendizagem.

4. **Estrutura ADDHP**: baseia-se na aprendizagem do gradiente da função Q. Em ADDHP, somente é necessário o modelo do sistema para o treinamento do crítico.

Uma estrutura similar a ADHDP de Werbos é o Q-learning de Watkins (WATKINS, 1989; WATKINS; DAYAN, 1992). O Q-learning, proposto por Watkins (1989) na sua tese doutoral, é uma forma de aprendizagem por reforço livre de modelo onde a aprendizagem é similar ao método de diferenças temporais de Sutton (SUTTON, 1984; SUTTON, 1988). O Q-learning de Watkins foi desenvolvido para processos de decisão de Markov em tempo discreto onde o conjunto de estados e ações são finitos. Além disso, o Q-learning proposto por Watkins é um método *off-policy*.

Sutton e Barto (2018, p. 103, 129) classificam os métodos de aprendizagem por reforço em duas classes: *on-policy* e *off-policy*. No método *on-policy*, a política que é usada para gerar os dados (*beahavior policy*) é a mesma política que está sendo melhorada ou avaliada pelo crítico (*target policy*) (Figura 2 (a)). Por outro lado, no método *off-policy* essas duas políticas são separadas e podem ser diferentes (Figura 2 (b)).

Kiumarsi, Lewis e Jiang (2017) afirmam que na sociedade de controle o que é conhecido como Q-learning é o algoritmo SARSA (SUTTON; BARTO, 2018, p. 129). O algoritmo SARSA foi introduzido por Rummery e Niranjan (1994) que é baseado nas ideias de Watkins combinado com o algoritmo de diferenças temporais e o uso do *back-propagation* de redes neurais.

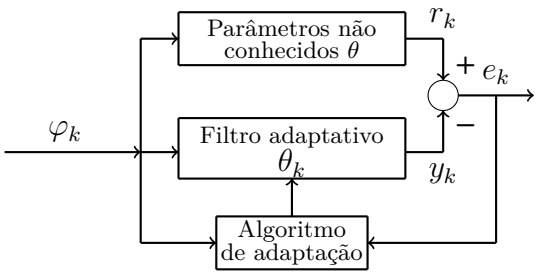


Figura 3 – Esquema de filtragem adaptativa para aproximação dos parâmetros da solução da equação de Bellman.

Bradtke, Ydstie e Barto (1994a) apresentam um algoritmo adaptativo, baseado em Q-learning, para o controle LQR usando iteração de política. O método apresentado usa o algoritmo de mínimos quadrados recursivos (RLS - *recursive least-squares*) para a aprendizagem da função Q. Lewis e Vrabie (2009) descrevem formulações matemáticas para o Q-learning, onde para o aprendizagem da função valor é usado o algoritmo RLS, métodos tipo gradiente ou redes neurais.

1.1 Justificativa, Motivação e Relevância

Q-learning tem sido aplicado em diferentes áreas do conhecimento e atualmente é uma área de pesquisa em aberto (DAS et al., 2020; LI et al., 2019; LIU; MURPHEY, 2020; LOW; ONG; CHEAH, 2019; MU et al., 2019; SAADATMAND et al., 2020; WANG et al., 2020; ZHAO; LEE; CHEN, 2020; SAHOO; VAMVOUDAKIS, 2020).

Em Q-learning, uma estrutura paramétrica (chamada de crítico) aprende a função valor resolvendo uma equação de Bellman *online*, sem o conhecimento do modelo do sistema. Para o problema do LQR, a aprendizagem da função valor usando a equação de Bellman pode ser vista como um problema de filtragem adaptativa linear (Figura 3). A Figura 3 ilustra um esquema de filtragem adaptativa utilizado para aproximar os parâmetros da solução da equação de Bellman, onde o vetor de regressores φ_k contém informação sobre o estado atual $\mathbf{x}_k$ e o próximo estado $\mathbf{x}_{k+1}$, enquanto que r_k é o custo de transição do estado $\mathbf{x}_k$ até o estado $\mathbf{x}_{k+1}$ após ser aplicada a ação de controle $\mathbf{u}_k = \mathbf{K}\mathbf{x}_k$, sendo $\mathbf{K}$ uma matriz de ganho de dimensões apropriadas.

No estudo do comportamento de algoritmos de filtragem adaptativa, uma condição de suficiência para obter convergência é a condição de persistência de excitação (BITMEAD, 1984; GOODWIN; SIN, 2009). Uma maneira de obter a condição de persistência de excitação para aprender a função valor no contexto de Q-learning é adicionando um ruído de prova (*probing noise*) na ação de controle (BRADTKE; YDSTIE; BARTO, 1994b; LEWIS; VRABIE, 2009). O ruído de prova também é conhecido como ruído de exploração (*exploration noise*) (HAGEN; KRÖSE, 1998; AL-TAMIMI; LEWIS; ABU-

KHALAF, 2007; SILVA, 2014; JIANG; JIANG, 2017, p. 14).

Escolher o ruído de prova não é uma tarefa trivial para problemas de RL (JIANG; JIANG, 2017, p. 14). Em problemas práticos, vários tipos de ruído de prova têm sido usado, como por exemplo, ruído aleatório (BRADTKE; YDSTIE; BARTO, 1994b; HAGEN; KRÖSE, 1998; AL-TAMIMI; LEWIS; ABU-KHALAF, 2007), ruído de prova exponencialmente decrescente (VAMVOUDAKIS; LEWIS, 2011), soma de sinusoides com diferentes frequências (JIANG; JIANG, 2011; JIANG; JIANG, 2012; RIZVI; LIN, 2019).

Adicionar um ruído de prova na ação de controle afeta os estados do sistema. Além disso, o ruído de prova afeta o desempenho do algoritmo de adaptação usado no filtro adaptativo, ou seja, o ruído de prova afeta a aprendizagem da função valor.

Visando a aprendizagem da função valor, Bradtke, Ydstie e Barto (1994b) usam um sinal aleatória de distribuição normal para obter a condição de persistência de excitação no contexto de Q-learning baseado no RLS. Não obstante, tais autores não mostram o efeito nos estados em relação à variância; além disso, o método apresentado pode falhar se a variância do sinal aleatória é "pequena".

Hagen e Kröse (1998) usam ruído branco de distribuição normal para aprender a função valor no problema de controle ótimo LQR. No entanto, não mostram o efeito nos estados em relação à variância. Além disso, usam batelada de mínimos quadrados ao invés de algoritmos de estimação de parâmetros em tempo real para identificar a função valor.

No contexto de controle H_∞, Al-Tamimi, Lewis e Abu-Khalaf (2007) mostram claramente com um exemplo que os estados não convergem assintoticamente para zero durante o processo de aprendizagem do controlador devido ao ruído de prova.

Em Kiumarsi, Lewis e Jiang (2017), para obter convergência dos estados para zero, o ruído de prova é desligado depois do processo de aprendizagem do controlador.

Em relação a Q-learning em tempo continuo, em Vamvoudakis, Vrabie e Lewis (2009), a condição de persistência de excitação é removida após um certo tempo para obter convergência dos estados para o vetor zero. Por outro lado, em Vamvoudakis (2017), o ruído de prova é removido após um certo tempo para obter convergência à solução ótima.

Neste trabalho de pesquisa, realiza-se um estudo sobre o efeito do ruído de prova nos estados, e entradas de controle, do sistema baseado na sua matriz de covariância (YÁNEZ; SOUZA, 2022). Além disso, as propriedades de convergência da matriz de covariância do vetor de estado são estabelecidas, considerando as condições de excitação pelo ruído de prova. Também, visando evitar o alto custo computacional e os problemas de instabilidade do algoritmo RLS na aprendizagem da função valor, neste trabalho, um projeto de controlador ótimo adaptativo LQR é proposto baseado em iteração de política e no algoritmo de mínimo quadrado médio normalizado (NLMS - *normalized least-mean-squares*).

1.2 Metodologia

O estudo do efeito do ruído de prova é feito no sentido do cálculo das matrizes de covariância dos estados, as entrada de controle e das saídas do sistema em cada amostragem do tempo e em regime permanente. Além disso, estuda-se de maneira empírica o efeito do ruido de prova no desempenho numérico dos algoritmos de filtragem adaptativa NLMS e RLS na aprendizagem da função valor de ação.

1.3 Objetivos

1.3.1 Objetivo Geral

Estudar o efeito do ruído de prova no problema de controle ótimo LQR para sistemas lineares discretos, invariantes no tempo, via Q-learning e iteração de política.

1.3.2 Objetivos Específicos

1. Demonstrar a necessidade do ruído de prova para a aprendizagem da função valor de ação através da formulação de um novo teorema.

2. Calcular a matriz de covariância em cada tempo de amostragem dos estados e entradas de controle, afetado pelo ruído de prova em função das matrizes do modelo do sistema.

3. Desenvolver um teorema estabelecendo convergência da sequência de matrizes de covariância dos estados, e entrada de controle, afetados pelo ruído de prova.

4. Mostrar numericamente que o algoritmo Q-learning baseado em RLS pode falhar no processo de aprendizagem da função valor de ação quando a variância do ruído de prova é baixa.

5. Desenvolver um controlador, mostrando um processo de filtragem na aprendizagem da função valor para o problema do LQR com realimentação de estado, baseado em Q-learning junto com iteração de política e o algoritmo NLMS.

6. Desenvolver um controlador, mostrando um processo de filtragem na aprendizagem da função valor para o problema do LQR com realimentação de saída, baseado em Q-learning junto com iteração de política e o algoritmo NLMS.

1.4 Contribuições

A principais contribuições neste trabalho são:

1. Caracterização e solução do problema da equação de Lyapunov, livre de modelo, como um esquema de filtragem adaptativa para a aprendizagem da função valor de estado e valor de ação.

2. Estabelecimento de uma fórmula fechada para a matriz de covariância dos estados, e entrada de controle, afetados pelo ruído de prova, em cada tempo de amostragem.

3. Estabelecimento da convergência da sequência de matrizes de covariância dos estados e entrada de controle, que dependem de uma solução da equação de Lyapunov multiplicada pela variância do ruído de prova.

4. Estabelecimento de uma fórmula fechada para a matriz de covariância das saídas afetadas pelo ruído de prova, em cada tempo de amostragem.

5. Estabelecimento da convergência da sequência de matrizes de covariância das saídas, que dependem de uma solução da equação de Lyapunov e a matriz de saída multiplicada pela variância do ruído de prova.

1.5 Publicações

1. Yánez, W. J. L.; Souza, F. d. C. de. On the effect of probing noise in optimal control LQR via Q-learning using adaptive filtering algorithms. *European Journal of Control*, Vol 65, May (2022), DOI: 10.1016/j.ejcon.2022.100633 (QUALIS A2).

2. Yánez, W. J. L.; Souza, F. d. C. de. Considerações sobre o controle ótimo LQR online com solução da equação algébrica de Riccati baseada em algoritmos de filtragem adaptativa. *Simpósio Brasileiro de Automação Inteligente*, Vol 1, No 1 (2021): SBAI2021, https://doi.org/10.20906/sbai.v1i1.2635.

1.6 Organização da Tese

A abordagem da programação dinâmica e aprendizagem por reforço para estados discretos e finitos é apresentada no Capítulo 2. O Capítulo 3 apresenta a teoria de controle ótimo em tempo discreto, com horizonte finito e infinito, e a importância das equações de Riccati no controle LQR. No Capítulo 4, mostra-se a técnica de programação dinâmica, que baseia-se no princípio de otimalidade de Bellman, para a solução do problema de controle ótimo LQR. A técnica de programação dinâmica adaptativa Q-learning, livre de modelo, no contexto do LQR, e a necessidade do ruído de prova, são apresentadas no Capítulo 5. No Capítulo 6, demonstra-se o efeito do ruído de prova inserido no sistema para obter a condição de persistência de excitação, condição necessária para obter convergência dos algoritmos de adaptação do filtro. O Capítulo 7 apresenta o caso do problema do

LQR com realimentação de saída e o efeito do ruído e prova inserido no sistema. Por fim, conclusões e comentários finais são expostos no Capítulo 8.

2 Programação Dinâmica e Aprendizagem por Reforço

O modelo formal por trás dos problemas que resolvem as técnicas de programação dinâmica e aprendizagem por reforço é o processo de decisão de Markov, que é um modelo de decisão sequencial. Um modelo de decisão sequencial é formado por um conjunto de etapas de decisão, um conjunto de estados do sistema, um conjunto de ações disponíveis, um conjunto de recompensas ou custos imediatos dependentes do estado e da ação e um conjunto de probabilidades de transição dependentes do estado e da ação (PUTERMAN, 1994, p. 1).

Um problema de decisão sequencial pode ser representado como na Figura 4, na qual em um determinado tempo, um agente ou controlador observa o estado de um sistema. Com base nesse estado, o agente escolhe uma ação. A escolha da ação produz dois resultados: o agente recebe uma recompensa imediata (ou custo imediato) e o sistema evolui para um novo estado em um ponto subsequente no tempo de acordo com uma distribuição de probabilidade determinada pela escolha da ação. Nesse ponto subsequente no tempo, o agente enfrenta um problema semelhante, mas agora o sistema pode estar em um estado diferente e pode haver um conjunto diferente de ações para escolher.

Problemas de decisão sequencial aparecem em uma grande variedade de campos, incluindo controle automático, inteligência artificial, investigação de operações, economia e medicina (BUSONIU et al., 2010, p. 1).

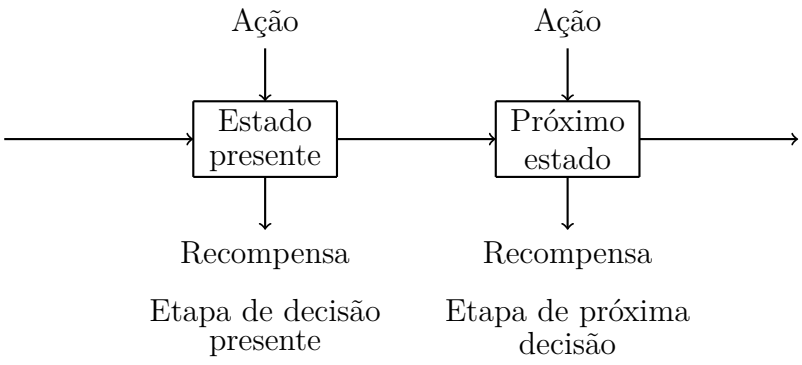


Figura 4 – Problema de decisão sequencial (PUTERMAN, 1994, p. 2). Um agente observa o estado presente e escolhe uma ação produzindo uma recompensa e um próximo estado. No próximo estado o agente enfrenta um problema semelhante.

Uma ferramenta para resolver problemas de decisão sequencial é a programação dinâmica (DP - *Dynamic Programming*), desenvolvida por Bellman (1957). No contexto da teoria de controle, se um modelo do sistema é conhecido, então a programação dinâmica

pode ser usada para resolver o problema de controle ótimo. Entretanto, a programação dinâmica é de utilidade limitada na aprendizagem por reforço, tanto por causa da necessidade de conhecimento do modelo do sistema quanto por causa de seu custo computacional elevado.

A aprendizagem por reforço é um método que consiste em aprender como mapear situações em ações, de maneira a maximizar um sinal de recompensa (SUTTON; BARTO, 2018, p. 1). No contexto da teoria de controle, as situações são os estados, as ações são as entradas de controle e o sinal de recompensa é um custo numérico. Uma vantagem da aprendizagem por reforço é que a mesma pode ser aplicada sem o conhecimento do modelo do sistema (BUSONIU et al., 2010, p. 13)..

Métodos de aprendizagem por reforço *offline* são aplicáveis se os dados são obtidos antecipadamente. Métodos de aprendizagem por reforço *online* aprendem uma solução interagindo com o sistema, e assim podem ser aplicados mesmo quando os dados não são obtidos antecipadamente. Se um modelo do sistema é conhecido, métodos de aprendizagem por reforço também podem ser aplicados simplesmente usando o modelo ao invés do sistema real para gerar os dados.

Os principais elementos dos métodos de aprendizagem por reforço são a política, a função de recompensa, a função valor e o modelo do ambiente (SUTTON; BARTO, 2018, p. 6):

- A política define o comportamento do agente em cada instante de tempo.
- A função de recompensa envia um sinal numérico para o agente por cada ação tomada.
- A função valor é a quantidade total de recompensa que o agente pode esperar no futuro.
- Modelo do sistema (opcional).

2.1 Processos de decisão de Markov

Problemas de programação dinâmica e aprendizagem por reforço podem ser formulados com a ajuda de processos de decisão de Markov (MDP-*Markov decision processes*) (BUSONIU et al., 2010, p. 14). Um MDP é um modelo de decisão sequencial particular, na qual o conjunto de ações disponíveis, as recompensas e as probabilidades de transição, dependem somente do estado atual e da ação atual e não dos estados visitados e ações escolhidas no passado. A seguir, apresenta-se o MDP em um enfoque determinístico e estocástico.

2.1.1 Enfoque determinístico

Um processo de decisão de Markov determinístico é uma tupla $(\mathbf{X}, \mathbf{U}, f, \rho)$ onde:

1. $\mathbf{X}$ é um conjunto de estados
2. $\mathbf{U}$ é um conjunto de ações
3. $f : \mathbf{X} \times \mathbf{U} \longrightarrow \mathbf{X}$ é uma função de transição

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \tag{2.1}$$

4. $\rho : \mathbf{X} \times \mathbf{U} \longrightarrow \mathbb{R}$ é uma função de recompensa tal que

$$\|\rho\|_\infty = \sup\{|\rho(\mathbf{x}_k, \mathbf{u}_k)| \ : \ \mathbf{x}_k \in \mathbf{X}, \mathbf{u}_k \in \mathbf{U}\} < \infty. \tag{2.2}$$

O controlador escolhe ações de acordo com sua política $\mathbf{h} : \mathbf{X} \to \mathbf{U}$

$$\mathbf{u}_k = \mathbf{h}(\mathbf{x}_k). \tag{2.3}$$

Dados f e ρ, o estado atual $\mathbf{x}_k$ e a ação atual $\mathbf{u}_k$ são suficientes para determinar o próximo estado $\mathbf{x}_{k+1}$ e a recompensa $\rho(\mathbf{x}_k, \mathbf{u}_k)$. Essa é a chamada propriedade de Markov (BUSONIU et al., 2010, p. 15).

Alguns MDP têm estados terminais, uma vez atingido, não pode ser mais deixado. Todas as recompensas recebidas em estados terminais são iguais a 0.

Exemplo 2.1.1 *Considere o problema do robô de limpeza (BUSONIU et al., 2010, p. 15) dado na Figura 5: Nesse problema, o estado* $\mathbf{x}$ *descreve a posição do robô, e a ação*

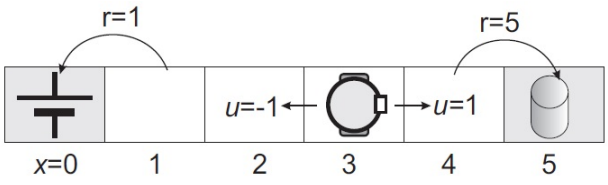


Figura 5 – O problema do robô de limpeza

$\mathbf{u}$ *descreve a direção do seu movimento. Os estado* 0 *e* 5 *são terminais. O conjunto de estados é* $\mathbf{X} = \{0, 1, 2, 3, 4, 5\}$ *e o conjunto de ações é* $\mathbf{U} = \{-1, 1\}$.

A função de transição é:

$$f(\mathbf{x}, \mathbf{u}) = \begin{cases} \mathbf{x} + \mathbf{u} & \text{se} \quad 1 \leq \mathbf{x} \leq 4 \\ \mathbf{x} & \text{se} \quad \mathbf{x} = 0 \text{ ou } \mathbf{x} = 5 \end{cases} \tag{2.4}$$

e a função de recompensa é:

$$\rho(\mathbf{x}, \mathbf{u}) = \begin{cases} 5 & \text{se} \quad \mathbf{x} = 4 \text{ e } \mathbf{u} = 1 \\ 1 & \text{se} \quad \mathbf{x} = 1 \text{ e } \mathbf{u} = -1 \\ 0 & \text{se} \quad \text{outro caso} \end{cases} \tag{2.5}$$

Exemplo 2.1.2 *Seja o modelo linear em espaço de estados*

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \tag{2.6}$$

sendo $\mathbf{x}_k \in \mathbb{R}^n$ *o estado com estado inicial* $\mathbf{x}_0$, $\mathbf{u}_k \in \mathbb{R}^m$ *é a ação de controle. Seja a função de recompensa*

$$-r(\mathbf{x}_k, \mathbf{u}_k) = -(\mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k), \tag{2.7}$$

na qual $\mathbf{Q} = \mathbf{Q}^\top \geq 0$, $\mathbf{R} = \mathbf{R}^\top > 0$ *e*

$$\mathbf{u}_k = -\mathbf{K}\mathbf{x}_k \tag{2.8}$$

com $\mathbf{K} \in \mathbb{R}^{m \times n}$ *tal que os autovalores de* $\mathbf{A} - \mathbf{B}\mathbf{K}$ *têm norma menor a* 1. *Então*

$$|-r(\mathbf{x}_k, \mathbf{u}_k)| = |\mathbf{x}_k^\top (\mathbf{Q} + \mathbf{K}^\top \mathbf{R}\mathbf{K})\mathbf{x}_k| \longrightarrow 0 \tag{2.9}$$

quando $k \longrightarrow \infty$. *Dessa forma, equações (2.6) e (2.7), com* $\mathbf{X} = \mathbb{R}^n$ *e* $\mathbf{U} = \mathbb{R}^m$, *é um MDP.*

Note que a recompensa $-r(\mathbf{x}_k, \mathbf{u}_k)$ em (2.7) é equivalente à função custo $r(\mathbf{x}_k, \mathbf{u}_k)$.

Otimalidade no sentido determinístico

O objetivo em aprendizagem por reforço é encontrar uma política ótima que maximize o alcance para todo estado inicial $\mathbf{x}_0$. O alcance é uma recompensa cumulativa ao longo de uma trajetória começando em $\mathbf{x}_0$.

O alcance é definido como

$$R^{\mathbf{h}}(\mathbf{x}_0) = \sum_{k=0}^{\infty} \gamma^k r_{k+1} = \sum_{k=0}^{\infty} \gamma^k \rho(\mathbf{x}_k, \mathbf{h}(\mathbf{x}_k)) \tag{2.10}$$

no qual $\gamma \in [0, 1)$ é o fator de desconto e $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{h}(\mathbf{x}_k))$ para $k \geq 0$. Matematicamente, o fator de desconto assegura que o alcance é limitado se a recompensa é limitada. Existem outros tipos de alcance. O alcance sem desconto é obtido tomando $\gamma = 1$. Infelizmente, o alcance sem desconto com horizonte infinito é frequentemente não limitado (BUSONIU et al., 2010, p. 16).

Exemplo 2.1.3 *Seja o processo de processo de decisão de Markov como no Exemplo 2.1.2. Seja o alcance sem desconto definido por*

$$R^{\mathbf{K}}(\mathbf{x}_0) = -\sum_{k=0}^{\infty}(\mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k). \tag{2.11}$$

Nesse caso, o alcance é sempre limitado e

$$R^{\mathbf{K}}(\mathbf{x}_0) = -\mathbf{x}_k^\top \mathbf{S}\mathbf{x}_k \tag{2.12}$$

no qual $\mathbf{S}$ *satisfaz a equação de Lyapunov (LEWIS; VRABIE, 2009)*

$$(\mathbf{A}-\mathbf{B}\mathbf{K})^\top \mathbf{S}(\mathbf{A}-\mathbf{B}\mathbf{K}) - \mathbf{S} + \mathbf{Q} + \mathbf{K}^\top \mathbf{R}\mathbf{K} = \mathbf{0}. \tag{2.13}$$

Funções valor e equações de Bellman no sentido determinístico

Existem dois tipos de funções valor: a função valor de estado (ou função-V) e a função valor de ação (ou função-Q).

Seja $(\mathbf{X}, \mathbf{U}, f, \rho)$ um MDP. A função $Q^{\mathbf{h}}$

$$Q^{\mathbf{h}} : \mathbf{X} \times \mathbf{U} \longrightarrow \mathbb{R} \tag{2.14}$$

de uma política

$$\mathbf{h} : \mathbf{X} \longrightarrow \mathbf{U} \tag{2.15}$$

é definida como

$$Q^{\mathbf{h}}(\mathbf{x}, \mathbf{u}) = \rho(\mathbf{x}, \mathbf{u}) + \gamma R^{\mathbf{h}}(f(\mathbf{x}, \mathbf{u})). \tag{2.16}$$

Note que se $(\mathbf{x}_0, \mathbf{u}_0)$=$(\mathbf{x}, \mathbf{u})$, $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{h}(\mathbf{x}_k))$ para $k \geq 0$ e $\mathbf{u}_k = \mathbf{h}(\mathbf{x}_k)$, então (2.16) pode ser reescrita como

$$\begin{aligned} Q^{\mathbf{h}}(\mathbf{x}, \mathbf{u}) &= \rho(\mathbf{x}_0, \mathbf{u}_0) + \gamma R^{\mathbf{h}}(\mathbf{x}_1) && (2.17)\\ &= \rho(\mathbf{x}_0, \mathbf{h}(\mathbf{x}_0)) + \gamma \sum_{k=1}^{\infty} \gamma^{k-1} \rho(\mathbf{x}_k, \mathbf{h}(\mathbf{x}_k)) && (2.18)\\ &= \sum_{k=0}^{\infty} \gamma^{k} \rho(\mathbf{x}_k, \mathbf{h}(\mathbf{x}_k)) && (2.19)\\ &= R^{\mathbf{h}}(\mathbf{x}_0). && (2.20) \end{aligned}$$

A função Q^* ótima é definida como

$$Q^*(\mathbf{x}, \mathbf{u}) = \max_{\mathbf{h}} Q^{\mathbf{h}}(\mathbf{x}, \mathbf{u}). \tag{2.21}$$

Para cada política $\mathbf{h}^*$ tal que

$$\mathbf{h}^*(\mathbf{x}) \in \arg\max_{\mathbf{u}}(Q^*(\mathbf{x}, \mathbf{u})) \tag{2.22}$$

é ótima (ou seja, maximiza o alcance). Em geral, dada uma função $Q(\mathbf{x}, \mathbf{u})$, uma política que satisfaz

$$\mathbf{h}(\mathbf{x}) \in \arg\max_{\mathbf{u}}(Q(\mathbf{x}, \mathbf{u})) \tag{2.23}$$

se diz que é *greedy* em Q. De maneira que, para encontrar uma política ótima, primeiro é encontrado Q^* e logo é usado (2.22) para calcular uma política *greedy* em Q^*.

As funções $Q^{\mathbf{h}}$ e Q^* são caracterizadas recursivamente por equações de *Bellman*. A equação de Bellman para $Q^{\mathbf{h}}$ é definida por

$$Q^{\mathbf{h}}(\mathbf{x}, \mathbf{u}) = \rho(\mathbf{x}, \mathbf{u}) + \gamma Q^{\mathbf{h}}(f(\mathbf{x}, \mathbf{u}), \mathbf{h}(f(\mathbf{x}, \mathbf{u}))). \tag{2.24}$$

A equação de otimalidade de Bellman caracteriza Q^*

$$Q^*(\mathbf{x}, \mathbf{u}) = \rho(\mathbf{x}, \mathbf{u}) + \gamma \max_{\mathbf{u}'} Q^*(f(\mathbf{x}, \mathbf{u}), \mathbf{u}'). \tag{2.25}$$

Por outro lado, a função $V^{\mathbf{h}}$

$$V^{\mathbf{h}} : \mathbf{X} \longrightarrow \mathbf{X} \tag{2.26}$$

de uma política

$$\mathbf{h} : \mathbf{X} \longrightarrow \mathbf{U} \tag{2.27}$$

é definida por

$$V^{\mathbf{h}}(\mathbf{x}) = R^{\mathbf{h}}(\mathbf{x}) = Q^{\mathbf{h}}(\mathbf{x}, \mathbf{h}(\mathbf{x})). \tag{2.28}$$

A função V^* ótima é definida por

$$V^*(\mathbf{x}) = \max_{\mathbf{h}} V^{\mathbf{h}}(\mathbf{x}) = \max_{\mathbf{u}} Q^*(\mathbf{x}, \mathbf{u}). \tag{2.29}$$

Uma política ótima $\mathbf{h}^*$ pode ser calculada via V^* da forma

$$\mathbf{h}^*(\mathbf{x}) \in \arg\max_{\mathbf{u}}[\rho(\mathbf{x}, \mathbf{u}) + \gamma V^*(f(\mathbf{x}, \mathbf{u}))]. \tag{2.30}$$

As funções $V^{\mathbf{h}}$ e V^* satisfazem as equações de Bellman

$$V^{\mathbf{h}}(\mathbf{x}) = \rho(\mathbf{x}, \mathbf{h}(\mathbf{x})) + \gamma V^{\mathbf{h}}(f(\mathbf{x}, \mathbf{h}(\mathbf{x}))) \tag{2.31}$$

$$V^*(\mathbf{x}) = \max_{\mathbf{u}} \left[\rho(\mathbf{x}, \mathbf{u}) + \gamma V^*(f(\mathbf{x}, \mathbf{u}))\right]. \tag{2.32}$$

2.1.2 Enfoque estocástico

Em um MDP estocástico, o próximo estado não é determinístico dado o estado e a ação atual. Ao invés disso, o próximo estado é uma variável aleatória, e o estado e a ação atual fornecem a função de densidade de probabilidade dessa variável aleatória.

Formalmente, um MDP estocástico é uma tupla $(\mathbf{X}, \mathbf{U}, \tilde{f}, \tilde{\rho})$ na qual:

1. $\mathbf{X}$ é um conjunto de estados

2. $\mathbf{U}$ é um conjunto de ações

3. $\tilde{f} : \mathbf{X} \times \mathbf{U} \times \mathbf{X} \longrightarrow [0, \infty)$ é uma função de transição de probabilidade

$$P(\mathbf{x}_{k+1} \in \mathbf{X}_{k+1} | \mathbf{x}_k, \mathbf{u}_k) = \int_{\mathbf{X}_{k+1}} \tilde{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}') d\mathbf{x}', \tag{2.33}$$

sendo P o operador de probabilidade do estado $\mathbf{x}_{k+1}$, este pertence a $\mathbf{X}_{k+1} \subset \mathbf{X}$ devido à ação $\mathbf{u}_k$ aplicada no estado atual $\mathbf{x}_k$.

4. $\tilde{\rho} : \mathbf{X} \times \mathbf{U} \times \mathbf{X} \longrightarrow \mathbb{R}$ é uma função de recompensa tal que

$$\|\tilde{\rho}\|_\infty = \sup\{|\tilde{\rho}(\mathbf{x}, \mathbf{u}, \mathbf{x}')| \ : \ \mathbf{x}, \mathbf{x}' \in \mathbf{X}, \mathbf{u} \in \mathbf{U}\} < \infty. \tag{2.34}$$

Quando o conjunto de estados $\mathbf{X}$ é discreto, a função de transição de probabilidade é da forma $\tilde{f} : \mathbf{X} \times \mathbf{U} \times \mathbf{X} \longrightarrow [0, 1]$, onde

$$P(\mathbf{x}_{k+1} = \mathbf{x}' | \mathbf{x}_k, \mathbf{u}_k) = \tilde{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}'). \tag{2.35}$$

No caso estocástico, a propriedade de Markov diz que $\mathbf{x}_k$ e $\mathbf{u}_k$ determinam a função de densidade de probabilidade do próximo estado.

Exemplo 2.1.4 *Considere de novo o problema do robô de limpeza como no Exemplo 2.1.1. Suponha que a transição de estado não é determinístico. Quando o robô tenta se mover em uma direção, o sucesso é com probabilidade de* 0.8*. Com probabilidade de* 0.15 *ele permanece no mesmo lugar. Com probabilidade de* 0.05 *ele se move para o lado contrário (Figura 6).*

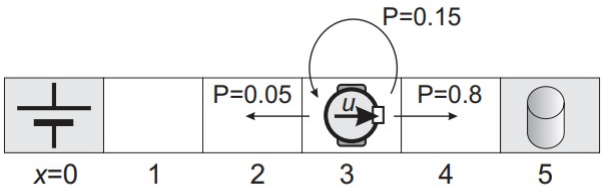


Figura 6 – O problema do robô de limpeza estocástico

A função de transição que modela as transições probabilísticas são descritas na Tabela 1.

O robô recebe recompensa como no caso determinístico

$$\tilde{\rho}(\mathbf{x}, \mathbf{u}, \mathbf{x}') = \begin{cases} 5 & \text{se} \quad \mathbf{x} \neq 5 \text{ e } \mathbf{x}' = 5 \\ 1 & \text{se} \quad \mathbf{x} \neq 0 \text{ e } \mathbf{x}' = 0 \\ 0 & \text{se} \quad \text{outro caso} \end{cases} \tag{2.36}$$

Tabela 1 – Dinâmica do MDP robô de limpeza estocástico.

$(\mathbf{x}, \mathbf{u})$	$\tilde{f}(\mathbf{x}, \mathbf{u}, 0)$	$\tilde{f}(\mathbf{x}, \mathbf{u}, 1)$	$\tilde{f}(\mathbf{x}, \mathbf{u}, 2)$	$\tilde{f}(\mathbf{x}, \mathbf{u}, 3)$	$\tilde{f}(\mathbf{x}, \mathbf{u}, 4)$	$\tilde{f}(\mathbf{x}, \mathbf{u}, 5)$
$(0, -1)$	1	0	0	0	0	0
$(1, -1)$	$0,8$	$0,15$	$0,05$	0	0	0
$(2, -1)$	0	$0,8$	$0,15$	$0,05$	0	0
$(3, -1)$	0	0	$0,8$	$0,15$	$0,05$	0
$(4, -1)$	0	0	0	$0,8$	$0,15$	$0,05$
$(5, -1)$	0	0	0	0	0	1
$(0, 1)$	1	0	0	0	0	0
$(1, 1)$	$0,05$	$0,15$	$0,8$	0	0	0
$(2, 1)$	0	$0,05$	$0,15$	$0,8$	0	0
$(3, 1)$	0	0	$0,05$	$0,15$	$0,8$	0
$(4, 1)$	0	0	0	$0,05$	$0,15$	$0,8$
$(5, 1)$	0	0	0	0	0	1

Exemplo 2.1.5 *Seja* $(\mathbf{X}, \mathbf{U}, f, r)$ *o processo de decisão de Markov como no Exemplo 2.1.2 e considere*

$$\mathbf{v}_k = \begin{bmatrix} v_k^1 & v_k^2 & \dots & v_k^m \end{bmatrix}^\top \tag{2.37}$$

um processo aleatório normal de média $\mathbf{0}$ *e covariância* $\Gamma[\mathbf{v}_k] = \sigma^2\mathbf{I}$*, sendo* σ_v^2 *a variância de* v_k^i*, para todo* $1 \leq i \leq m$ *e todo* $k \geq 0$*.*

Seja $(\tilde{\mathbf{X}}, \tilde{\mathbf{U}}, \tilde{f}, \tilde{r})$ *em que*

$$\tilde{\mathbf{X}} = \{\tilde{\mathbf{x}}_k \mid \tilde{\mathbf{x}}_{k+1} = \mathbf{A}\tilde{\mathbf{x}}_k + \mathbf{B}\tilde{\mathbf{u}}_k, \tilde{\mathbf{x}}_0 = \mathbf{x}_0\} \tag{2.38}$$

$$\tilde{\mathbf{U}} = \{\tilde{\mathbf{u}}_k = \mathbf{K}\tilde{\mathbf{x}}_k + \mathbf{v}_k\}. \tag{2.39}$$

Para cada $\mathbf{x}$ *e* $\mathbf{u}$*,* $\tilde{f}(\mathbf{x}, \mathbf{u}, \cdot)$ *é uma função de densidade com distribuição normal e*

$$-\tilde{r}(\tilde{\mathbf{x}}_k, \tilde{\mathbf{u}}_k, \tilde{\mathbf{x}}_{k+1}) = -(\tilde{\mathbf{x}}_k^\top \mathbf{Q}\tilde{\mathbf{x}}_k + \tilde{\mathbf{u}}_k^\top \mathbf{R}\tilde{\mathbf{u}}_k). \tag{2.40}$$

Então $(\tilde{\mathbf{X}}, \tilde{\mathbf{U}}, \tilde{f}, \tilde{r})$ *é um MDP estocástico.*

Otimalidade no sentido estocástico

O alcance com desconto e horizonte infinito de um estado inicial $\mathbf{x}_0$ sob uma política $\mathbf{h}$ é definido como

$$R^{\mathbf{h}}(\mathbf{x}_0) = \lim_{K \longrightarrow \infty} \mathbf{E}_{\mathbf{x}_{k+1} \sim \tilde{f}(\mathbf{x}_k, \mathbf{h}(\mathbf{x}_k), \cdot)} \left[\sum_{k=0}^{K} \gamma^k \tilde{\rho}(\mathbf{x}_k, \mathbf{h}(\mathbf{x}_k), \mathbf{x}_{k+1}) \right], \tag{2.41}$$

na qual $\mathbf{E}$ denota o operador esperança com respeito à variável aleatória $\mathbf{x}_{k+1}$ e a notação $\mathbf{x}_{k+1} \sim \tilde{f}(\mathbf{x}_k, \mathbf{h}(\mathbf{x}_k), \cdot)$ significa que o estado $\mathbf{x}_{k+1}$ é determinado de acordo com a função de densidade $\tilde{f}$.

Funções valor e equações de Bellman no sentido estocástico

No caso estocástico, a função $Q^{\mathbf{h}}$, de uma política $\mathbf{h}$, é a esperança do alcance sob a transição estocástica. Seja $(\tilde{\mathbf{X}}, \tilde{\mathbf{U}}, \tilde{f}, \tilde{\rho})$ um MDP estocástico. A função-Q

$$Q^{\mathbf{h}} : \tilde{\mathbf{X}} \times \tilde{\mathbf{U}} \longrightarrow \mathbb{R} \tag{2.42}$$

de uma política

$$\mathbf{h} : \tilde{\mathbf{X}} \longrightarrow \tilde{\mathbf{U}} \tag{2.43}$$

é definida como

$$Q^{\mathbf{h}}(\mathbf{x}, \mathbf{u}) = \mathbf{E}_{\mathbf{x}' \sim \tilde{f}(\mathbf{x},\mathbf{u},\cdot)} \left[\rho(\mathbf{x}, \mathbf{u}, \mathbf{x}') + \gamma R^{\mathbf{h}}(\mathbf{x}') \right]. \tag{2.44}$$

A função Q^* ótima é definida de maneira similar a (2.21)

$$Q^*(\mathbf{x}, \mathbf{u}) = \max_{\mathbf{h}} Q^{\mathbf{h}}(\mathbf{x}, \mathbf{u}). \tag{2.45}$$

Da mesma forma que no caso determinístico, políticas ótimas podem ser calculadas desde Q^*

$$\mathbf{h}^*(\mathbf{x}) \in \arg\max_{\mathbf{u}}(Q^*(\mathbf{x}, \mathbf{u})). \tag{2.46}$$

As equações de Bellman $Q^{\mathbf{h}}$ e Q^* no sentido estocástico são definas como

$$Q^{\mathbf{h}}(\mathbf{x}, \mathbf{u}) = \mathbf{E}_{\mathbf{x}' \sim \tilde{f}(\mathbf{x},\mathbf{u},\cdot)} \left[\tilde{\rho}(\mathbf{x}, \mathbf{u}, \mathbf{x}') + \gamma Q^{\mathbf{h}}(\mathbf{x}', \mathbf{h}(\mathbf{x}')) \right] \tag{2.47}$$

$$Q^*(\mathbf{x}, \mathbf{u}) = \mathbf{E}_{\mathbf{x}' \sim \tilde{f}(\mathbf{x},\mathbf{u},\cdot)} \left[\tilde{\rho}(\mathbf{x}, \mathbf{u}, \mathbf{x}') + \gamma \max_{\mathbf{u}'} Q^*(\mathbf{x}', \mathbf{u}') \right]. \tag{2.48}$$

A função $V^{\mathbf{h}}$, de uma política $\mathbf{h}$, e a função V^* ótima são definidas como

$$V^{\mathbf{h}}(\mathbf{x}) = R^{\mathbf{h}}(\mathbf{x}) \tag{2.49}$$

$$V^*(\mathbf{x}) = \max_{\mathbf{h}} V^{\mathbf{h}}(\mathbf{x}). \tag{2.50}$$

No entanto, o cálculo de políticas ótimas usando V^* torna-se mais difícil, já que envolve uma esperança

$$\mathbf{h}^*(\mathbf{x}) \in \arg\max_{\mathbf{u}} \mathbf{E}_{\mathbf{x}' \sim \tilde{f}(\mathbf{x},\mathbf{u},\cdot)}[\tilde{\rho}(\mathbf{x}, \mathbf{u}, \mathbf{x}') + \gamma V^*(\mathbf{x}')]. \tag{2.51}$$

Em contraste, calcular uma política desde Q^* é tão simples quanto no caso determinístico.

As equações de Bellman para $V^{\mathbf{h}}$ e V^* são obtidas de (2.31) e (2.32)

$$V^{\mathbf{h}}(\mathbf{x}) = \mathbf{E}_{\mathbf{x}' \sim \tilde{f}(\mathbf{x},\mathbf{h}(\mathbf{x}),\cdot)} \left[\tilde{\rho}(\mathbf{x}, \mathbf{h}(\mathbf{x}), \mathbf{x}') + \gamma V^{\mathbf{h}}(\mathbf{x}') \right] \tag{2.52}$$

$$V^*(\mathbf{x}) = \max_{\mathbf{u}} \mathbf{E}_{\mathbf{x}' \sim \tilde{f}(\mathbf{x},\mathbf{h}(\mathbf{x}),\cdot)} \left[\tilde{\rho}(\mathbf{x}, \mathbf{u}, \mathbf{x}') + \gamma V^*(\mathbf{x}') \right]. \tag{2.53}$$

2.2 Iteração de valor

Técnicas de iteração de valor usam a equação de otimalidade de Bellman para calcular iterativamente uma função valor ótima, logo uma política ótima é obtida a partir dessa função valor ótima.

2.2.1 Iteração de valor baseada em modelo

Seja D o conjunto de todas as funções Q. A função iteração Q

$$T : D \longrightarrow D \tag{2.54}$$

calcula o lado direito da equação de otimalidade de Bellman (2.25) ou (2.48) para cada função Q. No caso determinístico

$$[T(Q)](\mathbf{x}, \mathbf{u}) = \rho(\mathbf{x}, \mathbf{u}) + \gamma \max_{\mathbf{u}'} Q(f(\mathbf{x}, \mathbf{u}), \mathbf{u}') \tag{2.55}$$

e no caso estocástico

$$[T(Q)](\mathbf{x}, \mathbf{u}) = \mathbf{E}_{\mathbf{x}' \sim \tilde{f}(\mathbf{x}, \mathbf{h}(\mathbf{x}), \cdot)} \left[\tilde{\rho}(\mathbf{x}, \mathbf{u}, \mathbf{x}') + \gamma \max_{\mathbf{u}'} Q(\mathbf{x}', \mathbf{u}') \right]. \tag{2.56}$$

O algoritmo de iteração Q começa com uma função Q arbitrária Q_0 e em cada iteração l atualiza a função Q usando

$$Q_{l+1} = T(Q_l). \tag{2.57}$$

Por outro lado, para cada par de funções Q e Q', se cumpre que

$$\|T(Q) - T(Q')\|_\infty \leq \gamma \|Q - Q'\|_\infty. \tag{2.58}$$

Se $\gamma < 1$ então T é uma contração. Portanto, existe um único ponto fixo de T (KREYSZIG, 1978, p. 300). Usando a função Q iteração e a equação de otimalidade de Bellman tem-se que Q^* é um ponto fixo de T, ou seja

$$Q^* = T(Q^*). \tag{2.59}$$

Então, o único ponto fixo de T é Q^*. Logo, uma política ótima pode ser calculada desde Q^* usando (2.46).

O Algoritmo 1 e o Algoritmo 2 mostram a iteração Q para MDP determinístico e estocástico, respectivamente.

2.2.2 Iteração de valor livre de modelo e a necessidade de exploração

A seguir é considerado o algoritmo de iteração de valor e estuda-se o algoritmo Q-learning, o método mais utilizado para esta classe de algoritmos. O Q-learning começa com

Algoritmo 1 Iteração Q para MDP determinístico

Input: dinâmica $\mathbf{f}$, recompensa ρ, fator de desconto γ
$Q_0 \longleftarrow 0$
for $l = 0, 1, 2, \ldots$ **do**
  for cada $(\mathbf{x}, \mathbf{u})$ **do**
    $Q_{l+1}(\mathbf{x}, \mathbf{u}) \longleftarrow \rho(\mathbf{x}, \mathbf{u}) + \gamma \max_{\mathbf{u}'} Q_l(f(\mathbf{x}, \mathbf{u}), \mathbf{u}')$
  end for
end for
Output: $Q^* = Q_l$

Algoritmo 2 Iteração Q para MDP estocástico

Input: dinâmica $\tilde{\mathbf{f}}$, recompensa $\tilde{\rho}$, fator de desconto γ
$Q_0 \longleftarrow 0$
for $l = 0, 1, 2, \ldots$ **do**
  for cada $(\mathbf{x}, \mathbf{u})$ **do**
    $Q_{l+1}(\mathbf{x}, \mathbf{u}) \longleftarrow \sum_{\mathbf{x}'} \tilde{f}(\mathbf{x}, \mathbf{u}, \mathbf{x}')[\tilde{\rho}(\mathbf{x}, \mathbf{u}, \mathbf{x}') + \gamma \max_{\mathbf{u}'} Q_l(\mathbf{x}', \mathbf{u}')]$
  end for
end for
Output: $Q^* = Q_l$

uma função arbitrária Q_0 e atualiza sem modelo, somente usando transições de estados observados e recompensas observadas, ou seja usando tuplas da forma $(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{k+1}, \mathbf{r}_{k+1})$. Depois de cada transição, a função Q é atualizada como segue

$$Q_{k+1}(\mathbf{x}_k, \mathbf{u}_k) = Q_k(\mathbf{x}_k, \mathbf{u}_k) + \alpha_k[r_{k+1} + \gamma \max_{\mathbf{u}'} + Q_k(\mathbf{x}_{k+1}, \mathbf{u}') - Q_k(\mathbf{x}_k, \mathbf{u}_k)], \tag{2.60}$$

sendo $\gamma \in (0, 1]$ é a taxa de aprendizagem. O termo

$$r_{k+1} + \gamma \max_{\mathbf{u}'} + Q_k(\mathbf{x}_{k+1}, \mathbf{u}') - Q_k(\mathbf{x}_k, \mathbf{u}_k) \tag{2.61}$$

em (2.60) é a diferença temporal, ou seja, a diferença entre a estimativa atualizada $r_{k+1} + \gamma \max_{\mathbf{u}'} + Q_k(\mathbf{x}_{k+1}, \mathbf{u}')$ do valor Q de $(\mathbf{x}_k, \mathbf{u}_k)$ e a estimativa atual $Q_k(\mathbf{x}_k, \mathbf{u}_k)$. No caso determinístico, o estimado é a função Q-iteração (2.55) aplicada a Q_k em $(\mathbf{x}_k, \mathbf{u}_k)$, na qual $\rho(\mathbf{x}_k, \mathbf{u}_k)$ é substituído pela recompensa observada r_{k+1} e $f(\mathbf{x}_k, \mathbf{u}_k)$ pelo próximo estado $\mathbf{x}_{k+1}$. No caso estocástico, essas substituições fornecem uma única amostra da quantidade aleatória cuja esperança é calculada de (2.56).

Quando $k \longrightarrow \infty$, Q learning converge assintoticamente a Q^* sob as seguintes condições:

- A soma $\sum_{k=0}^{\infty} \alpha_k^2$ produz uma valor finito, enquanto que $\sum_{k=0}^{\infty} \alpha_k$ produz um valor infinito.
- Todos os pares estado-ação são (assintoticamente) visitados infinitamente.

A primeira condição é satisfeita se

$$\alpha_k = \frac{1}{k}. \tag{2.62}$$

A segunda condição pode ser satisfeita se, entre outras coisas, o controlador tiver uma probabilidade diferente de zero de selecionar qualquer ação em cada estado encontrado; isso é chamado exploração (BUSONIU et al., 2010, p. 29). O controlador também tem que explorar seu conhecimento atual para obter um bom desempenho, por exemplo selecionando ações greedy na atual função Q. Uma maneira clássica de equilibrar a exploração com *exploitation* em Q-learning é a *exploration* ϵ-greedy que seleciona ações seguindo

$$\mathbf{u}_k = \begin{cases} \mathbf{u} \in \arg\max_{\bar{\mathbf{u}}} Q_k(\mathbf{x}_k, \bar{\mathbf{u}}) & \text{com } P(\mathbf{u}|\mathbf{x}_k) = 1 - \epsilon_k \\ \text{ação aleatória uniformemente em } \mathbf{U} & \text{com } P(\mathbf{u}|\mathbf{x}_k) = \epsilon_k, \end{cases}. \tag{2.63}$$

na qual $\epsilon_k \in (0, 1)$ é a probabilidade de exploração no passo k.

O Algoritmo 3 mostra o Q-learning com exploração ϵ-greedy.

Algoritmo 3 Q-learning com exploração ϵ-greedy

1: **Input:** fator de desconto γ, $\{\epsilon_k\}_{k=0}^{\infty}$ exploração, $\{\alpha_k\}_{k=0}^{\infty}$ taxa de aprendizagem
2: $Q_0 \longleftarrow 0$
3: Estado inicial medido $\mathbf{x}_0$
4: **for** cada passo do tempo $k = 0, 1, 2, \ldots$ **do**
5: $\quad \mathbf{u}_k \longleftarrow \begin{cases} \mathbf{u} \in \arg\max_{\bar{\mathbf{u}}} Q_k(\mathbf{x}_k, \bar{\mathbf{u}}) & \text{com } P(\mathbf{u}|\mathbf{x}_k) = 1 - \epsilon_k \\ \text{ação aleatória uniformemente em } \mathbf{U} & \text{com } P(\mathbf{u}|\mathbf{x}_k) = \epsilon_k, \end{cases}$.
6: $\quad$ aplique $\mathbf{u}_k$, meça o próximo estado $\mathbf{x}_{k+1}$ e recompensa a r_{k+1}
7: $\quad Q_{k+1}(\mathbf{x}_k, \mathbf{u}_k) \longleftarrow Q_k(\mathbf{x}_k, \mathbf{u}_k) + \alpha_k[r_{k+1} + \gamma \max_{\mathbf{u}'} Q_k(\mathbf{x}_{k+1}, \mathbf{u}') - Q_k(\mathbf{x}_k, \mathbf{u}_k)]$
8: **end for**

Observação 2.2.1 *O Algoritmo 3 e todos os algoritmos apresentados no presente capítulo são para MDP finitos. Portanto, as funções Q e as políticas podem ser representadas como uma tabela indexada pelos estados e ações discretas (Tabelas 2 e 3, respectivamente).*

Tabela 2 – Função Q representada como uma tabela indexada pelos estados e ações.

	$\mathbf{u}_1$	$\mathbf{u}_2$	$\cdots$	$\mathbf{u}_m$
$\mathbf{x}_1$	$Q(\mathbf{x}_1, \mathbf{u}_1)$	$Q(\mathbf{x}_1, \mathbf{u}_2)$	$\cdots$	$Q(\mathbf{x}_1, \mathbf{u}_m)$
$\mathbf{x}_2$	$Q(\mathbf{x}_2, \mathbf{u}_1)$	$Q(\mathbf{x}_2, \mathbf{u}_2)$	$\cdots$	$Q(\mathbf{x}_2, \mathbf{u}_m)$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
$\mathbf{x}_n$	$Q(\mathbf{x}_n, \mathbf{u}_1)$	$Q(\mathbf{x}_n, \mathbf{u}_2)$	$\cdots$	$Q(\mathbf{x}_n, \mathbf{u}_m)$

Tabela 3 – Política $\mathbf{h}$ representada como uma tabela indexada pelos estados.

Estados	Política $\mathbf{h}$
$\mathbf{x}_1$	$\mathbf{u}_1 = \mathbf{h}(\mathbf{x}_1)$
$\mathbf{x}_2$	$\mathbf{u}_2 = \mathbf{h}(\mathbf{x}_2)$
$\vdots$	$\vdots$
$\mathbf{x}_n$	$\mathbf{u}_n = \mathbf{h}(\mathbf{x}_n)$

2.3 Iteração de política

Algoritmos de iteração de política avaliam políticas para construir suas funções valor e, em seguida, usam essas funções valor para encontrar uma política melhorada. Como um exemplo de iteração de política, é considerado um algoritmo que avalia políticas usando funções Q. Esse algoritmo começa com uma política arbitrária $\mathbf{h}_0$. Em cada iteração l a função $Q^{\mathbf{h}_l}$ da atual política $\mathbf{h}_l$ é determinada; esse passo é chamado *avaliação de política*. A avaliação de política é realizada resolvendo-se a equação de Bellman (2.24) no caso determinístico, ou (2.47) no caso estocástico. Quando a avaliação da política estiver concluída, a nova política $\mathbf{h}_{l+1}$ que é greedy em $Q^{\mathbf{h}}$ é encontrada:

$$\mathbf{h}_{l+1}(\mathbf{x}) \in \arg\max_{\mathbf{u}} Q^{\mathbf{h}_l}(\mathbf{x}, \mathbf{u}). \tag{2.64}$$

O passo (2.64) é chamada de *melhoria de política*. O procedimento é resumido no Algoritmo 4.

Algoritmo 4 Iteração de política com funções Q

1: **Input:** Política $\mathbf{h}_0(\mathbf{x})$
2: Estado inicial medido $\mathbf{x}_0$
3: **for** $l = 0, 1, 2, \ldots$ **do**
4: Encontrar $Q^{\mathbf{h}_l}(\mathbf{x}, \mathbf{u})$
5: $\mathbf{h}_{l+1}(\mathbf{x}) \in \arg\max_{\mathbf{u}} Q^{\mathbf{h}_l}(\mathbf{x}, \mathbf{u})$
6: **end for**
7: **Output:** $\mathbf{h}^*(\mathbf{x}) = \mathbf{h}_{l+1}(\mathbf{x})$, $Q^*(\mathbf{x}, \mathbf{u}) = Q^{\mathbf{h}_{l+1}}(\mathbf{x}, \mathbf{u})$

Observação 2.3.1 *O Algoritmo 4, em cada iteração l, avalia a política $\mathbf{h}_l(\mathbf{x})$ até encontrar seu valor $Q^{\mathbf{h}_l}(\mathbf{x}, \mathbf{u})$ caraterizada pela equação de Bellman (2.24). Então, uma nova política melhorada $\mathbf{h}_{l+1}(\mathbf{x})$ é determinada como em (2.64). O processo é repetido até obter convergência da sequência $Q^{\mathbf{h}_l}(\mathbf{x}, \mathbf{x})$ para a função valor ótima $Q^*(\mathbf{x}, \mathbf{u})$. Então uma política ótima $\mathbf{h}^*(\mathbf{x})$ é obtida a partir de $Q^*(\mathbf{x}, \mathbf{u})$.*

2.4 Iteração de política baseada em modelo

O algoritmo de iteração de política é baseado nas equações de Bellman (2.24) (no caso determinístico) e (2.47) (no caso estocástico). Uma função avaliação de política

$T : D \longrightarrow D$ é definida no caso determinístico como

$$[T^{\mathbf{h}}(Q)](\mathbf{x}, \mathbf{u}) = \rho(\mathbf{x}, \mathbf{u}) + \gamma Q(f(\mathbf{x}, \mathbf{u}), \mathbf{h}(f(\mathbf{x}, \mathbf{u}))) \tag{2.65}$$

e no caso estocástico

$$[T^{\mathbf{h}}(Q)](\mathbf{x}, \mathbf{u}) = \mathbf{E}_{\mathbf{x}' \sim \tilde{f}(\mathbf{x}, \mathbf{u}, \cdot)} \left[\tilde{\rho}(\mathbf{x}, \mathbf{u}, \mathbf{x}') + \gamma Q(\mathbf{x}', \mathbf{h}(\mathbf{x}')) \right]. \tag{2.66}$$

Quando o espaço de estados é numerável, o modelo de transição (2.35) é apropriado e a função política de avaliação para o caso estocástico (2.66) pode ser escrita como

$$[T^{\mathbf{h}}(Q)](\mathbf{x}, \mathbf{u}) = \sum_{\mathbf{x}'} \tilde{f}(\mathbf{x}, \mathbf{u}, \mathbf{x}') \left[\tilde{\rho}(\mathbf{x}, \mathbf{u}, \mathbf{x}') + \gamma Q(\mathbf{x}', \mathbf{h}(\mathbf{x}')) \right]. \tag{2.67}$$

Avaliação de política para funções Q começam com uma função Q arbitrária $Q_0^{\mathbf{h}}$ e em cada iteração τ atualiza a função Q usando

$$Q_{\tau+1}^{\mathbf{h}} = T^{\mathbf{h}}(Q_\tau^{\mathbf{h}}). \tag{2.68}$$

Para cada par de funções Q, Q' e fator de desconto $\gamma < 1$ tem-se

$$\|T^{\mathbf{h}}(Q) - T^{\mathbf{h}}(Q')\|_\infty \leq \gamma \|Q - Q'\|_\infty, \tag{2.69}$$

ou seja, $T^{\mathbf{h}}$ tem um único ponto fixo. Escrevendo em termos de $T^{\mathbf{h}}$, a equação de Bellman (2.24) ou (2.47) estabelece que o único ponto fixo é a atual $Q^{\mathbf{h}}$:

$$Q^{\mathbf{h}} = T^{\mathbf{h}}(Q^{\mathbf{h}}). \tag{2.70}$$

O Algoritmo 5 mostra a avaliação de política para funções Q no caso de MDP determinístico enquanto que o Algoritmo 6 é usado para MDP estocástico com espaço de estado numerável.

Algoritmo 5 Avaliação de política com funções Q em DMP determinístico

1: **Input:** Política $\mathbf{h}(\mathbf{x})$, dinâmica f, recompensa ρ, fator de desconto γ
2: $Q_0^{\mathbf{h}}(\mathbf{x}, \mathbf{u}) \longleftarrow 0$
3: **for** $\tau = 0, 1, 2, \ldots$ **do**
4: **for** cada $(\mathbf{x}, \mathbf{u})$ **do**
5: $Q_{\tau+1}^{\mathbf{h}}(\mathbf{x}, \mathbf{u}) \longleftarrow \rho(\mathbf{x}, \mathbf{u}) + \gamma Q_\tau^{\mathbf{h}}(f(\mathbf{x}, \mathbf{u}), \mathbf{h}(f(\mathbf{x}, \mathbf{u})))$
6: **end for**
7: **end for**
8: **Output:** $Q^{\mathbf{h}}(\mathbf{x}, \mathbf{u}) = Q_{\tau+1}^{\mathbf{h}}(\mathbf{x}, \mathbf{u})$

Algoritmo 6 Avaliação de política com funções Q em DMP estocásticos com espaço de estado numerável

Input: Política $\mathbf{h}(\mathbf{x})$, dinâmica $\mathbf{f}$, recompensa $\tilde{\rho}$, fator de desconto γ
$Q_0^{\mathbf{h}}(\mathbf{x},\mathbf{u}) \longleftarrow 0$
for $\tau = 0, 1, 2, \ldots$ **do**
  for cada $(\mathbf{x},\mathbf{u})$ **do**
    $Q_{\tau+1}^{\mathbf{h}}(\mathbf{x},\mathbf{u}) \longleftarrow \sum_{\mathbf{x}'} \tilde{f}(\mathbf{x},\mathbf{u},\mathbf{x}')\left[\tilde{\rho}(\mathbf{x},\mathbf{u},\mathbf{x}') + \gamma Q_{\tau}^{\mathbf{h}}(\mathbf{x}',\mathbf{h}(\mathbf{x}'))\right]$
  end for
end for
Output: $Q^{\mathbf{h}}(\mathbf{x},\mathbf{u}) = Q_{\tau+1}^{\mathbf{h}}(\mathbf{x},\mathbf{u})$

2.5 Iteração de política livre de modelo

A seguir será apresentado um algoritmo de iteração de política livre de modelo chamado SARSA. O nome SARSA é obtido juntando as iniciais de cada elemento nos dados empregado no algoritmo : *state* (estado), *action* (ação), *reward* (recompensa), *state* (próximo estado) e *action* (próxima ação). Formalmente, essa tupla é denotada por

$$(\mathbf{x}_k, \mathbf{u}_k, r_{k+1}, \mathbf{x}_{k+1}, \mathbf{u}_{k+1}). \tag{2.71}$$

O algoritmo SARSA começa com uma função Q inicial Q_0 e atualiza em cada passo usando tuplas da forma (2.71) como segue

$$Q_{k+1}(\mathbf{x}_k,\mathbf{u}_k) = Q_k(\mathbf{x}_k,\mathbf{u}_k) + \alpha_k[r_{k+1} + \gamma Q_k(\mathbf{x}_{k+1},\mathbf{u}_{k+1}) - Q_k(\mathbf{x}_k,\mathbf{u}_k)], \tag{2.72}$$

onde $\gamma \in (0, 1]$ é a taxa de aprendizagem. O termo

$$r_{k+1} + \gamma Q_k(\mathbf{x}_{k+1},\mathbf{u}_{k+1}) - Q_k(\mathbf{x}_k,\mathbf{u}_k) \tag{2.73}$$

é a diferença temporal, obtida como a diferença entre a estimativa atualizada $r_{k+1} + \gamma Q_k(\mathbf{x}_{k+1},\mathbf{u}_{k+1})$ do valor Q de $(\mathbf{x}_k,\mathbf{u}_k)$ e a estimativa atual $Q_k(\mathbf{x}_k,\mathbf{u}_k)$. Note que a diferença temporal em (2.61) inclui o máximo valor Q no próximo estado, a diferença temporal SARSA inclui o valor Q da ação atual no próximo estado, ou seja, o SARSA realiza online e livre de modelo a avaliação de política que atualmente está sendo seguida.

O Algoritmo 7 mostra o método SARSA com exploração ϵ-greedy.

2.6 Busca de política

O método de busca de política usa técnicas de optimização para diretamente buscar uma política ótima que maximize o alcance para cada estado inicial, ou seja, a busca de política resolve o problema

$$\max_{\mathbf{h}} \left[\sum_{k=0}^{\infty} \gamma^k \rho(\mathbf{x}_k, \mathbf{h}(\mathbf{x}_k))\right], \text{ para todo } \mathbf{x}_0 \tag{2.74}$$

Algoritmo 7 SARSA com exploração ϵ-greedy

Input: fator de desconto γ, $\{\epsilon_k\}_{k=0}^{\infty}$ exploração, $\{\alpha_k\}_{k=0}^{\infty}$ taxa de aprendizagem
$Q_0 \longleftarrow 0$
Estado inicial medido $\mathbf{x}_0$
$\mathbf{u}_0 \longleftarrow \begin{cases} \mathbf{u} \in \arg\max_{\bar{\mathbf{u}}} Q_0(\mathbf{x}_0, \bar{\mathbf{u}}) & \text{com } P(\mathbf{u}|\mathbf{x}_0) = 1 - \epsilon_0 \\ \text{ação aleatória uniformemente em } \mathbf{U} & \text{com } P(\mathbf{u}|\mathbf{x}_0) = \epsilon_0, \end{cases}$.
for cada passo do tempo $k = 0, 1, 2, \ldots$ **do**
  aplique $\mathbf{u}_k$, meça o próximo estado $\mathbf{x}_{k+1}$ e recompensa a r_{k+1}
  $\mathbf{u}_{k+1} \longleftarrow \begin{cases} \mathbf{u} \in \arg\max_{\bar{\mathbf{u}}} Q_k(\mathbf{x}_{k+1}, \bar{\mathbf{u}}) & \text{com } P(\mathbf{u}|\mathbf{x}_{k+1}) = 1 - \epsilon_{k+1} \\ \text{ação aleatória uniformemente em } \mathbf{U} & \text{com } P(\mathbf{u}|\mathbf{x}_{k+1}) = \epsilon_{k+1}, \end{cases}$.
  $Q_{k+1}(\mathbf{x}_k, \mathbf{u}_k) \longleftarrow Q_k(\mathbf{x}_k, \mathbf{u}_k) + \alpha_k[r_{k+1} + \gamma Q_k(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}) - Q_k(\mathbf{x}_k, \mathbf{u}_k)]$
end for

no caso determinístico, e

$$\max_{\mathbf{h}} \left(\lim_{\mathbf{K}\to\infty} \mathbf{E}_{\mathbf{x}_{k+1}\sim\tilde{f}(\mathbf{x}_k,\mathbf{h}(\mathbf{x}_k),\cdot)} \left[\sum_{k=0}^{\mathbf{K}} \gamma^k \tilde{\rho}(\mathbf{x}_k, \mathbf{h}(\mathbf{x}_k), \mathbf{x}_{k+1}) \right] \right), \text{ para todo } \mathbf{x}_0 \tag{2.75}$$

no caso estocástico. Teoricamente, qualquer método de otimização pode ser usado para a busca de uma política ótima. No entanto, para um problema geral, o critério de otimização poderia ser uma função não diferenciável com vários mínimos locais. Portanto, técnicas de otimização global sem derivada são mais apropriados que técnicas locais baseadas em gradiente. Exemplos particulares de técnicas globais, sem derivadas, incluem algoritmos genéticos (GOLDBER, 1989), busca tabu (GLOVERA; LAGUNA, 1977), busca de padrão (TORCZON, 1997; LEWIS; TORCZON, 2000), otimização *cross-entropy* (RUBINSTEIN; KROESE, 1989), dentre outras.

Considere o procedimento de estimação do alcance de uma busca de política baseada em modelo. Os alcances são somas infinitas de recompensas com descontos (2.10), (2.41). No entanto, na prática, o alcance tem que ser estimado em um tempo finito. A soma infinita no alcance pode ser aproximada com um número finito de passos K. Para garantir que a aproximação obtida dessa maneira tem um limitante $\epsilon_{MC} > 0$ da soma infinita, K pode ser escolhido como (MANNOR; RUBINSTEIN; GAT, 2003):

$$K = log_\gamma \frac{\epsilon_{MC}(1-\gamma)}{\|\rho\|_\infty}. \tag{2.76}$$

Avaliar o critério de otimização da busca de política requer a estimativa precisa do alcance desde todos os estados iniciais. Esse procedimento é muito custoso, já que algoritmos de otimização requerem muitas avaliações do critério, geralmente mais do que iteração de valor e iteração de política.

O presente capítulo apresentou as principais técnicas de programação dinâmica e aprendizagem por reforço quando o conjunto de estados e ações é discreto. No entanto, em problemas de controle, o conjunto de estados e ações é contínuo. Nesse caso, a função valor

e a política têm que ser aproximadas usando aproximadores paramétricos. No Capítulo 5 será apresentado o método Q-learning baseado em iteração de política para resolver o problema de controle ótimo LQR, no qual um aproximador paramétrico linear é usado para aproximar a função valor.

2.7 Considerações finais

Neste capítulo, as principais técnicas de programação dinâmica e aprendizagem por reforço, iteração de política e iteração de valor, foram apresentadas. Todos os métodos neste capítulo são para processos de decisão de Markov com estados discretos e finitos. Como consequência, as funções valor podem ser representadas como uma tabela indexada pelos estados e ações discretas. No entanto, no contexto do controle ótimo LQR, os estados são contínuos e infinitos. Nesses casos, é necessário aproximar a função valor usando estimadores paramétricos que serão apresentados no Capítulo 5.

3 Controle Ótimo Discreto

Seja a planta geral não linear em tempo discreto

$$\mathbf{x}_{k+1} = \mathbf{f}^k(\mathbf{x}_k, \mathbf{u}_k) \tag{3.1}$$

com condição inicial $\mathbf{x}_0$. O sobrescrito k na função

$$\mathbf{f}^k : \mathbb{R}^n \times \mathbb{R}^m \longrightarrow \mathbb{R}^n \tag{3.2}$$

indica que, em geral, o sistema pode ter dinâmica variante no tempo. Os vetores $\mathbf{x}_k \in \mathbb{R}^n$ e $\mathbf{u}_k \in \mathbb{R}^m$ são o estado do sistema e a ação de controle, respectivamente, no tempo k.

Além disso, seja o índice de desempenho escalar associado à planta (3.1)

$$J_i = \phi(N, \mathbf{x}_N) + \sum_{k=i}^{N-1} L^k(\mathbf{x}_k, \mathbf{u}_k), \tag{3.3}$$

na qual $[i, N]$ é o intervalo em uma escala de tempo discreto com uma etapa de amostra fixa, sendo as funções escalares

$$\phi : \mathbb{R} \times \mathbb{R}^n \longrightarrow \mathbb{R} \text{ e } L^k : \mathbb{R}^n \times \mathbb{R}^m \longrightarrow \mathbb{R} \tag{3.4}$$

invariante no tempo e variante no tempo, respectivamente. O problema de controle ótimo é encontrar a sequência de controle

$$\mathbf{u}_i^*, \mathbf{u}_{i+1}^*, \ldots, \mathbf{u}_{N-1}^* \tag{3.5}$$

que impulsiona o sistema (3.1) ao longo de uma trajetória

$$\mathbf{x}_{i+1}^*, \mathbf{x}_{i+2}^*, \ldots, \mathbf{x}_N^* \tag{3.6}$$

tal que (3.3) é minimizada (Figura 7).

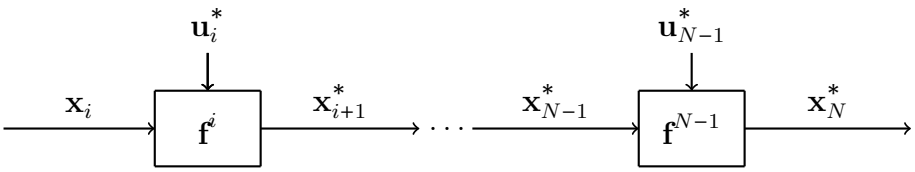


Figura 7 – A sequência de controle $\mathbf{u}_i^*, \ldots, \mathbf{u}_{N-1}^*$ aplicado ao sistema (3.1), com estado inicial $\mathbf{x}_i$, produz a sequência de estados $\mathbf{x}_{i+1}^*, \ldots, \mathbf{x}_N^*$ tal que (3.3) é mínimo.

Uma maneira de resolver o problema de controle ótimo é usar a abordagem de multiplicador de Lagrange. Note que, para cada instante do tempo k, há uma restrição $\mathbf{f}^k(\mathbf{x}_k, \mathbf{u}_k)$ no intervalo $[i, N]$.

Seja $\lambda_k \in \mathbb{R}^n$ e seja $\bar{J}$ o índice de desempenho aumentado (LEWIS; VRABIE; SYRMOS, 2012, p. 22)

$$\bar{J} = \phi(N, \mathbf{x}_N) + \sum_{k=i}^{N-1} \left[L^k(\mathbf{x}_k, \mathbf{u}_k) + \lambda_{k+1}^\top \left(\mathbf{f}^k(\mathbf{x}_k, \mathbf{u}_k) - \mathbf{x}_{k+1} \right) \right] \tag{3.7}$$

e seja a função Hamiltoniana

$$H^k(\mathbf{x}_k, \mathbf{u}_k) = L^k(\mathbf{x}_k, \mathbf{u}_k) + \lambda_{k+1}^\top \mathbf{f}^k(\mathbf{x}_k, \mathbf{u}_k). \tag{3.8}$$

Então,

$$\begin{aligned}
\bar{J} &= \phi(N, \mathbf{x}_N) + \sum_{k=i}^{N-1} \left[H^k(\mathbf{x}_k, \mathbf{u}_k) - \lambda_{k+1}^\top \mathbf{x}_{k+1} \right] && (3.9)\\
&= \phi(N, \mathbf{x}_N) + \sum_{k=i}^{N-1} H^k(\mathbf{x}_k, \mathbf{u}_k) - \sum_{k=i}^{N-2} [\lambda_{k+1}^\top \mathbf{x}_{k+1}] - \lambda_N^\top \mathbf{x}_N && (3.10)\\
&= \phi(N, \mathbf{x}_N) - \lambda_N^\top \mathbf{x}_N + H^i(\mathbf{x}_i, \mathbf{u}_i) + \sum_{k=i+1}^{N-1} H^k(\mathbf{x}_k, \mathbf{u}_k) - \sum_{k=i+1}^{N-1} [\lambda_k^\top \mathbf{x}_k] && (3.11)\\
&= \phi(N, \mathbf{x}_N) - \lambda_N^\top \mathbf{x}_N + H^i(\mathbf{x}_i, \mathbf{u}_i) + \sum_{k=i+1}^{N-1} [H^k(\mathbf{x}_k, \mathbf{u}_k) - \lambda_k^\top \mathbf{x}_k]. && (3.12)
\end{aligned}$$

Pela teoria de multiplicador de Lagrange, tem-se que em um mínimo restrito, $d\bar{J}$ deve ser igual a zero. Assim,

$$\begin{aligned}
d\bar{J} &= \left(\frac{\partial \phi}{\partial \mathbf{x}_N} - \lambda_N \right)^\top d\mathbf{x}_N + \left(H^i_{\mathbf{x}_i} \right)^\top d\mathbf{x}_i + \left(H^i_{\mathbf{u}_i} \right)^\top d\mathbf{u}_i \\
&\quad + \sum_{k=i+1}^{N-1} \left[(H^k_{\mathbf{x}_k} - \lambda_k)^\top d\mathbf{x}_k + (H^k_{\mathbf{u}_k})^\top d\mathbf{u}_k \right] + \sum_{k=i+1}^{N} \left[(H^{k-1}_{\lambda_k} - \mathbf{x}_k)^\top d\lambda_k \right], && (3.13)
\end{aligned}$$

onde

$$H^k_{\mathbf{x}_k} = \frac{\partial H^k}{\partial \mathbf{x}_k}, \; H^k_{\mathbf{u}_k} = \frac{\partial H^k}{\partial \mathbf{u}_k}, \; H^{k-1}_{\lambda_k} = \frac{\partial H^{k-1}}{\partial \lambda_k}. \tag{3.14}$$

As condições necessárias para um mínimo restrito são

$$\begin{aligned}
\mathbf{x}_{k+1} &= \frac{\partial H^k}{\partial \lambda_{k+1}}, \; k = i, \ldots, N-1, && (3.15)\\
\lambda_k &= \frac{\partial H^k}{\partial \mathbf{x}_k}, \; k = i, \ldots, N-1, && (3.16)\\
0 &= \frac{\partial H^k}{\partial \mathbf{u}_k}, \; k = i, \ldots, N-1, && (3.17)
\end{aligned}$$

que surgem dos termos contidos nos somatórios e do coeficiente de $d\mathbf{u}_i$ e

$$\begin{aligned}
\left(\frac{\partial \phi}{\partial \mathbf{x}_N} - \lambda_N \right)^\top d\mathbf{x}_N &= 0 && (3.18)\\
\left(\frac{\partial H^i}{\partial \mathbf{x}_i} \right)^\top d\mathbf{x}_i &= 0. && (3.19)
\end{aligned}$$

Escrevendo (3.15), (3.16) e (3.17) em função de (3.8) produz a formulação na Tabela 4.

Tabela 4 – Controlador ótimo discreto não linear.

Modelo do sistema:
$\mathbf{x}_{k+1} = \mathbf{f}^k(\mathbf{x}_k, \mathbf{u}_k),\ k > i$

Índice de desempenho:
$J_i = \phi(N, \mathbf{x}_N) + \sum_{k=i}^{N-1} L^k(\mathbf{x}_k, \mathbf{u}_k)$

Hamiltoniano:
$H^k(\mathbf{x}_k, \mathbf{u}_k) = L^k(\mathbf{x}_k, \mathbf{u}_k) + \lambda_{k+1}^\top \mathbf{f}^k(\mathbf{x}_k, \mathbf{u}_k)$

Equação de estado:
$\mathbf{x}_{k+1} = \dfrac{\partial H^k}{\partial \lambda_{k+1}} = \mathbf{f}^k(\mathbf{x}_k, \mathbf{u}_k)$

Equação de co-estado:
$\lambda_k = \dfrac{\partial H^k}{\partial \mathbf{x}_k} = \left(\dfrac{\partial \mathbf{f}^k}{\partial \mathbf{x}_k}\right)^\top \lambda_{k+1} + \dfrac{\partial L^k}{\partial \mathbf{x}_k}$

Condição de estacionariedade:
$0 = \dfrac{\partial H^k}{\partial \mathbf{u}_k} = \left(\dfrac{\partial \mathbf{f}^k}{\partial \mathbf{u}_k}\right)^\top \lambda_{k+1} + \dfrac{\partial L^k}{\partial \mathbf{u}_k}$

Condições de contorno:
$\left(\dfrac{\partial L^i}{\partial \mathbf{x}_i} + \left(\dfrac{\partial \mathbf{f}^i}{\partial \mathbf{x}_i}\right)^\top \lambda_{i+1}\right)^T d\mathbf{x}_i = 0$
$\left(\dfrac{\partial \phi}{\partial \mathbf{x}_N} - \lambda_N\right)^\top d\mathbf{x}_N = 0$

3.1 O regulador linear quadrático

Nesta seção, estuda-se o caso particular quando o sistema em (3.1) é linear e o índice de desempenho em (3.2) é uma superfície quadrática e convexa.

Seja a planta linear

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k, \tag{3.20}$$

onde $\mathbf{A}_k \in \mathbb{R}^{n\times n}$ e $\mathbf{B}_k \in \mathbb{R}^{n\times m}$. Seja o índice de desempenho associado à planta (3.20)

$$J_i = \frac{1}{2}\mathbf{x}_N^\top \mathbf{S}_N \mathbf{x}_N + \frac{1}{2}\sum_{k=i}^{N-1}\left[\mathbf{x}_k^\top \mathbf{Q}_k \mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}_k \mathbf{u}_k\right] \tag{3.21}$$

definido ao longo do intervalo de tempo discreto $[i, N]$, onde $\mathbf{Q}_k$ e $\mathbf{S}_N$ são matrizes simétricas e positivas semidefinidas e $\mathbf{R}_k$ é uma matriz simétrica positiva definida para todo k.

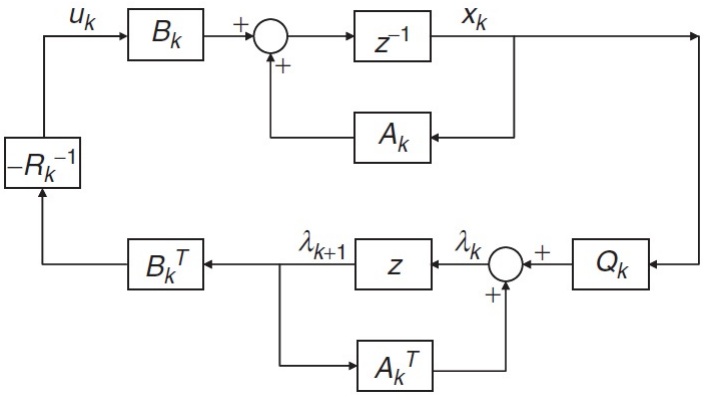


Figura 8 – Formulação de co-estado do controlador ótimo discreto LQR (LEWIS; VRABIE; SYRMOS, 2012)

O objetivo é encontrar a sequência de controle

$$\mathbf{u}_i, \mathbf{u}_{i+1}, \ldots, \mathbf{u}_{N-1} \tag{3.22}$$

que minimiza J_i. Esse problema é conhecido como o problema do regulador linear quadrático (LQR-*linear quadratic regulator*) (LEWIS; VRABIE; SYRMOS, 2012, p. 32). Para resolver o problema do LQR escreve-se a função Hamiltoniana

$$H^k = \frac{1}{2}\left(\mathbf{x}_k^\top \mathbf{Q}_k \mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}_k \mathbf{u}_k\right) + \lambda_{k+1}^\top (\mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k). \tag{3.23}$$

As equações de estado e co-estado são

$$\mathbf{x}_{k+1} = \frac{\partial H^k}{\partial \lambda_{k+1}} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k \tag{3.24}$$

$$\lambda_k = \frac{\partial H^k}{\partial \mathbf{x}_k} = \mathbf{Q}_k \mathbf{x}_k + \mathbf{A}_k^\top \lambda_{k+1} \tag{3.25}$$

e a condição de estacionariedade é

$$0 = \frac{\partial H^k}{\partial \mathbf{u}_k} = \mathbf{R}_k \mathbf{u}_k + \mathbf{B}_k^\top \lambda_{k+1}. \tag{3.26}$$

De (3.26), segue-se que

$$\mathbf{u}_k = -\mathbf{R}_k^{-1} \mathbf{B}_k^\top \lambda_{k+1}. \tag{3.27}$$

De maneira que se é conhecida a sequência de co-estados então a sequência de controle ótimo é determinada usando (3.27). A Figura 8 mostra o diagrama de blocos do controlador ótimo. Note de (3.27) que o sistema pode prever ou antecipar as entradas futuras. Portanto, o controlador não pode ser implementado dessa forma já que o sistema é não causal (CHEN, 1999, p. 6).

Substituindo (3.27) na equação de estado (3.24) obtém-se

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k - \mathbf{B}_k \mathbf{R}_k^{-1} \mathbf{B}_k^\top \lambda_{k+1}. \tag{3.28}$$

As equações de estado e co-estado podem ser escritas como

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \lambda_k \end{bmatrix} \begin{bmatrix} \mathbf{A}_k & -\mathbf{B}_k\mathbf{R}_k^{-1}\mathbf{B}_k^\top \\ \mathbf{Q}_k & \mathbf{A}_k^\top \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \lambda_{k+1} \end{bmatrix}. \tag{3.29}$$

A matriz

$$\mathbf{H} = \begin{bmatrix} \mathbf{A}_k & -\mathbf{B}_k\mathbf{R}_k^{-1}\mathbf{B}_k^\top \\ \mathbf{Q}_k & \mathbf{A}_k^\top \end{bmatrix} \tag{3.30}$$

é conhecida como matriz Hamiltoniana discreta e o sistema (3.29) é conhecido como sistema Hamiltoniano discreto (LEWIS; VRABIE; SYRMOS, 2012).

Se a matriz $\mathbf{A}$ tem inversa, então (3.28) pode ser reescrita como

$$\mathbf{x}_k = \mathbf{A}_k^{-1}\mathbf{x}_{k+1} + \mathbf{A}_k^{-1}\mathbf{B}_k\mathbf{R}_k^{-1}\mathbf{B}_k^\top\lambda_{k+1}. \tag{3.31}$$

De (3.25) e (3.31) segue-se que (3.29) pode ser reescrita como

$$\begin{bmatrix} \mathbf{x}_k \\ \lambda_k \end{bmatrix} \begin{bmatrix} \mathbf{A}_k^{-1} & \mathbf{A}_k^{-1}\mathbf{B}_k\mathbf{R}_k^{-1}\mathbf{B}_k^\top \\ \mathbf{Q}_k\mathbf{A}_k^{-1} & \mathbf{A}_k^\top + \mathbf{Q}_k\mathbf{A}_k^{-1}\mathbf{B}_k\mathbf{R}_k^{-1}\mathbf{B}_k^\top \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k+1} \\ \lambda_{k+1} \end{bmatrix}. \tag{3.32}$$

3.1.1 Estado final livre e controle em malha fechada

Nesta seção, considera-se o caso quando o estado final $\mathbf{x}_N$, no índice de desempenho (3.3), é livre, ou seja, não é restrito a tomar um valor exato, ele pode ser variado ao determinar o controle ótimo.

As equações de estado (3.28) e co-estado (3.25) para o problema do LQR são

$$\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k - \mathbf{B}_k\mathbf{R}_k^{-1}\mathbf{B}_k^\top\lambda_{k+1} \tag{3.33}$$

$$\lambda_k = \mathbf{Q}_k\mathbf{x}_k + \mathbf{A}_k^\top\lambda_{k+1}. \tag{3.34}$$

O controle é dado por (3.27)

$$\mathbf{u}_k = -\mathbf{R}_k^{-1}\mathbf{B}_k^\top\lambda_{k+1}. \tag{3.35}$$

Como o estado inicial $\mathbf{x}_i$ é conhecido e fixo, então $d\mathbf{x}_i = 0$. Portanto, a condição em (3.19) é satisfeita. Como o estado final $\mathbf{x}_N$ é livre, então $d\mathbf{x}_N \neq 0$. De acordo com a condição de contorno (3.18), é necessário que

$$\lambda_N = \frac{\partial\phi}{\partial\mathbf{x}_N}. \tag{3.36}$$

Note de (3.21) e (3.3), que a função ϕ no estado final $\mathbf{x}_N$ é

$$\phi = \frac{1}{2}\mathbf{x}_N^\top\mathbf{S}_N\mathbf{x}_N, \tag{3.37}$$

de maneira que

$$\lambda_N = \mathbf{S}_N\mathbf{x}_N. \tag{3.38}$$

A condição (3.38) é a relação entre o co-estado final e o estado final que é a nova condição terminal para o problema de estado final livre.

As equações (3.33) e (3.34) definem um problema *two-point boundary-value* já que as condições de contorno necessárias para sua solução são o estado inicial $\mathbf{x}_i$ e o co-estado final λ_N. Para resolver esse problema, é usado o método de varredura (BRYSON; HO, 1975, p. 47). Suponha que uma relação do tipo (3.38) é verdadeira para $k \leq N$:

$$\lambda_k = \mathbf{S}_k \mathbf{x}_k \tag{3.39}$$

para uma sequência de matrizes $\mathbf{S}_k \in \mathbb{R}^{n \times n}$. De (3.39) e (3.33) obtém-se

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k - \mathbf{B}_k \mathbf{R}_k^{-1} \mathbf{B}_k^\top \mathbf{S}_{k+1} \mathbf{x}_{k+1}. \tag{3.40}$$

Então, resolvendo (3.40)

$$\mathbf{x}_{k+1} = (\mathbf{I} + \mathbf{B}_k \mathbf{R}_k^{-1} \mathbf{B}_k^\top \mathbf{S}_{k+1})^{-1} \mathbf{A}_k \mathbf{x}_k. \tag{3.41}$$

Logo, substituindo (3.39) em (3.34) obtém-se

$$\mathbf{S}_k \mathbf{x}_k = \mathbf{Q}_k \mathbf{x}_k + \mathbf{A}_k^\top \mathbf{S}_{k+1} \mathbf{x}_{k+1} \tag{3.42}$$

e de (3.41) segue-se que

$$\mathbf{S}_k \mathbf{x}_k = \mathbf{Q}_k \mathbf{x}_k + \mathbf{A}_k^\top \mathbf{S}_{k+1} (\mathbf{I} + \mathbf{B}_k \mathbf{R}_k^{-1} \mathbf{B}_k^\top \mathbf{S}_{k+1})^{-1} \mathbf{A}_k \mathbf{x}_k. \tag{3.43}$$

Como $\mathbf{x}_k$ é geralmente não-nulo e (3.43) é verdadeira para todas as sequências de estados $\mathbf{x}_i$, então

$$\mathbf{S}_k = \mathbf{Q}_k + \mathbf{A}_k^\top \mathbf{S}_{k+1} (\mathbf{I} + \mathbf{B}_k \mathbf{R}_k^{-1} \mathbf{B}_k^\top \mathbf{S}_{k+1})^{-1} \mathbf{A}_k. \tag{3.44}$$

Usando o lema da inversão da matriz, obtém-se

$$\mathbf{S}_k = \mathbf{A}_k^\top \left[\mathbf{S}_{k+1} - \mathbf{S}_{k+1} \mathbf{B}_k (\mathbf{B}_k^\top \mathbf{S}_{k+1} \mathbf{B}_k + \mathbf{R}_k)^{-1} \mathbf{B}_k^\top \mathbf{S}_{k+1} \right] \mathbf{A}_k + \mathbf{Q}_k. \tag{3.45}$$

A equação em (3.45) é chamada *equação de Riccati discreta* (DRE-*discrete Riccati equation*), a qual se desenvolve para trás no tempo. A sequência $\mathbf{S}_k$ pode ser calculada *offline* apenas com o conhecimento das matrizes da planta e do índice de desempenho. Então, (3.41), com o estado inicial $\mathbf{x}_i$, produz a trajetória ótima. No entanto, não é conhecida ainda a entrada de controle ótima para mover a planta ao longo dessa trajetória. Para determinar o controle ótimo, de (3.35), (3.39) e (3.20), obtém-se

$$\begin{aligned} \mathbf{u}_k &= -\mathbf{R}_k^{-1} \mathbf{B}_k^\top \lambda_{k+1} && (3.46) \\ &= -\mathbf{R}_k^{-1} \mathbf{B}_k^\top \mathbf{S}_{k+1} \mathbf{x}_{k+1} && (3.47) \\ &= -\mathbf{R}_k^{-1} \mathbf{B}_k^\top \mathbf{S}_{k+1} (\mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k), && (3.48) \end{aligned}$$

assim

$$(\mathbf{I} + \mathbf{R}_k^{-1}\mathbf{B}_k^\top \mathbf{S}_{k+1}\mathbf{B}_k)\mathbf{u}_k = -\mathbf{R}_k^{-1}\mathbf{B}_k^\top \mathbf{S}_{k+1}\mathbf{A}_k\mathbf{x}_k \tag{3.49}$$

e

$$\mathbf{u}_k = -(\mathbf{B}_k^\top \mathbf{S}_{k+1}\mathbf{B}_k + \mathbf{R}_k)^{-1}\mathbf{B}_k^\top \mathbf{S}_{k+1}\mathbf{A}_k\mathbf{x}_k. \tag{3.50}$$

Então,

$$\mathbf{u}_k = -\mathbf{K}_k\mathbf{x}_k \tag{3.51}$$

onde $\mathbf{K}_k$ é a *sequência de ganhos de Kalman*

$$\mathbf{K}_k = (\mathbf{B}_k^\top \mathbf{S}_{k+1}\mathbf{B}_k + \mathbf{R}_k)^{-1}\mathbf{B}_k^\top \mathbf{S}_{k+1}\mathbf{A}_k. \tag{3.52}$$

Essas equações estão resumidas na Tabela 5 e um diagrama de blocos do esquema de controle aparece na Figura 9.

Observação 3.1.1 *Note que na Tabela 5 não foi feita nenhuma premissa de controlabilidade na planta. No entanto, independentemente das propriedades de controlabilidade da planta, o controle ótimo minimiza o índice de desempenho.*

Tabela 5 – Controle ótimo discreto LQR.

Modelo do sistema:
$\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k,\ k > i$
Índice de desempenho:
$J_i = \frac{1}{2}\mathbf{x}_N^\top \mathbf{S}_N\mathbf{x}_N + \frac{1}{2}\sum_{k=i}^{N-1}[\mathbf{x}_k^\top \mathbf{Q}_k\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}_k\mathbf{u}_k]$
Requisitos:
$\mathbf{S}_N \geq 0, \mathbf{Q}_k \geq 0, \mathbf{R}_k > 0$ e todas simétricas
Controle com realimentação ótima:
$\mathbf{S}_k = \mathbf{A}_k^\top\left[\mathbf{S}_{k+1} - \mathbf{S}_{k+1}\mathbf{B}_k(\mathbf{B}_k^\top \mathbf{S}_{k+1}\mathbf{B}_k + \mathbf{R}_k)^{-1}\mathbf{B}_k^\top \mathbf{S}_{k+1}\right]\mathbf{A}_k + \mathbf{Q}_k,\ k < N,\ \ \mathbf{S}_N$ dado
$\mathbf{K}_k = (\mathbf{B}_k^\top \mathbf{S}_{k+1}\mathbf{B}_k + \mathbf{R}_k)^{-1}\mathbf{B}_k^\top \mathbf{S}_{k+1}\mathbf{A}_k$
$\mathbf{u}_k = -\mathbf{K}_k\mathbf{x}_k$
$J_i^* = \frac{1}{2}\mathbf{x}_i^\top \mathbf{S}_i\mathbf{x}_i$

O sistema em malha fechada com realimentação ótima (3.51) é

$$\mathbf{x}_{k+1} = (\mathbf{A}_k - \mathbf{B}_k\mathbf{K}_k)\mathbf{x}_k, \tag{3.53}$$

que fornece uma alternativa para calcular a trajetória de estado ótima (3.41).

Se substituído (3.52) na equação de Riccati (3.45), obtém-se

$$\mathbf{S}_k = \mathbf{A}_k^\top \mathbf{S}_{k+1}\mathbf{A}_k - \mathbf{A}_k^\top \mathbf{S}_{k+1}\mathbf{B}_k\mathbf{K}_k + \mathbf{Q}_k \tag{3.54}$$

$$= \mathbf{A}_k^\top \mathbf{S}_{k+1}(\mathbf{A}_k - \mathbf{B}_k\mathbf{K}_k) + \mathbf{Q}_k, \tag{3.55}$$

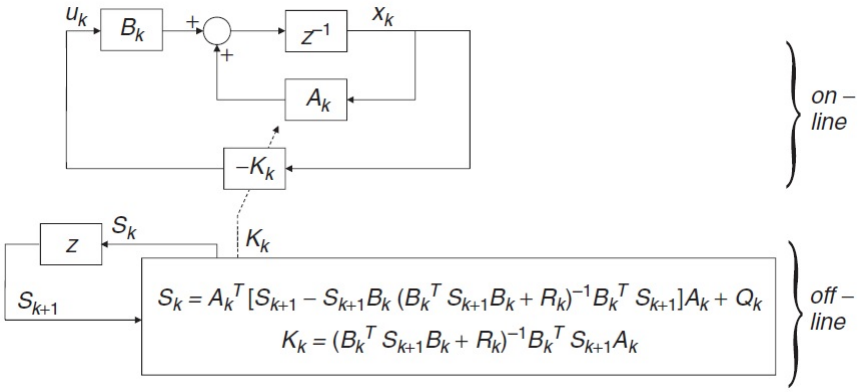


Figura 9 – Controle ótimo LQR em malha fechada (LEWIS; VRABIE; SYRMOS, 2012)

com $k < N$ e condição de contorno $\mathbf{S}_N$ dada como a matriz de ponderação do estado final em (3.21). A sequência $\mathbf{S}_k$ em (3.55) pode ser calculada *offline*, para trás no tempo, sabendo apenas as matrizes da planta (3.20) e as matrizes do índice de desempenho (3.21).

Portanto, uma alternativa mais eficiente de calcular $\mathbf{S}_k$ e $\mathbf{K}_k$ é usar a recursão

$$\mathbf{K}_k = (\mathbf{B}_k^\top \mathbf{S}_{k+1}\mathbf{B}_k + \mathbf{R}_k)^{-1}\mathbf{B}_k^\top \mathbf{S}_{k+1}\mathbf{A}_k \tag{3.56}$$

$$\mathbf{S}_k = \mathbf{A}_k^\top \mathbf{S}_{k+1}(\mathbf{A}_k - \mathbf{B}_k\mathbf{K}_k) + \mathbf{Q}_k. \tag{3.57}$$

Por outro lado, note que (3.56) é equivalente as seguintes equações

$$(\mathbf{B}_k^\top \mathbf{S}_{k+1}\mathbf{B}_k + \mathbf{R}_k)\mathbf{K}_k = \mathbf{B}_k^\top \mathbf{S}_{k+1}\mathbf{A}_k, \tag{3.58}$$

$$\mathbf{B}_k^\top \mathbf{S}_{k+1}(\mathbf{B}_k\mathbf{K}_k - \mathbf{A}_k) + \mathbf{R}_k\mathbf{K}_k = 0 \tag{3.59}$$

$$-\mathbf{K}_k^\top \mathbf{B}_k^\top \mathbf{S}_{k+1}(\mathbf{A}_k - \mathbf{B}_k\mathbf{K}_k) + \mathbf{K}_k^\top \mathbf{R}_k\mathbf{K}_k = 0. \tag{3.60}$$

Logo, de (3.57) e (3.60) obtém-se

$$\mathbf{S}_k = \mathbf{A}_k^\top \mathbf{S}_{k+1}(\mathbf{A}_k - \mathbf{B}_k\mathbf{K}_k) + \mathbf{Q}_k - \mathbf{K}_k^\top \mathbf{B}_k^T \mathbf{S}_{k+1}(\mathbf{A}_k - \mathbf{B}_k\mathbf{K}_k) + \mathbf{K}_k^\top \mathbf{R}_k\mathbf{K}_k \tag{3.61}$$

$$= (\mathbf{A}_k^\top \mathbf{S}_{k+1} - \mathbf{K}_k^\top \mathbf{B}_k^\top \mathbf{S}_{k+1})(\mathbf{A}_k - \mathbf{B}_k\mathbf{K}_k) + \mathbf{Q}_k + \mathbf{K}_k^\top \mathbf{R}_k\mathbf{K}_k \tag{3.62}$$

$$= (\mathbf{A}_k - \mathbf{B}_k\mathbf{K}_k)^\top \mathbf{S}_{k+1}(\mathbf{A}_k - \mathbf{B}_k\mathbf{K}_k) + \mathbf{Q}_k + \mathbf{K}_k^\top \mathbf{R}_k\mathbf{K}_k. \tag{3.63}$$

Portanto, a equação em (3.57) é equivalente a

$$\mathbf{S}_k = (\mathbf{A}_k - \mathbf{B}_k\mathbf{K}_k)^\top \mathbf{S}_{k+1}(\mathbf{A}_k - \mathbf{B}_k\mathbf{K}_k) + \mathbf{Q}_k + \mathbf{K}_k^\top \mathbf{R}_k\mathbf{K}_k. \tag{3.64}$$

A equação em (3.64) é chamada *versão estabilizada de Joseph da equação de Riccati.*

3.1.2 Simulação computacional

Considere a planta linear variante no tempo contínuo (PANG; BIAN; JIANG, 2019)

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t), \tag{3.65}$$

onde $\mathbf{x}(t) = [q\ \dot{q}]^\top$, $q = [\alpha\ \beta\ \gamma]^\top$, α é o ângulo *row*, β é o ângulo *pitch* e γ é o ângulo *yaw*,

$$\mathbf{A}(t) = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{D}(t) & -\mathbf{K}(t) \end{bmatrix}, \quad \mathbf{B}(t) = \begin{bmatrix} \mathbf{0} \\ \mathbf{L}(t) \end{bmatrix} \tag{3.66}$$

$$\mathbf{D}(t) = \begin{bmatrix} 0 & 0 & d(t) \\ 0 & 0 & 0 \\ -d(t) & 0 & 0 \end{bmatrix}, \quad \mathbf{K}(t) = \begin{bmatrix} k_1(t) & 0 & 0 \\ 0 & k_2(t) & 0 \\ 0 & 0 & k_3(t) \end{bmatrix} \tag{3.67}$$

$$\mathbf{L}(t) = \begin{bmatrix} \frac{1}{J_{\mathbf{x}}(t)} & 0 & 0 \\ 0 & \frac{1}{J_{\mathbf{y}}(t)} & 0 \\ 0 & 0 & \frac{1}{J_{\mathbf{z}}(t)} \end{bmatrix}, \quad \begin{cases} d(t) = \dfrac{\omega_0(J_{\mathbf{y}}(t)) - J_{\mathbf{x}}(t) - J_{\mathbf{z}}(t))}{J_{\mathbf{x}}(t)} \\ k_1(t) = \dfrac{4\omega_0^2(J_{\mathbf{y}}(t)) - J_{\mathbf{z}}(t))}{J_{\mathbf{x}}(t)} \\ k_2(t) = \dfrac{3\omega_0^2(J_{\mathbf{x}}(t)) - J_{\mathbf{z}}(t))}{J_{\mathbf{y}}(t)} \\ k_3(t) = \dfrac{\omega_0^2(J_{\mathbf{y}}(t)) - J_{\mathbf{x}}(t))}{J_{\mathbf{z}}(t)} \end{cases}, \tag{3.68}$$

$$\begin{cases} J_{\mathbf{x}}(t) = 1070 + 15.5t \\ J_{\mathbf{y}}(t) = 2150 - 11t \\ J_{\mathbf{z}(t)} = 1300 - 7.5t. \end{cases} . \tag{3.69}$$

As variáveis $J_{\mathbf{x}}(t)$, $J_{\mathbf{y}}(t)$, $J_{\mathbf{z}}(t)$ são as componentes de momentos de inércia da aeronave em relação ao sistema de coordenadas e $\omega_0 = 0.0011$ rad/seg é a taxa orbital. O sistema (3.65) é discretizado com tempo de amostra $h = 0.1$s e obtém-se o sistema linear discreto variante no tempo

$$\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k \tag{3.70}$$

onde

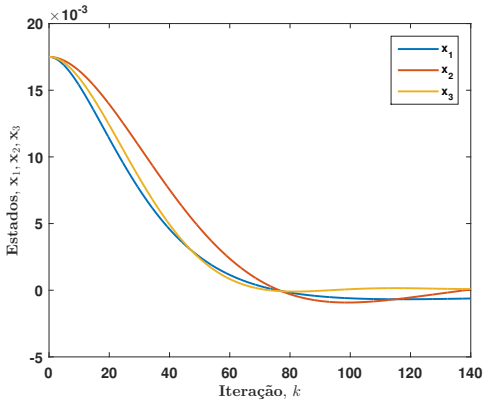
$$\mathbf{A}_k = \mathbf{I} + \mathbf{A}(kh)h,\ \mathbf{B}_k = \mathbf{B}(kh)h. \tag{3.71}$$

O tempo final é $N = 1400$, as matrizes $\mathbf{Q}_k = 10\mathbf{I}_6$ e $\mathbf{R}_k = \mathbf{I}_6$, para todo k, onde $\mathbf{I}_6$ é a matriz identidade de ordem 6. A condição inicial é $q_0 = [0,0175\quad 0,0175\quad 0,0175]^\top$, $\dot{q}_0 = [0\quad 0\quad 0]^\top$. A Figura 10 mostra a trajetória ótima do sistema, a entrada de controle ótimo e o custo ótimo.

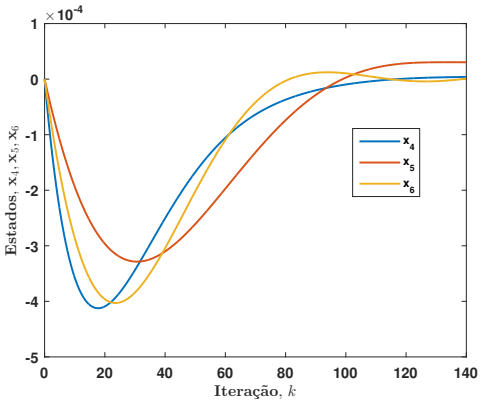
3.2 Controle em malha fechada em estado estacionário

O sistema em malha fechada com realimentação ótima (3.51) para um sistema linear invariante no tempo é

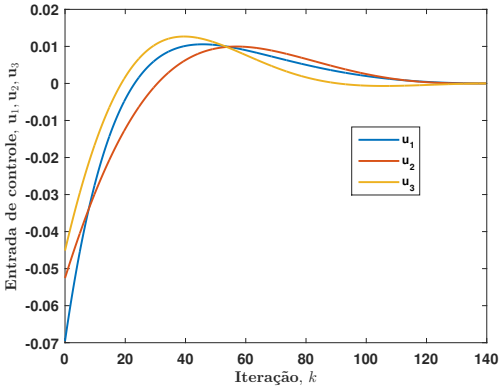
$$\mathbf{x}_{k+1} = (\mathbf{A} - \mathbf{B}\mathbf{K}_k)\mathbf{x}_k, \tag{3.72}$$



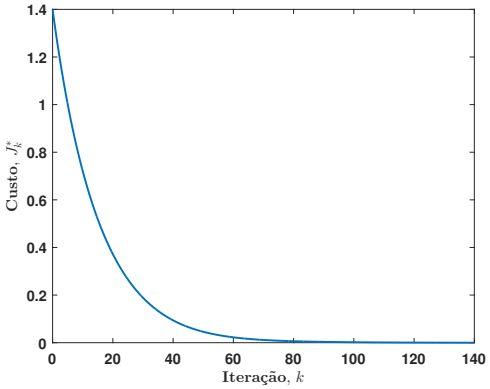
(a) Trajetória dos ângulos da aeronave.



(b) Trajetória das velocidades dos ângulos da aeronave.



(c) Entrada de controle ótimo.



(d) Custo ótimo J_k^*.

Figura 10 – Resultados da simulação para o sistema com matrizes em (3.66). (a) Trajetória dos ângulos da aeronave $q = [x_1\ x_2\ x_3]$. (b) Trajetória das velocidades dos ângulos da aeronave $\dot{q} = [x_4\ x_5\ x_6]$. (c) Entrada de controle ótimo. (d) Custo ótimo.

onde a sequência de ganho $\mathbf{K}_k$ é dada em função da sequência $\mathbf{S}_k$ da solução da equação de Riccati

$$\mathbf{S}_k = \mathbf{A}^\top \left[\mathbf{S}_{k+1} - \mathbf{S}_{k+1}\mathbf{B}(\mathbf{B}^\top \mathbf{S}_{k+1}\mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^\top \mathbf{S}_{k+1}\right]\mathbf{A} + \mathbf{Q} \tag{3.73}$$

$$\mathbf{K}_k = (\mathbf{B}^\top \mathbf{S}_{k+1}\mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^\top \mathbf{S}_{k+1}\mathbf{A}. \tag{3.74}$$

O sistema em malha fechada (3.72) é variante no tempo, já que $\mathbf{K}_k$ é variante no tempo. Essa realimentação variante no tempo nem sempre é conveniente para ser implementada; ela requer o armazenamento de uma sequência de matrizes em $\mathbb{R}^{m\times n}$ (a sequência de matrizes de ganho de controle (3.74)). Ao invés do ganho variante no tempo (3.74), é de interesse usar um ganho de realimentação sub-ótimo que não minimiza o índice de desempenho, mas é um ganho constante

$$\mathbf{u}_k = -\mathbf{K}\mathbf{x}_k. \tag{3.75}$$

A realimentação (3.75) é mais fácil de implementar que (3.51). Um candidato a ganho de realimentação constante pode ser o limite de $\mathbf{K}_k$ quando o tempo final N tende a infinito (ou equivalentemente quando $k \to -\infty$) (LEWIS; VRABIE; SYRMOS, 2012, p. 65).

3.2.1 Realimentação sub-ótima e equação de Lyapunov

Seja a planta invariante no tempo

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k. \tag{3.76}$$

Será usada a realimentação de estados

$$\mathbf{u}_k = -\mathbf{K}_k\mathbf{x}_k \tag{3.77}$$

para uma sequência arbitrária de matrizes $\mathbf{K}_k$ e vai se determinar o valor resultante do índice de desempenho

$$J_i = \frac{1}{2}\mathbf{x}_N^\top\mathbf{S}_N\mathbf{x}_N + \frac{1}{2}\sum_{k=i}^{N-1}[\mathbf{x}_k^\top\mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top\mathbf{R}\mathbf{u}_k]. \tag{3.78}$$

Note que

$$\frac{1}{2}\sum_{k=i}^{N-1}[\mathbf{x}_{k+1}^\top\mathbf{S}_{k+1}\mathbf{x}_{k+1} + \mathbf{x}_k^\top(\mathbf{Q} - \mathbf{S}_k + \mathbf{K}_k^\top\mathbf{R}\mathbf{K}_k)\mathbf{x}_k] = J_i - \frac{1}{2}\mathbf{x}_i^\top\mathbf{S}_i\mathbf{x}_i. \tag{3.79}$$

Levando em conta a equação de estado (3.76) e o controle (3.77), obtém-se

$$J_i = \frac{1}{2}\mathbf{x}_i^\top\mathbf{S}_i\mathbf{x}_i + \frac{1}{2}\sum_{k=i}^{N-1}\mathbf{x}_k^\top[(\mathbf{A} - \mathbf{B}\mathbf{K}_k)^\top\mathbf{S}_{k+1}(\mathbf{A} - \mathbf{B}\mathbf{K}_k) + \mathbf{Q} - \mathbf{S}_k + \mathbf{K}_k^\top\mathbf{R}\mathbf{K}_k]\mathbf{x}_k. \tag{3.80}$$

Suponha que $\mathbf{S}_k$ satisfaz (3.64), ou seja

$$\mathbf{S}_k = (\mathbf{A} - \mathbf{B}\mathbf{K}_k)^\top\mathbf{S}_{k+1}(\mathbf{A} - \mathbf{B}\mathbf{K}_k) + \mathbf{Q} + \mathbf{K}_k^\top\mathbf{R}\mathbf{K}_k, \tag{3.81}$$

com condição de contorno $\mathbf{S}_N$. Então

$$J_i = \frac{1}{2}\mathbf{x}_i^\top\mathbf{S}_i\mathbf{x}_i. \tag{3.82}$$

Portanto, para todo ganho $\mathbf{K}_k$, o custo resultante em $[k, N]$ é dado em cada tempo k por

$$J_k = \frac{1}{2}\mathbf{x}_k^\top\mathbf{S}_k\mathbf{x}_k. \tag{3.83}$$

A equação em (3.81) é chamada *equação de Lyapunov*. A equação de Lyapunov torna-se a versão estabilizada de Joseph da equação de Riccati somente quando a sequência de ganho ótima (3.56) é usada.

O resumo para el a solução do problema do LQR com realimentação sub-ótima é dada na Tabela 6.

3.2.2 Equação algébrica de Riccati

A equação de Riccati (3.73) é resolvida para trás no tempo começando no tempo N. Se $\mathbf{S}_k \to \mathbf{S}_\infty$, quando $k \to -\infty$, então de (3.73) segue-se que

$$\mathbf{S}_\infty = \mathbf{A}^\top \left[\mathbf{S}_\infty - \mathbf{S}_\infty \mathbf{B}(\mathbf{B}^\top \mathbf{S}_\infty \mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^\top \mathbf{S}_\infty\right]\mathbf{A} + \mathbf{Q}. \tag{3.84}$$

A equação

$$\mathbf{S} = \mathbf{A}^\top \left[\mathbf{S} - \mathbf{S}\mathbf{B}(\mathbf{B}^\top \mathbf{S}\mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^\top \mathbf{S}\right]\mathbf{A} + \mathbf{Q}. \tag{3.85}$$

é chamada *equação algébrica de Riccati discreta* (DARE-*discrete algebraic Riccati equation*). Portanto, $\mathbf{S}_\infty$ é uma solução da equação em (3.85). No entanto, a equação (3.85) pode ter várias soluções.

O ganho correspondente a $\mathbf{S}_\infty$ é o *ganho de Kalman em estado estacionário*

$$\mathbf{K}_\infty = (\mathbf{B}^\top \mathbf{S}_\infty \mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^\top \mathbf{S}_\infty \mathbf{A}. \tag{3.86}$$

De (3.83) tem-se que o custo resultante associado ao controle (3.86) é

$$J_k = \frac{1}{2}\mathbf{x}_k^\top \mathbf{S}_k \mathbf{x}_k. \tag{3.87}$$

onde $\mathbf{S}_k$ satisfaz a equação de Lyapunov

$$\mathbf{S}_k = (\mathbf{A} - \mathbf{B}\mathbf{K}_\infty)^\top \mathbf{S}_{k+1}(\mathbf{A} - \mathbf{B}\mathbf{K}_\infty) + \mathbf{Q} + \mathbf{K}_\infty^\top \mathbf{R}\mathbf{K}_\infty \tag{3.88}$$

com condição de contorno $\mathbf{S}_N$.

Os seguintes teoremas estabelecem condições para a existência de solução única da DARE (LEWIS; VRABIE; SYRMOS, 2012, p. 71-72)

Teorema 3.2.1 *Suponha que* $(\mathbf{A},\mathbf{B})$ *é estabilizável. Então, para cada* $\mathbf{S}_N$ *simétrica e positiva semidefinida, no limite existe uma solução* $\mathbf{S}_\infty$ *para (3.73). Além disso,* $\mathbf{S}_\infty$ *é uma solução positiva semidefinida para a equação algébrica de Riccati (3.85).*

Teorema 3.2.2 *Seja* $\mathbf{Q} = \mathbf{C}^\top\mathbf{C} \geq 0$ *e* $\mathbf{R} > 0$. *Suponha que* $(\mathbf{A},\mathbf{C})$ *é observável. Então,* $(\mathbf{A},\mathbf{B})$ *é estabilizável se, e somente se,*

(a) No limite, existe uma única solução positiva definida $\mathbf{S}_\infty$ *da equação de Riccati (3.73). Além disso,* $\mathbf{S}_\infty$ *é a única solução positiva definida da equação algébrica de Ricccati (3.85).*

(b) O sistema em malha fechada

$$\mathbf{x}_{k+1} = (\mathbf{A} - \mathbf{B}\mathbf{K}_\infty)\mathbf{x}_k \tag{3.89}$$

é assintoticamente estável, ou seja, os autovalores da matriz $\mathbf{A} - \mathbf{B}\mathbf{K}_\infty$ *pertencem ao círculo unitário (CHEN, 1999, p. 131), sendo* $\mathbf{K}_\infty$ *como em (3.86).*

Tabela 6 – Controle sub-ótimo discreto LQR.

Modelo do sistema:
$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k,\ k > i$

Índice de desempenho:
$J_i = \frac{1}{2}\mathbf{x}_N^\top \mathbf{S}_N \mathbf{x}_N + \frac{1}{2}\sum_{k=i}^{N-1}\left[\mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k\right]$

Requisitos:
$\mathbf{S}_N \geq 0, \mathbf{Q} \geq 0, \mathbf{R} > 0$ e todas simétricas

Cálculo do ganho:
$\mathbf{S}_\infty = \mathbf{A}^\top\left[\mathbf{S}_\infty - \mathbf{S}_\infty\mathbf{B}(\mathbf{B}^\top\mathbf{S}_\infty\mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^\top\mathbf{S}_\infty\right]\mathbf{A} + \mathbf{Q}$ (solução DARE)
$\mathbf{K}_\infty = (\mathbf{B}^\top\mathbf{S}_\infty\mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^\top\mathbf{S}_\infty\mathbf{A}$ (ganho de Kalman em estado estacionário)

Controle com realimentação sub-ótimo:
$\mathbf{S}_k = (\mathbf{A} - \mathbf{B}\mathbf{K}_\infty)^\top\mathbf{S}_{k+1}(\mathbf{A} - \mathbf{B}\mathbf{K}_\infty) + \mathbf{Q} + \mathbf{K}_\infty^\top\mathbf{R}\mathbf{K}_\infty,\ k < N,\ \ \mathbf{S}_N$ dado
$\mathbf{u}_k = -\mathbf{K}_\infty\mathbf{x}_k$
$J_k = \frac{1}{2}\mathbf{x}_k^\top\mathbf{S}_k\mathbf{x}_k$

3.2.3 Simulação computacional

Seja a planta escalar (LEWIS; VRABIE; SYRMOS, 2012, p. 66)

$$x_{k+1} = ax_k + bu_k \tag{3.90}$$

com índice de desempenho

$$J_0 = \frac{1}{2}s_N x_N^2 + \frac{1}{2}\sum_{k=0}^{N-1}(qx_k^2 + ru_k^2). \tag{3.91}$$

O controle ótimo é uma realimentação variante no tempo

$$u_k = -K_k x_k, \tag{3.92}$$

com ganho determinado pela equação de Riccati (3.73) e (3.74)

$$s_k = \frac{a^2 r s_{k+1}}{b^2 s_{k+1} + r} + q, \tag{3.93}$$

$$K_k = \frac{abs_{k+1}}{b^2 s_{k+1} + r}. \tag{3.94}$$

Para os parâmetros $a = 1,05, b = 0,01, q = r = s_N = 5$ e estado final $N = 100$, obtém-se

$$s_\infty = s_0 = 5151,7 \tag{3.95}$$

$$K_\infty = K_0 = 9,808. \tag{3.96}$$

A correspondente solução da equação de Riccati s_k^* é mostrada na Figura 11 (a). Aplicando o ganho (3.92) com $x_0 = 10$, a trajetória ótima x_k^* é mostrada na Figura 11 (b). O custo ótimo é dado por

$$J_k^* = \frac{1}{2} s_k^* (x_k^*)^2 \tag{3.97}$$

que é mostrado na Figura 11 (c).

Por outro lado, o controle sub-ótimo é a realimentação invariante no tempo

$$u_k = -9.808 x_k, \tag{3.98}$$

sendo o custo dado por

$$J_k = \frac{1}{2} s_k x_k^2, \tag{3.99}$$

na qual s_k satisfaz a equação de Lyapunov (3.88), que nesse caso torna-se

$$s_k = (a - bK_\infty)^2 s_{k+1} + rK_\infty^2 + q. \tag{3.100}$$

A Figura 11 (a) mostra a sequência em (3.100). Aplicando o ganho sub-ótimo (3.98) com $x_0 = 10$, a trajetória sub-ótima x_k é mostrada na Figura 11 (b). O custo (3.99) é mostrado na Figura (11) (c).

3.2.4 Controle ótimo LQR com horizonte infinito

Seja o índice de desempenho associado a (3.76)

$$J_0 = \frac{1}{2} \sum_{k=0}^{\infty} [\mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k] \tag{3.101}$$

com $(\mathbf{A},\mathbf{B})$ estabilizável e $(\mathbf{A},\sqrt{\mathbf{Q}})$ observável. Então, os Teoremas 3.2.1 e 3.2.2 garantem que, para cada $\mathbf{S}_N$, existe a solução no limite $\mathbf{S}_\infty$ de (3.73) que é a única solução definida positiva da equação algébrica de Riccati (3.85).

Então, o controle de realimentação em estado estacionário

$$\mathbf{u}_k = -\mathbf{K}_\infty \mathbf{x}_k \tag{3.102}$$

no qual

$$\mathbf{K}_\infty = (\mathbf{B}^\top \mathbf{S}_\infty \mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^\top \mathbf{S}_\infty \mathbf{A}, \tag{3.103}$$

e $\mathbf{S}_\infty$ satisfaz a DARE

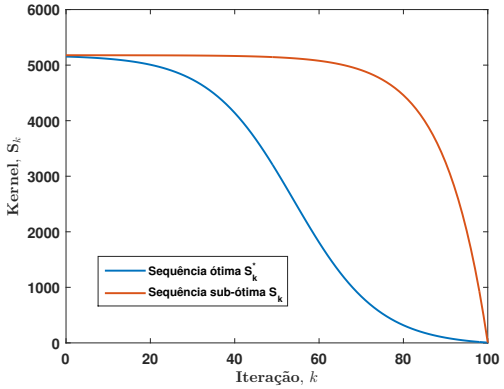
$$\mathbf{S}_\infty = \mathbf{A}^\top \left[\mathbf{S}_\infty - \mathbf{S}_\infty \mathbf{B} (\mathbf{B}^\top \mathbf{S}_\infty \mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^\top \mathbf{S}_\infty \right] \mathbf{A} + \mathbf{Q}, \tag{3.104}$$

é o controle ótimo que minimiza (3.101) sobre o intervalo $[0, +\infty]$. O custo ótimo em cada instante de tempo é

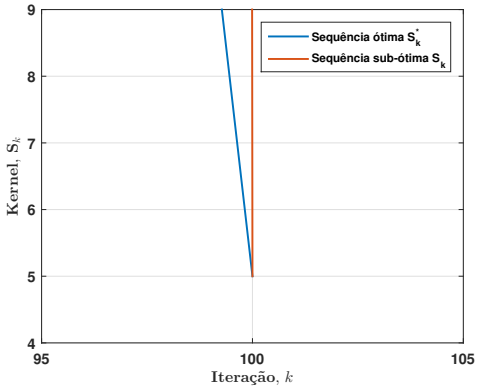
$$J_k^* = \frac{1}{2} \mathbf{x}_k^\top \mathbf{S}_\infty \mathbf{x}_k. \tag{3.105}$$

Essas equações são resumidas na Tabela 7.

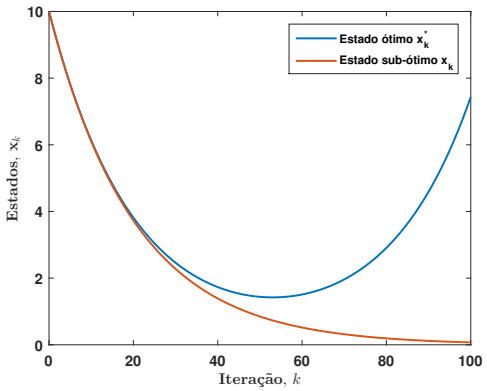
A Tabela 8 mostra um resumo sobre os diferentes tipos de matrizes de ganho e kernel para o problema de controle ótimo LQR discreto.



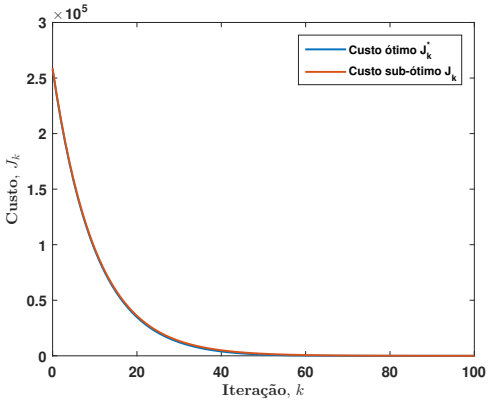
(a) Trajetória do kernel com $s_N = 5$



(b) Trajetória do kernel com $s_N = 5$



(c) Trajetória ótima x_k^* e sub-ótima x_k



(d) Custo ótimo J_k^* e sub-ótimo J_k

Figura 11 – Comportamento em malha fechada do controle ótimo e sub-ótimo do sistema em (3.90). (a)-(b) Custo do kernel. (b) Trajetórias do estado. (c) Custos.

3.2.5 Solução iterativa da DARE

Note que para resolver o problema do LQR com horizonte infinito é necessária a solução $\mathbf{S}_\infty$ da equação algébrica de Riccati (3.85) que é uma equação não linear em $\mathbf{S}$. Uma maneira de encontrar $\mathbf{S}_\infty$ é iterar para trás no tempo a equação de Riccati (3.73) até obter $\mathbf{S}_0 = \mathbf{S}_\infty$. Outra maneira de obter $\mathbf{S}_\infty$ é com método de Schur (LAUB, 1979).

Um método iterativo para achar a solução da equação de Riccati $\mathbf{S}_\infty$ e o ganho de realimentação ótimo $\mathbf{K}_\infty$ é dado no Algoritmo 8 (LEWIS; VRABIE, 2009), também conhecido como método de Hewer (HEWER, 1971). O método de Hewer começa com uma matriz de ganho admissível $\mathbf{K}(0)$, ou seja, tal que o sistema em malha fechada $\mathbf{x}_{k+1} = (\mathbf{A} - \mathbf{B}\mathbf{K}(0))\mathbf{x}_k$ é assintoticamente estável.

O Algoritmo 8 em cada iteração resolve uma equação de Lyapunov e encontra uma matriz de ganho até $\mathbf{S}(j)$ e $\mathbf{K}(j+1)$ convergir para $\mathbf{S}_\infty$ e $\mathbf{K}_\infty$, respectivamente.

Tabela 7 – Controle ótimo discreto LQR com horizonte infinito.

Modelo do sistema:
$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k,\ k \geq 0$

Índice de desempenho:
$J_0 = \frac{1}{2}\sum_{k=0}^{\infty}[\mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k]$

Requisitos:
$\mathbf{Q} \geq 0, \mathbf{R} > 0$ e todas simétricas

Cálculo do ganho:
$\mathbf{S}_\infty = \mathbf{A}^\top\left[\mathbf{S}_\infty - \mathbf{S}_\infty\mathbf{B}(\mathbf{B}^\top\mathbf{S}_\infty\mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^\top\mathbf{S}_\infty\right]\mathbf{A} + \mathbf{Q}$ (solução DARE)
$\mathbf{K}_\infty = (\mathbf{B}^\top\mathbf{S}_\infty\mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^\top\mathbf{S}_\infty\mathbf{A}$ (ganho de Kalman em estado estacionário)

Controle com realimentação ótimo:
$\mathbf{u}_k = -\mathbf{K}_\infty\mathbf{x}_k$
$J_k^* = \frac{1}{2}\mathbf{x}_k^\top\mathbf{S}_\infty\mathbf{x}_k$

Tabela 8 – Matrizes de ganho e kernel para os diferentes tipos de controle.

Tipo de controle	Matriz de ganho $\mathbf{K}_k$	Kernel $\mathbf{S}_k$
Controle ótimo	Variante no tempo (3.56)	Variante no tempo (3.57)
Controle sub-ótimo	Invariante no tempo (3.86)	Variante no tempo (3.88)
Horizonte infinito	Invariante no tempo (3.103)	Invariante no tempo (3.104)

Algoritmo 8

1: **Input:** Matriz de ganho admissível $\mathbf{K}(0)$
2: **for** $j = 0, 1, \ldots$ **do**
3: $\quad\mathbf{S}(j) = (\mathbf{A} - \mathbf{B}\mathbf{K}(j))^\top\mathbf{S}(j)(\mathbf{A} - \mathbf{B}\mathbf{K}(j)) + \mathbf{Q} + \mathbf{K}^\top(j)\mathbf{R}\mathbf{K}(j)$
4: $\quad\mathbf{K}(j+1) = (\mathbf{B}^\top\mathbf{S}(j)\mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^\top\mathbf{S}(j)\mathbf{A}$
5: **end for**

3.2.6 Simulação computacional

Seja a planta linear invariante no tempo (NETO; RêGO, 2014)

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \tag{3.106}$$

onde

$$\mathbf{A} = \begin{bmatrix} 0,912 & 0,083 & -0,001 \\ 0,372 & 0,943 & -0,010 \\ 0,000 & 0,000 & 0,368 \end{bmatrix},\ \mathbf{B} = \begin{bmatrix} 0,000 & 0,043 \\ -0,006 & 0,968 \\ 0,632 & 0,000 \end{bmatrix}. \tag{3.107}$$

Além $\mathbf{Q} = \mathbf{I}_3^\top\mathbf{I}_3$ e $\mathbf{R} = \mathbf{I}_2$, na qual $\mathbf{I}_3$ e $\mathbf{I}_2$ são as matrizes identidades de ordem 3 e 2 respectivamente. O estado inicial é $\mathbf{x}_0 = [3\ 5\ 2]^\top$. O sistema (3.106) com as matrizes em

(3.107) é estabilizável já que os autovalores da matriz $\mathbf{A} - \mathbf{BK}$, na qual

$$\mathbf{K} = \begin{bmatrix} -0,0005 & -0,0011 & 0,4655 \\ 0,3843 & 0,8817 & -0,0070 \end{bmatrix}, \tag{3.108}$$

têm norma menor que 1 (CHEN, 1999, p. 131). Além disso, $(\mathbf{A}, \mathbf{I}_3)$ é observável já que a matriz

$$\begin{bmatrix} \mathbf{I}_3 \\ \mathbf{I}_3\mathbf{A} \\ \mathbf{I}_3\mathbf{A}^2 \end{bmatrix} \in \mathbb{R}^{9\times 3} \tag{3.109}$$

tem posto igual a 3 (CHEN, 1999, p. 171).

Para $\mathbf{S}_N = \mathbf{I}_3$ e $N = 100$, iterando (3.73) para trás no tempo, obtém-se

$$\mathbf{S}_\infty = \mathbf{S}_0 = \begin{bmatrix} 5,3735 & 0,7396 & -0,0100 \\ 0,7396 & 1,6203 & -0,0067 \\ -0,0100 & -0,0067 & 1,1038 \end{bmatrix}. \tag{3.110}$$

Portanto, o ganho de controle ótimo é

$$\mathbf{K}_\infty = (\mathbf{B}^\top\mathbf{S}_\infty\mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^\top\mathbf{S}_\infty\mathbf{A} \tag{3.111}$$

$$= \begin{bmatrix} -0,0049 & -0,0038 & 0,1782 \\ 0,5633 & 0,6130 & -0,0066 \end{bmatrix}. \tag{3.112}$$

A Figura 12 mostra o comportamento do sistema em (3.106). A trajetória ótima dos estados do sistema (3.106), aplicando o controlador ótimo

$$\mathbf{u}_k = -\mathbf{K}_\infty\mathbf{x}_k, \tag{3.113}$$

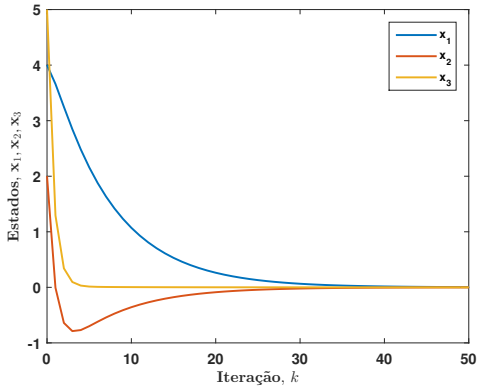
sendo $\mathbf{K}_\infty$ como em (3.112), é mostrada na Figura 12-(a). A Figura 12-(b) mostra a sequência $\mathbf{u}_k$ de ações de controle ótimo em (3.113). A Figura 12-(c) mostra o comportamento da sequência escalar de custo ótimo

$$J_k^* = \frac{1}{2}\mathbf{x}_k^\top\mathbf{S}_\infty\mathbf{x}_k, \tag{3.114}$$

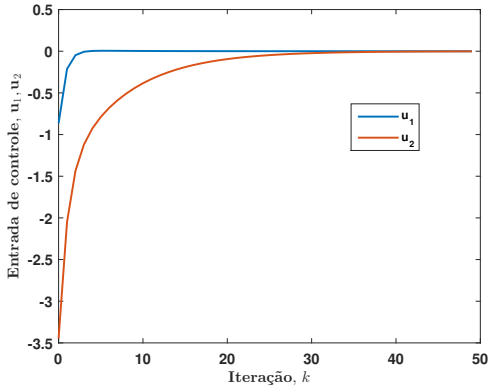
sendo $\mathbf{S}_\infty$ como em (3.110).

Observação 3.2.1 *Na presente seção e nos seguintes capítulos, para os resultados teóricos e experimentais, será usada a norma vetorial em $\mathbb{R}^n$ definida da seguinte forma: para cada vetor $\mathbf{x} = [x_1\ x_2\ \cdots\ x_n]^\top \in \mathbb{R}^n$*

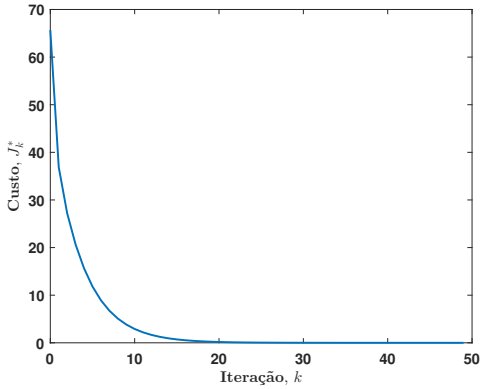
$$\|\mathbf{x}\| = \sqrt{\mathbf{x}^\top\mathbf{x}} = \sqrt{\sum_{i=1}^{n} x_i^2}. \tag{3.115}$$



(a) Trajetória ótima dos estados



(b) Entrada de controle ótimo



(c) Custo ótimo J_k^*

Figura 12 – Comportamento em malha fechada do controle ótimo com horizonte infinito do sistema em (3.106). (a) Trajetória ótima dos estado. (b) Entradas de controle ótimo. (c) Custo ótimo.

Além disso, para cada matriz $\mathbf{A} \in \mathbb{R}^{m\times n}$*, será usada a norma matricial definida por*

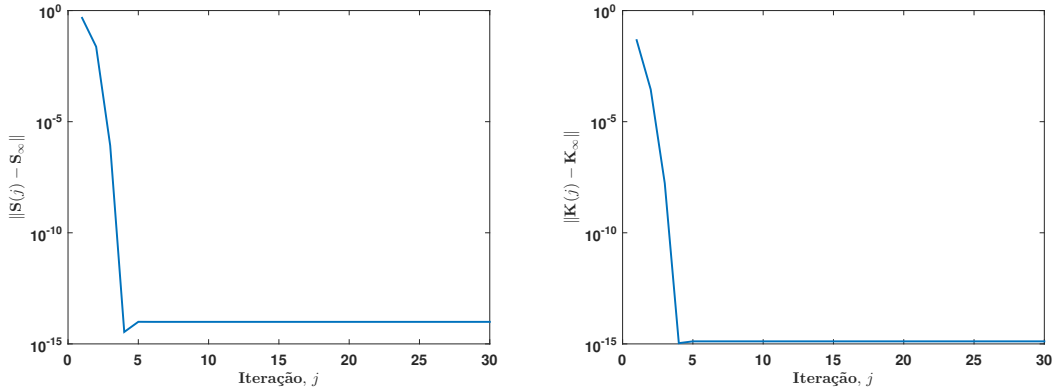
$$\|\mathbf{A}\| = \sup_{\mathbf{x}\neq 0} \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|}. \tag{3.116}$$

A norma da matriz $\mathbf{A}$ *em (3.116) é igual ao maior valor singular da matriz* $\mathbf{A}$ *(TREFETHEN; III, 1997, p. 34).*

A matriz $\mathbf{S}_\infty$ em (3.110) foi calculada para trás no tempo e logo a matriz $\mathbf{K}_\infty$ foi calculada como em (3.112). A seguir será usado o Algoritmo iterativo 8 para calcular as matrizes $\mathbf{S}_\infty$ e $\mathbf{K}_\infty$. A Figura 13-(a) mostra a convergência de $\mathbf{S}(j)$, enquanto que a Figura 13-(b) mostra a convergência de $\mathbf{K}(j)$, no Algoritmo 8, sendo $\mathbf{K}(0)$ como em (3.108).

3.3 Considerações finais

Neste capítulo, apresentou-se a teoria clássica de controle ótimo LQR. Verifica-se que a equação de Riccati e a equação algébrica de Riccati, para resolver o problema de



(a) Convergência de $\|\mathbf{S}(j) - \mathbf{S}_\infty\|$ (b) Convergência de $\|\mathbf{K}(j) - \mathbf{K}_\infty\|$

Figura 13 – Convergência de $\mathbf{S}(j)$ e $\mathbf{K}(j)$ no Algoritmo 8 para o sistema em (3.106). A matriz de ganho inicial $\mathbf{K}(0)$ é como em (3.108).

controle ótimo e sub-ótimo, respectivamente, dependem das matrizes do modelo do sistema. No entanto, em alguns casos, é difícil ou custoso obter as matrizes do modelo do sistema. Além do mais, em projetos de controle *online*, não é possível resolver o problema do LQR via equações de Riccati. Portanto, nesses casos, para resolver o problema de controle ótimo é necessário uma abordagem livre de modelo. A abordagem para o problema do LQR livre de modelo, será abordado no Capítulo 5.

4 Controle ótimo via programação dinâmica

No Capítulo 3, foi utilizado cálculo variacional para resolver o problema de controle ótimo LQR. Uma alternativa para esse mesmo problema é utilizar o princípio de otimalidade de Bellman (BELLMAN, 1957, p. 83), o qual estabelece que uma política ótima tem a seguinte propriedade: independentemente das decisões tomadas anteriormente, as decisões restantes devem constituir uma política ótima com respeito ao estado resultante das decisões anteriores.

Em outras palavras, se $\mathbf{u}_0^*, \cdots, \mathbf{u}_{N-1}^*$ é uma política ótima e é aplicada uma política arbitrária $\mathbf{u}_0, \cdots, \mathbf{u}_k$ $(k < N-1)$ com estado resultante $\mathbf{x}_{k+1}$, então as decisões restantes $\mathbf{u}_{k+1}^*, \cdots, \mathbf{u}_{N-1}^*$ constituem uma política ótima a partir de $\mathbf{x}_{k+1}$ (Figura 14).

$$\begin{array}{cccccccccc} & \mathbf{u}_0 & \mathbf{u}_1 & \cdots & \mathbf{u}_{k-1} & \mathbf{u}_k & \mathbf{u}_{k+1}^* & \cdots & \mathbf{u}_{N-1}^* \\ & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow & & \downarrow \\ \mathbf{x}_0 & \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_k & \mathbf{x}_{k+1} & \mathbf{x}_{k+2}^* & \cdots & \mathbf{x}_N^* \end{array}$$

Figura 14 – As decisões $\mathbf{u}_{k+1}^*, \cdots, \mathbf{u}_{N-1}^*$ constituem uma política ótima a partir de $\mathbf{x}_{k+1}$.

Seja a planta geral não linear variante no tempo

$$\mathbf{x}_{k+1} = \mathbf{f}^k(\mathbf{x}_k, \mathbf{u}_k) \tag{4.1}$$

e seja o índice de desempenho escalar associado à planta (4.1)

$$J_i = \phi(N, \mathbf{x}_N) + \sum_{k=i}^{N-1} L^k(\mathbf{x}_k, \mathbf{u}_k), \tag{4.2}$$

sendo $[i, N]$ o intervalo de interesse.

Suponha que é conhecido o custo ótimo $J_{k+1}^*(\mathbf{x}_{k+1})$ desde $k+1$ até N para todo os estados possíveis $\mathbf{x}_{k+1}$, e que também achamos a sequência de controle ótima desde $k+1$ até $N-1$ para todo $\mathbf{x}_{k+1}$

$$\mathbf{u}_{k+1}^*, \cdots, \mathbf{u}_{N-1}^*. \tag{4.3}$$

Por tanto

$$J_{k+1}^*(\mathbf{x}_{k+1}) = \phi(N, x_N) + \sum_{k=k+1}^{N-1} L^k(\mathbf{x}_k, \mathbf{u}_k^*). \tag{4.4}$$

Se implementar um controle arbitrário $\mathbf{u}_k$ no tempo k e usa-se a sequência de controle

ótimo (4.3), então o custo resultante é

$$J_k(\mathbf{x}_k) = \phi(N, \mathbf{x}_N) + \sum_{k=k}^{N-1} L^k(\mathbf{x}_k, \mathbf{u}_k^*) \tag{4.5}$$

$$= L^k(\mathbf{x}_k, \mathbf{u}_k) + \phi(N, x_N) + \sum_{k=k+1}^{N-1} L^k(\mathbf{x}_k, \mathbf{u}_k^*) \tag{4.6}$$

$$= L^k(\mathbf{x}_k, \mathbf{u}_k) + J_{k+1}^*(\mathbf{x}_{k+1}), \tag{4.7}$$

assim

$$J_k(\mathbf{x}_k) = L^k(\mathbf{x}_k, \mathbf{u}_k) + J_{k+1}^*(\mathbf{x}_{k+1}). \tag{4.8}$$

Então, pelo princípio de Bellman, o custo ótimo desde o tempo k é igual a

$$J_k^*(\mathbf{x}_k) = \min_{\mathbf{u}_k} \left(L^k(\mathbf{x}_k, \mathbf{u}_k) + J_{k+1}^*(\mathbf{x}_{k+1}) \right), \tag{4.9}$$

e o controle ótimo é

$$\mathbf{u}_k^* = \arg\min_{\mathbf{u}_k} \left(L^k(\mathbf{x}_k, \mathbf{u}_k) + J_{k+1}^*(\mathbf{x}_{k+1}) \right). \tag{4.10}$$

A equação em (4.9) é o princípio de otimalidade para sistemas discretos (LEWIS; VRABIE; SYRMOS, 2012, p. 264).

4.1 Controle ótimo LQR

Nesta seção será obtida a versão estabilizada de Joseph da equação de Riccati (3.64) e o ganho de Kalman (3.56) usando o princípio de otimalidade para sistemas discretos (4.9).

Seja a planta linear

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k, \tag{4.11}$$

onde $\mathbf{A}_k \in \mathbb{R}^{n \times n}$ e $\mathbf{B}_k \in \mathbb{R}^{n \times m}$. Seja o índice de desempenho associado à planta (4.11)

$$J_i = \frac{1}{2} \mathbf{x}_N^\top \mathbf{S}_N \mathbf{x}_N + \frac{1}{2} \sum_{k=i}^{N-1} \left[\mathbf{x}_k^\top \mathbf{Q}_k \mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}_k \mathbf{u}_k \right] \tag{4.12}$$

definido ao longo do intervalo de tempo discreto $[i, N]$, onde $\mathbf{Q}_k$ e $\mathbf{S}_N$ são matrizes simétricas e positivas semidefinidas e $\mathbf{R}_k$ é uma matriz simétrica positiva definida para todo k. Por simplicidade notacional, os subscritos nas matrizes da planta e do índice de desempenho serão omitidos.

Para $k = N$ tem-se que

$$J_N^* = \frac{1}{2} \mathbf{x}_N^\top \mathbf{S}_N \mathbf{x}_N \tag{4.13}$$

Para $k = N - 1$ tem-se que

$$J_{N-1} = \frac{1}{2} \mathbf{x}_{N-1}^\top \mathbf{Q} \mathbf{x}_{N-1} + \frac{1}{2} \mathbf{u}_{N-1}^\top \mathbf{R} \mathbf{u}_{N-1} + \frac{1}{2} \mathbf{x}_N^\top \mathbf{S}_N \mathbf{x}_N. \tag{4.14}$$

De acordo com o princípio de otimalidade (4.9), é necessário achar $\mathbf{u}^*_{N-1}$ minimizando (4.14). Para conseguir isso, utiliza-se a equação (4.11) para escrever

$$J_{N-1} = \frac{1}{2}\mathbf{x}_{N-1}^\top \mathbf{Q}\mathbf{x}_{N-1} + \frac{1}{2}\mathbf{u}_{N-1}^\top \mathbf{R}\mathbf{u}_{N-1} + \frac{1}{2}(\mathbf{A}\mathbf{x}_{N-1} + \mathbf{B}\mathbf{u}_{N-1})^\top \mathbf{S}_N(\mathbf{A}\mathbf{x}_{N-1} + \mathbf{B}\mathbf{u}_{N-1}). \quad (4.15)$$

Derivando (4.15) com relação a $\mathbf{u}_{N-1}$, tem-se

$$\begin{aligned}
\frac{\partial J_{N-1}}{\partial \mathbf{u}_{N-1}} &= \mathbf{R}\mathbf{u}_{N-1} + \mathbf{S}_N(\mathbf{A}\mathbf{x}_{N-1} + \mathbf{B}\mathbf{u}_{N-1})\mathbf{B} && (4.16)\\
&= \mathbf{R}\mathbf{u}_{N-1} + (\mathbf{A}\mathbf{x}_{N-1} + \mathbf{B}\mathbf{u}_{N-1})^\top \mathbf{S}_N\mathbf{B} && (4.17)\\
&= \mathbf{R}\mathbf{u}_{N-1} + \mathbf{B}^\top \mathbf{S}_N(\mathbf{A}\mathbf{x}_{N-1} + \mathbf{B}\mathbf{u}_{N-1}) && (4.18)\\
&= \mathbf{R}\mathbf{u}_{N-1} + \mathbf{B}^\top \mathbf{S}_N\mathbf{A}\mathbf{x}_{N-1} + \mathbf{B}^\top \mathbf{S}_N\mathbf{B}\mathbf{u}_{N-1} && (4.19)\\
&= (\mathbf{R} + \mathbf{B}^\top \mathbf{S}_N\mathbf{B})\mathbf{u}_{N-1} + \mathbf{B}^\top \mathbf{S}_N\mathbf{A}\mathbf{x}_{N-1}. && (4.20)
\end{aligned}$$

Portanto,

$$\mathbf{u}^*_{N-1} = -(\mathbf{R} + \mathbf{B}^\top \mathbf{S}_N\mathbf{B})^{-1}\mathbf{B}^\top \mathbf{S}_N\mathbf{A}\mathbf{x}_{N-1}. \quad (4.21)$$

Definindo o ganho de Kalman como

$$\mathbf{K}_{N-1} = (\mathbf{R} + \mathbf{B}^\top \mathbf{S}_N\mathbf{B})^{-1}\mathbf{B}^\top \mathbf{S}_N\mathbf{A}, \quad (4.22)$$

tem-se que

$$\mathbf{u}^*_{N-1} = -\mathbf{K}_{N-1}\mathbf{x}_{N-1}. \quad (4.23)$$

Portanto, o custo ótimo em $k = N-1$ é achado substituindo (4.23) em (4.15):

$$\begin{aligned}
J^*_{N-1} &= \frac{1}{2}\mathbf{x}_{N-1}^\top \mathbf{Q}\mathbf{x}_{N-1} + \frac{1}{2}(\mathbf{K}_{N-1}\mathbf{x}_{N-1})^\top \mathbf{R}(\mathbf{K}_{N-1}\mathbf{x}_{N-1}) && (4.24)\\
&\quad + \frac{1}{2}(\mathbf{A}\mathbf{x}_{N-1} - \mathbf{B}\mathbf{K}_{N-1}\mathbf{x}_{N-1})^\top \mathbf{S}_N(\mathbf{A}\mathbf{x}_{N-1} - \mathbf{B}\mathbf{K}_{N-1}\mathbf{x}_{N-1}) && (4.25)\\
&= \frac{1}{2}\mathbf{x}_{N-1}^\top \mathbf{Q}\mathbf{x}_{N-1} + \frac{1}{2}\mathbf{x}_{N-1}^\top \mathbf{K}_{N-1}^\top \mathbf{R}\mathbf{K}_{N-1}\mathbf{x}_{N-1} && (4.26)\\
&\quad + \frac{1}{2}(\mathbf{x}_{N-1}^\top \mathbf{A}^\top - \mathbf{x}_{N-1}^\top \mathbf{K}_{N-1}^\top \mathbf{B}^\top)\mathbf{S}_N(\mathbf{A}\mathbf{x}_{N-1} - \mathbf{B}\mathbf{K}_{N-1}\mathbf{x}_{N-1}) && (4.27)\\
&= \frac{1}{2}\mathbf{x}_{N-1}^\top \Big[\mathbf{Q} + \mathbf{K}_{N-1}^\top \mathbf{R}\mathbf{K}_{N-1} + (\mathbf{A} - \mathbf{B}\mathbf{K}_{N-1})^\top \mathbf{S}_N(\mathbf{A} - \mathbf{B}\mathbf{K}_{N-1})\Big]\mathbf{x}_{N-1}. && (4.28)
\end{aligned}$$

Definindo

$$\mathbf{S}_{N-1} = \mathbf{Q} + \mathbf{K}_{N-1}^\top \mathbf{R}\mathbf{K}_{N-1} + (\mathbf{A} - \mathbf{B}\mathbf{K}_{N-1})^\top \mathbf{S}_N(\mathbf{A} - \mathbf{B}\mathbf{K}_{N-1}), \quad (4.29)$$

obtém-se

$$J^*_{N-1} = \frac{1}{2}\mathbf{x}_{N-1}^\top \mathbf{S}_{N-1}\mathbf{x}_{N-1}. \quad (4.30)$$

Para $k = N-2$ tem-se que

$$J_{N-2} = \frac{1}{2}\mathbf{x}_N^\top \mathbf{S}_N\mathbf{x}_N + \frac{1}{2}\Big[\mathbf{x}_{N-2}^\top \mathbf{Q}\mathbf{x}_{N-2} + \mathbf{u}_{N-2}^\top \mathbf{R}\mathbf{u}_{N-2} + \mathbf{x}_{N-1}^\top \mathbf{Q}\mathbf{x}_{N-1} + \mathbf{u}_{N-1}^\top \mathbf{R}\mathbf{u}_{N-1}\Big]. \quad (4.31)$$

Usando (4.15) tem-se que

$$J_{N-2} = \frac{1}{2}\left[\mathbf{x}_{N-2}^{\top}\mathbf{Q}\mathbf{x}_{N-2} + \mathbf{u}_{N-2}^{\top}\mathbf{R}\mathbf{u}_{N-2}\right] + J_{N-1}. \tag{4.32}$$

Aplicando o controle ótimo u_{N-1}^{*} em (4.23) e a equação em (4.30) à equação em (4.32) obtém-se

$$\begin{aligned} J_{N-2} &= \frac{1}{2}\mathbf{x}_{N-2}^{\top}\mathbf{Q}\mathbf{x}_{N-2} + \frac{1}{2}\mathbf{u}_{N-2}^{\top}\mathbf{R}\mathbf{u}_{N-2} + J_{N-1}^{*} &(4.33)\\ &= \frac{1}{2}\mathbf{x}_{N-2}^{\top}\mathbf{Q}\mathbf{x}_{N-2} + \frac{1}{2}\mathbf{u}_{N-2}^{\top}\mathbf{R}\mathbf{u}_{N-2} + \frac{1}{2}\mathbf{x}_{N-1}^{\top}\mathbf{S}_{N-1}\mathbf{x}_{N-1}. &(4.34)\end{aligned}$$

Portanto,

$$J_{N-2} = \frac{1}{2}\mathbf{x}_{N-2}^{\top}\mathbf{Q}\mathbf{x}_{N-2} + \frac{1}{2}\mathbf{u}_{N-2}^{\top}\mathbf{R}\mathbf{u}_{N-2} + \frac{1}{2}\mathbf{x}_{N-1}^{\top}\mathbf{S}_{N-1}\mathbf{x}_{N-1}. \tag{4.35}$$

De acordo com o princípio de otimalidade (4.9), é necessário achar $\mathbf{u}_{N-2}^{*}$ minimizando (4.35). Note que (4.35) tem a mesma forma que (4.14). Portanto, o controle ótimo e o custo em $k = N-2$ é dado por (4.22), (4.23) e (4.29), (4.30) substituindo N por $N-1$. Se continua-se diminuindo k e aplicando-se o princípio de otimalidade, tem-se para $k = N-1, \ldots, 1, 0$

$$\begin{aligned} \mathbf{K}_k &= (\mathbf{R} + \mathbf{B}^{\top}\mathbf{S}_{k+1}\mathbf{B})^{-1}\mathbf{B}^{\top}\mathbf{S}_{k+1}\mathbf{A}, &(4.36)\\ \mathbf{u}_k^{*} &= -\mathbf{K}_k\mathbf{x}_k, &(4.37)\\ \mathbf{S}_k &= \mathbf{Q} + \mathbf{K}_k^{\top}\mathbf{R}\mathbf{K}_k + (\mathbf{A} - \mathbf{B}\mathbf{K}_k)^{\top}\mathbf{S}_{k+1}(\mathbf{A} - \mathbf{B}\mathbf{K}_k), &(4.38)\\ J_k^{*} &= \frac{1}{2}\mathbf{x}_k^{\top}\mathbf{S}_k\mathbf{x}_k. &(4.39)\end{aligned}$$

Note que (4.38) coincide com a versão estabilizada de Joseph da equação de Riccati (3.64) e (4.36) coincide com o ganho de Kalman (3.56).

4.2 Controle ótimo LQR com horizonte infinito

Seja a planta linear invariante no tempo

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \tag{4.40}$$

Seja o índice de desempenho associado à planta (4.40)

$$J_0 = \frac{1}{2}\sum_{k=0}^{\infty} L(\mathbf{x}_k, \mathbf{u}_k) = \frac{1}{2}\sum_{k=0}^{\infty}[\mathbf{x}_k^{\top}\mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^{\top}\mathbf{R}\mathbf{u}_k] \tag{4.41}$$

definido ao longo do intervalo de tempo discreto $[0, +\infty)$, onde $\mathbf{Q}$ é simétrica e positiva semidefinida, $\mathbf{R}$ é simétrica e positiva definida.

Seja $\mathbf{h}$ uma política de controle, na qual $\mathbf{u}_k = \mathbf{h}(\mathbf{x}_k)$, para todo $k \geq 0$. Seja o valor $V^{\mathbf{h}}(\mathbf{x}_k)$ de seguir a política $\mathbf{h}$ no estado $\mathbf{x}_k$ definida por

$$V^{\mathbf{h}}(\mathbf{x}_k) = \frac{1}{2}\sum_{i=k}^{\infty} L(\mathbf{x}_i, \mathbf{u}_i). \tag{4.42}$$

Note que o valor em (4.42) pode ser reescrita como

$$V^{\mathbf{h}}(\mathbf{x}_k) = \frac{1}{2}\sum_{i=k}^{\infty} L(\mathbf{x}_i, \mathbf{u}_i) \tag{4.43}$$

$$= \frac{1}{2}L(\mathbf{x}_k, \mathbf{u}_k) + \frac{1}{2}\sum_{i=k+1}^{\infty} L(\mathbf{x}_i, \mathbf{u}_i) \tag{4.44}$$

$$= \frac{1}{2}L(\mathbf{x}_k, \mathbf{u}_k) + V^{\mathbf{h}}(\mathbf{x}_{k+1}). \tag{4.45}$$

Então, obtém-se a equação de Bellman

$$V^{\mathbf{h}}(\mathbf{x}_k) = \frac{1}{2}L(\mathbf{x}_k, \mathbf{u}_k) + V^{\mathbf{h}}(\mathbf{x}_{k+1}). \tag{4.46}$$

Usando o princípio de otimalidade de Bellman, tem-se que

$$V^*(\mathbf{x}_k) = \min_{\mathbf{u}_k}\left(\frac{1}{2}L(\mathbf{x}_k, \mathbf{u}_k) + V^*(\mathbf{x}_{k+1})\right) \tag{4.47}$$

e a política de controle ótima é

$$\mathbf{u}_k^* = \arg\min_{\mathbf{u}_k}\left(\frac{1}{2}L(\mathbf{x}_k, \mathbf{u}_k) + V^*(\mathbf{x}_{k+1})\right). \tag{4.48}$$

Note que (4.46) e (4.47) coincidem com (2.31) e (2.32), respectivamente. A equação em (4.47) é chamada equação de Hamilton-Jacobi-Bellman (HJB).

A seguir será mostrado que a equação HJB (4.47) é equivalente à DARE (3.85) e a equação em (4.48) é equivalente ao ganho de Kalman em estado estacionário (3.103).

O custo de seguir a política $\mathbf{u}_k = -\mathbf{K}\mathbf{x}_k$ no estado $\mathbf{x}_k$ é

$$V^{\mathbf{K}}(\mathbf{x}_k) = \frac{1}{2}\mathbf{x}_k^\top \mathbf{S}\mathbf{x}_k, \tag{4.49}$$

onde $\mathbf{S} = \mathbf{S}^\top > 0$. Logo, substituindo (4.49) em (4.46) obtém-se

$$\mathbf{x}_k^\top \mathbf{S}\mathbf{x}_k = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k + \mathbf{x}_{k+1}^\top \mathbf{S}\mathbf{x}_{k+1}. \tag{4.50}$$

Em termos do ganho de realimentação $\mathbf{u}_k = -\mathbf{K}\mathbf{x}_k$ e (4.40), a equação em (4.50) pode ser reescrita como

$$\mathbf{x}_k^T \mathbf{S}\mathbf{x}_k = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k + \mathbf{x}_{k+1}^\top \mathbf{S}\mathbf{x}_{k+1} \tag{4.51}$$

$$= \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{x}_k^\top \mathbf{K}^\top \mathbf{R}\mathbf{K}\mathbf{x}_k + \mathbf{x}_k^\top (\mathbf{A} - \mathbf{B}\mathbf{K})^\top \mathbf{S}(\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}_k \tag{4.52}$$

$$= \mathbf{x}_k^\top \left(\mathbf{Q} + \mathbf{K}^\top \mathbf{R}\mathbf{K} + (\mathbf{A} - \mathbf{B}\mathbf{K})^\top \mathbf{S}(\mathbf{A} - \mathbf{B}\mathbf{K})\right)\mathbf{x}_k, \tag{4.53}$$

ou seja

$$\mathbf{x}_k^\top \mathbf{S}\mathbf{x}_k = \mathbf{x}_k^\top \left(\mathbf{Q} + \mathbf{K}^\top \mathbf{R}\mathbf{K} + (\mathbf{A} - \mathbf{B}\mathbf{K})^\top \mathbf{S}(\mathbf{A} - \mathbf{B}\mathbf{K})\right)\mathbf{x}_k. \tag{4.54}$$

Como a equação (4.54) tem-se que satisfazer para todo $\mathbf{x}_k$, então obtém-se a equação de Lyapunov

$$(\mathbf{A}-\mathbf{BK})^\top \mathbf{S}(\mathbf{A}-\mathbf{BK}) - \mathbf{S} + \mathbf{Q} + \mathbf{K}^\top \mathbf{R}\mathbf{K} = 0. \tag{4.55}$$

Por outro lado, usando (4.40), a equação em (4.50) pode ser reescrita como

$$\mathbf{x}_k^\top \mathbf{S}\mathbf{x}_k = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k + (\mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k)^\top \mathbf{S}(\mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k). \tag{4.56}$$

Então, para o valor ótimo

$$V^*(\mathbf{x}_k) = \frac{1}{2}\mathbf{x}_k^\top \mathbf{S}_\infty \mathbf{x}_k, \tag{4.57}$$

a equação (4.56) fica

$$\mathbf{x}_k^\top \mathbf{S}_\infty \mathbf{x}_k = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k + (\mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k)^\top \mathbf{S}_\infty(\mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k), \tag{4.58}$$

onde $\mathbf{u}_k = -\mathbf{K}_\infty \mathbf{x}_k$ é a lei de controle ótimo. Logo, derivando (4.58) com relação a $\mathbf{u}_k$, tem-se

$$\begin{aligned} 0 &= \mathbf{R}\mathbf{u}_k + \mathbf{B}^\top \mathbf{S}_\infty(\mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k) && (4.59)\\ &= \mathbf{R}\mathbf{u}_k + \mathbf{B}^\top \mathbf{S}_\infty \mathbf{A}\mathbf{x}_k + \mathbf{B}^\top \mathbf{S}_\infty \mathbf{B}\mathbf{u}_k && (4.60)\\ &= (\mathbf{R} + \mathbf{B}^\top \mathbf{S}_\infty \mathbf{B})\mathbf{u}_k + \mathbf{B}^\top \mathbf{S}_\infty \mathbf{A}\mathbf{x}_k. && (4.61) \end{aligned}$$

Então, a política ótima é

$$\mathbf{u}_k = -(\mathbf{R} + \mathbf{B}^\top \mathbf{S}_\infty \mathbf{B})^{-1}\mathbf{B}^\top \mathbf{S}_\infty \mathbf{A}\mathbf{x}_k, \tag{4.62}$$

onde a matriz de ganho ótima é dada por

$$\mathbf{K}_\infty = (\mathbf{R} + \mathbf{B}^\top \mathbf{S}_\infty \mathbf{B})^{-1}\mathbf{B}^\top \mathbf{S}_\infty \mathbf{A}. \tag{4.63}$$

Substituindo (4.63) na equação de Lyapunov (4.55) obtém-se a equação algébrica de Riccati

$$\mathbf{A}^\top \mathbf{S}_\infty \mathbf{A} - \mathbf{S}_\infty + \mathbf{Q} - \mathbf{A}^\top \mathbf{S}_\infty \mathbf{B}(\mathbf{R} + \mathbf{B}^T \mathbf{S}_\infty \mathbf{B})^{-1}\mathbf{B}^\top \mathbf{S}_\infty \mathbf{A} = 0. \tag{4.64}$$

Observação 4.2.1 *Note que a equação de Bellman em (4.50) é equivalente à equação de Lyapunov em (4.55). No entanto, a equação em (4.50) não depende das matrizes do modelo do sistema e pode ser resolvida em tempo real.*

4.2.1 A maldição da dimensionalidade

Os métodos apresentados no Capítulo 2 requerem que as funções-Q e políticas sejam representadas como uma tabela indexada pelos estados e ações discretas. Se a

quantidade de elementos dos conjunto de estados e ações é "pequeno", então a programação dinâmica é uma boa ferramenta para resolver o problema da estratégia de decisão ótima. No entanto, uma grande desvantagem dos métodos de programação dinâmica é que os cálculos podem ser proibitivamente custosos quando o conjunto de estados e o conjunto de ações têm um grande número de elementos (maldição da dimensionalidade de Bellman).

Já que a maioria dos problemas do mundo real enquadram nesta categoria, métodos de aproximação para programação dinâmica foram explorados desde o seu início (ver por exemplo (LUUS, 2019)) até a atualidade com o uso de métodos de programação dinâmica adaptativa.

A programação dinâmica adaptativa supera a "maldição da dimensionalidade" usando uma estrutura paramétrica para aproximar a função valor mediante métodos ator-crítico de aprendizagem por reforço. Os métodos ator-crítico têm duas estruturas paramétricas, uma estrutura paramétrica para avaliar a política de controle ótima (chamada de ator) e uma estrutura paramétrica para a função valor (chamada de crítico).

No próximo capítulo, será apresentado o Q-learning baseado em uma estrutura paramétrica para a função valor Q^h.

4.3 Considerações finais

Neste capítulo, apresentou-se a teoria de controle ótimo LQR desde a abordagem da programação dinâmica de Bellman. Similarmente ao capítulo anterior, a abordagem da programação dinâmica para o resolver o problema do LQR é *offline* e requer as matrizes do modelo do sistema. Além disso, uma das principais desvantagens da programação dinâmica é a "maldição da dimensionalidade", a qual ocorre quando o conjunto de estados e o conjunto de ações têm um grande número de elementos. Como consequência, surge uma classe de métodos conhecida como programação dinâmica adaptativa/aproximada.

5 Controle Ótimo LQR via Q-learning

Os métodos desenvolvidos no Capítulo 2 são para processos de decisão de Markov finitos. Como foi comentado na Observação 3 do Capítulo 2, as função Q e as políticas podem ser representadas como uma tabela indexada pelos estados e ações discretas. No entanto, os conjuntos de estados e ações de controle no problema do LQR são contínuos. Nesses casos, as funções Q e as políticas são aproximadas usando um aproximador paramétrico.

Por outro lado, os métodos desenvolvidos no Capítulo 3 são métodos *offline* que envolvem soluções de equações de Riccati e é necessário o conhecimento das matrizes do modelo do sistema. O Algoritmo 8, no Capítulo 3, resolve o problema de controle ótimo *offline*, sem resolver explicitamente a DARE, mas ele precisa do conhecimento das matrizes do modelo do sistema.

Uma abordagem livre das matrizes do modelo do sietma é o Q-learning que é forma de aprendizagem por reforço desenvolvido para processos de decisão de Markov discretos e finito (WATKINS, 1989; WATKINS; DAYAN, 1992). Para o caso de processos de decisão de Markov contínuos e infinitos, o Q-learning foi desenvolvido em Bradtke, Ydstie e Barto (1994a), Rummery e Niranjan (1994). O Q-learning resolve o problema de controle ótimo online sem o conhecimento de modelo, somente usando dados obtidos ao longo da trajetória do sistema.

Na atualidade, continua-se usando a aprendizagem por reforço para resolver o problema do LQR onde a dinâmica do sistema linear é desconhecida (LOPEZ; ALSALTI; MÜLLER, 2021; YAGHMAIE; GUSTAFSSON; LJUNG, 2022; FARJADNASAB; BABAZADEH, 2022).

Neste capítulo será desenvolvido o algoritmo iteração de política para o problema de controle LQR online e livre de modelo baseado no erro de diferença temporal.

5.1 Diferença temporal

Seja a planta linear invariante no

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \tag{5.1}$$

Seja o índice de desempenho associado à planta (5.1)

$$J(\mathbf{x}_0) = \frac{1}{2}\sum_{k=0}^{\infty} L(\mathbf{x}_k, \mathbf{u}_k) = \frac{1}{2}\sum_{k=0}^{\infty}[\mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k], \tag{5.2}$$

onde $\mathbf{Q}$ é uma matrize simétrica positiva semidefinida e $\mathbf{R}$ é uma matriz simétrica positiva definida.

Seja $\mathbf{h}$ uma política de controle e seja o valor $V^{\mathbf{h}}(\mathbf{x}_k)$ de seguir a política $\mathbf{h}$ no estado $\mathbf{x}_k$ definida como em (4.42)

$$V^{\mathbf{h}}(\mathbf{x}_k) = \frac{1}{2}\sum_{i=k}^{\infty} L(\mathbf{x}_i, \mathbf{u}_i). \tag{5.3}$$

Baseada na equação de Bellman (4.46), define-se um erro residual variante no tempo

$$e_k = \frac{1}{2}L(\mathbf{x}_k, \mathbf{u}_k) + V^{\mathbf{h}}(\mathbf{x}_{k+1}) - V^{\mathbf{h}}(\mathbf{x}_k). \tag{5.4}$$

O erro e_k em (5.4) é o erro de diferença temporal (TD-*temporal difference*). Se a equação de Bellman (4.46) valer, então o erro de TD é igual a zero. Portanto, para uma política de controle $\mathbf{u}_k = \mathbf{h}(\mathbf{x}_k)$ fixa pode-se resolver a equação $e_k = 0$ em cada instante de tempo k para o valor $V^{\mathbf{h}}(\cdot)$ que é a solução de mínimos quadrados da equação de TD

$$0 = \frac{1}{2}L(\mathbf{x}_k, \mathbf{u}_k) + V^{\mathbf{h}}(\mathbf{x}_{k+1}) - V^{\mathbf{h}}(\mathbf{x}_k). \tag{5.5}$$

Isso produz a melhor aproximação ao valor (5.3).

Resolver a equação de TD equivale a resolver uma equação de Bellman online, sem o conhecimento da dinâmica do sistema, somente usando dados obtidos ao longo da trajetória do sistema.

5.1.1 Diferença temporal no problema do LQR

A seguir será mostrado como resolver a equação de TD em (5.5) em tempo real, somente usando dados ao longo da trajetória do sistema para uma política de controle admissível $\mathbf{h}(\mathbf{x}_k) = -\mathbf{K}\mathbf{x}_k$.

O valor de seguir a política $\mathbf{h}(\mathbf{x}_k) = -\mathbf{K}\mathbf{x}_k$ é dado em (4.49) por

$$V^{\mathbf{h}}(\mathbf{x}_k) = \frac{1}{2}\mathbf{x}_k^{\top}\mathbf{S}\mathbf{x}_k, \tag{5.6}$$

sendo $\mathbf{S} = \mathbf{S}^{\top} > 0$. Logo, substituindo (5.6) em (5.5) obtém-se

$$\mathbf{x}_k^{\top}\mathbf{S}\mathbf{x}_k - \mathbf{x}_{k+1}^{\top}\mathbf{S}\mathbf{x}_{k+1} = \mathbf{x}_k^{\top}\mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^{\top}\mathbf{R}\mathbf{u}_k. \tag{5.7}$$

Para resolver (5.7) pode-se aproximar a função valor $V^{\mathbf{h}}(\cdot)$ usando um aproximador paramétrico.

Denotando por $\mathbf{S}^i$ a coluna i da matriz $\mathbf{S} = \mathbf{S}^{\top} \in \mathbb{R}^{n\times n}$, para $i = 1, 2, \ldots, n$ e

$$\mathbf{x}_k = [x_k^1 \; x_k^2 \; \cdots \; x_k^n]^{\top} \in \mathbb{R}^n. \tag{5.8}$$

Então,

$$\begin{aligned}
\mathbf{x}_k^\top \mathbf{S}\mathbf{x}_k &= \mathbf{x}_k^\top \left(x_k^1 \mathbf{S}^1 + x_k^2 \mathbf{S}^2 + \cdots + x_k^n \mathbf{S}^n \right) && (5.9)\\
&= x_k^1 \mathbf{x}_k^\top \mathbf{S}^1 + x_k^2 \mathbf{x}_k^\top \mathbf{S}^2 + \cdots + x_k^n \mathbf{x}_k^\top \mathbf{S}^n && (5.10)\\
&= (\mathbf{S}^1)^\top x_k^1 \mathbf{x}_k + (\mathbf{S}^2)^\top x_k^2 \mathbf{x}_k + \cdots + (\mathbf{S}^n)^\top x_k^n \mathbf{x}_k && (5.11)\\
&= \bar{\mathbf{S}}^\top (\mathbf{x}_k \otimes \mathbf{x}_k) && (5.12)\\
&= \bar{\mathbf{S}}^\top \bar{\mathbf{x}}_k, && (5.13)
\end{aligned}$$

onde $\bar{\mathbf{S}}^\top = [(\mathbf{S}^1)^\top \cdots (\mathbf{S}^n)^\top]$ e $\bar{\mathbf{x}}_k = \mathbf{x}_k \otimes \mathbf{x}_k$. Portanto, a equação de TD (5.7) fica

$$(\bar{\mathbf{x}}_k - \bar{\mathbf{x}}_{k+1})^\top \bar{\mathbf{S}} = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k. \tag{5.14}$$

Como $\mathbf{S}$ é simétrica então tem só $n(n+1)/2$ elementos independentes. Portanto, os termos redundantes em $\bar{\mathbf{x}}_k$ são removidos para definir uma base quadrática $\bar{\mathbf{x}}_k$ com $n(n+1)/2$ elementos independentes.

Explicitamente, para um vetor $\mathbf{x} = [x^1 \; x^2 \cdots x^n]^\top \in \mathbb{R}^n$, define-se

$$\bar{\mathbf{x}} = [(x^1)^2 \cdots x^1 x^n \; (x^2)^2 \cdots x^2 x^n \cdots (x^{n-1})^2 \; x^{n-1} x^n \; (x^n)^2]^\top \in \mathbb{R}^{n(n+1)/2}. \tag{5.15}$$

Para um vetor $\theta = [\theta^1 \; \theta^2 \cdots \; \theta^{n(n+1)/2}]^\top \in \mathbb{R}^{n(n+1)/2}$, define-se a matriz simétrica

$$\Theta = \begin{bmatrix} \theta^1 & \theta^2/2 & \cdots & \theta^n/2 \\ \theta^2/2 & \theta^{n+1} & \cdots & \theta^{2n-1}/2 \\ \vdots & \vdots & & \vdots \\ \theta^n/2 & \theta^{2n-1}/2 & \cdots & \theta^{n(n+1)/2} \end{bmatrix} \in \mathbb{R}^{n \times n}. \tag{5.16}$$

A matriz Θ é chamada reconstrução de θ, enquanto que θ é chamado vetorização de Θ.

Seja uma lei de controle $\mathbf{u}_k = -\mathbf{K}\mathbf{x}_k$ admissível. No tempo $k+1$ é medido o estado anterior $\mathbf{x}_k$, a ação de controle $\mathbf{u}_k$, o próximo estado $\mathbf{x}_{k+1}$, e assim é calculado o vetor de regressores $\varphi_k = \bar{\mathbf{x}}_k - \bar{\mathbf{x}}_{k+1}$ e a observação $r_k = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k$.

Então, é resolvida em tempo real a equação

$$\varphi_k^\top \theta = r_k \tag{5.17}$$

usando o algoritmo RLS ou NLMS. Logo, a reconstrução da solução em (5.17), denotado por Θ, é solução da equação de Bellman

$$\mathbf{x}_k^\top \Theta \mathbf{x}_k = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k + \mathbf{x}_{k+1}^\top \Theta \mathbf{x}_{k+1}, \tag{5.18}$$

ou seja, Θ é solução da equação de Lyapunov

$$(\mathbf{A} - \mathbf{B}\mathbf{K})^\top \Theta (\mathbf{A} - \mathbf{B}\mathbf{K}) - \Theta + \mathbf{Q} + \mathbf{K}^\top \mathbf{R}\mathbf{K} = 0. \tag{5.19}$$

O algoritmo RLS converge para os verdadeiros parâmetros θ if φ_k em (5.17) satisfaz a condição de persistência de excitação (GOODWIN; SIN, 2009)

$$\infty > \epsilon_1 \mathbf{I} \geq \frac{1}{N} \sum_{j=1}^{N} \varphi_{t-j} \varphi_{t-j}^{\top} \geq \epsilon_0 \mathbf{I} > 0, \tag{5.20}$$

para todo $t \geq N_0$ e $N \geq N_0$, na qual $\epsilon_0 \leq \epsilon_1$, N_0 é um número inteiro positivo e $\mathbf{I}$ é a matriz identidade de dimensão apropriada.

Enquanto que para o algoritmo NLMS, a condição de persistência de excitação é a seguinte (BITMEAD, 1984): existe um inteiro T, para todo k, tal que

$$\infty > \alpha_1 \mathbf{I} \geq \sum_{t=k}^{k+T} \varphi_t \varphi_t^{\top} \geq \alpha_2 \mathbf{I} > 0, \tag{5.21}$$

na qual α_1, α_2 são parâmetros positivos.

Note que persistência de excitação em (5.21) implica que, para todo k, a matriz

$$\Phi^{\top}(k,T)\Phi(k,T) = \sum_{t=k}^{k+T} \varphi_t \varphi_t^{\top} \tag{5.22}$$

é simétrica e positiva definida, sendo

$$\Phi(k,T) = \begin{bmatrix} \varphi_k^{\top} \\ \varphi_{k+1}^{\top} \\ \vdots \\ \varphi_{k+T}^{\top} \end{bmatrix}. \tag{5.23}$$

Portanto, para todo k, a equação em (5.17), ou seja,

$$\Phi(k,T)\theta = \begin{bmatrix} r_k \\ r_{k+1} \\ \vdots \\ r_{k+1} \end{bmatrix} \tag{5.24}$$

tem uma única solução de mínimos quadrados, que é a vetorização da solução da equação de Bellman (5.18).

Obviamente, as condições (5.20) e (5.21) estão intimamente relacionadas, embora (5.20) seja mais exigente. A condição em (5.20) implica que, para todo k (fixo) e todo T, a matriz

$$P_k^{-1}(0) + \Phi^{\top}(k,T)\Phi(k,T), \tag{5.25}$$

é não singular, sendo $P_k(0)$ uma matriz simétrica positiva definida.

O Algoritmo 9 é uma versão online do Algoritmo 8. O Algoritmo 9 é o método padrão para resolver o problema de controle ótimo LQR usando iteração de política baseado no aprendizagem da função valor $V^{\mathbf{h}}$.

Algoritmo 9 Iteração de política online

1: **Input:** Política de controle admissível $\mathbf{u}_k = -\mathbf{K}(0)\mathbf{x}_k$
2: **for** $j = 0, 1, \ldots$ **do**
3: Resolver $(\bar{\mathbf{x}}_k - \bar{\mathbf{x}}_{k+1})^\top \theta(j) = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k$
4: $\Theta(j)$ = reconstrução de $\theta(j)$
5: $\mathbf{K}(j+1) = (\mathbf{B}^\top \Theta(j)\mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^\top \Theta(j)\mathbf{A}$
6: $\mathbf{u}_k = -\mathbf{K}(j+1)\mathbf{x}_k$
7: **end for**

5.1.2 LQR como um problema de filtragem

Baseado no método de Hewer (Algoritmo 8) e no Algoritmo 9, nesta seção mostra-se como resolver o problema do LQR com horizonte infinito visto como um problema de filtragem adaptativa.

O objetivo é encontrar uma aproximação, em tempo real, da solução da equação de Lyapunov

$$\mathbf{S}(j) = (\mathbf{A} - \mathbf{B}\mathbf{K}(j))^\top \mathbf{S}(j)(\mathbf{A} - \mathbf{B}\mathbf{K}(j)) + \mathbf{Q} + \mathbf{K}^\top(j)\mathbf{R}\mathbf{K}(j) \tag{5.26}$$

no Algoritmo 8. Para conseguir isso, o conjunto dos números inteiros não negativos $\mathbb{Z}^+ = \{0, 1, 2, 3, \ldots\}$ é dividido em subconjuntos $[t_j, t_{j+1})$, disjuntos dois a dois, com $j = 0, 1, 2, \ldots$, tal que uma aproximação da solução $\mathbf{S}_j$ da equação de Lyapunov é obtida usando esquemas de filtragem adaptativa, para todo instante do tempo k no conjunto $[t_j, t_{j+1})$, de maneira que $\mathbf{S}_j \to \mathbf{S}_\infty$ quando $j \to \infty$ (YÁNEZ; SOUZA, 2021).

Seja J um número inteiro positivo fixo. Para cada $j = 0, 1, 2 \ldots$, seja

$$t_j = jJ. \tag{5.27}$$

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

$$[t_j, t_{j+1}) = \{t_j, t_j + 1, t_j + 2, \ldots, t_{j+1} - 1\}. \tag{5.28}$$

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

$$[t_j, t_{j+1}) \cap [t_i, t_{i+1}) = \varnothing, \tag{5.29}$$

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

$$\bigcup_{j=0}^{\infty} [t_j, t_{j+1}) = \mathbb{Z}. \tag{5.30}$$

A seguir, será desenvolvida uma lei de controle

$$\mathbf{u}_k = -\mathbf{K}(j)\mathbf{x}_k, \tag{5.31}$$

em que a matriz $\mathbf{K}(j)$ é obtida em tempo real, para $k \in [t_j, t_{j+1})$, tal que $\mathbf{K}(j) \to \mathbf{K}_\infty$.

Seja a lei de controle

$$\mathbf{u}_k = -\mathbf{K}(j)\mathbf{x}_k, \tag{5.32}$$

sendo

$$\mathbf{K}(j) = (\mathbf{R} + \mathbf{B}^\top \mathbf{S}(j-1)\mathbf{B})^{-1}\mathbf{B}^\top \mathbf{S}(j-1)\mathbf{A}, \tag{5.33}$$

com $j \geq 1$, e $\mathbf{S}_j$ é a solução da equação de Lyapunov (5.26), para $j \geq 0$ e $\mathbf{K}(0)$ admissível. Para a matriz (5.33), tem-se que a equação de Lyapunov (5.26) é equivalente à equação de Bellman

$$\mathbf{x}_k^\top \mathbf{S}(j)\mathbf{x}_k = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k + \mathbf{x}_{k+1}^\top \mathbf{S}(j)\mathbf{x}_{k+1}. \tag{5.34}$$

Logo, usando a notação em (5.15), a equação de Bellman (5.34) torna-se

$$(\bar{\mathbf{x}}_k - \bar{\mathbf{x}}_{k+1})^\top \theta(j) = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k. \tag{5.35}$$

Seja o filtro formado por (Figura 15):

$$\varphi_k = \bar{\mathbf{x}}_k - \bar{\mathbf{x}}_{k+1} \quad \text{(sinal de entrada)} \tag{5.36a}$$

$$y_k = \varphi_k^\top \theta_k \quad \text{(sinal de saída)} \tag{5.36b}$$

$$d_k = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k \quad \text{(sinal desejada)} \tag{5.36c}$$

$$e_k = d_k - y_k \quad \text{(sinal de erro)} \tag{5.36d}$$

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

Algoritmo 10 Algoritmo *online* para o LQR usando filtragem

Input: Política de controle admissível $\mathbf{u}_k = -\mathbf{K}(0)\mathbf{x}_k$
for $j = 0, 1, \ldots, N$ **do**
  for $k = t_j, t_j + 1, \ldots, t_{j+1} - 1$ **do**
    $\mathbf{u}_k = -\mathbf{K}(j+1)\mathbf{x}_k$
    $\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k$
    $d_k = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k$
    $\theta_{k+1} = \theta_k + f(\theta_k, e_k, \varphi_k)$
  end for
  $\mathbf{S}(j)$ = reconstrução de $\theta(j)$
  $\mathbf{K}(j+1) = (\mathbf{B}^\top \mathbf{S}(j)\mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^\top \mathbf{S}(j)\mathbf{A}$
end for

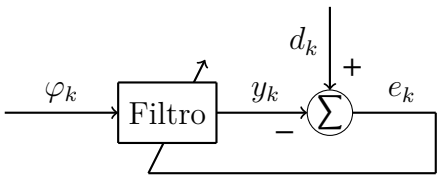


Figura 15 – Diagrama do filtro adaptativo para a identificação de $\theta(j)$ em (5.35).

No Algoritmo 10 é apresentado o algoritmo *online* para o LQR usando filtragem. Nesse caso, a função $f(\cdot, \cdot, \cdot)$ representa o algoritmo de adaptação de parâmetros, que em nosso caso pode ser o algoritmo RLS ou NLMS.

Como $k \in [t_j, t_{j+1})$, então (5.35) forma um conjunto de J equações com $n(n+1)/2$ incógnitas, que se pode determinar a solução por mínimos quadrados. Portanto, para obter uma única solução de mínimos quadrados, é necessário e suficiente que a matriz

$$\Phi^\top(t_j, t_{j+1}-1)\Phi(t_j, t_{j+1}-1) = \sum_{t=t_j}^{t_{j+1}-1} \varphi_t \varphi_t^\top \tag{5.37}$$

seja não singular. Uma condição necessária para que a matriz em (5.37) seja não singular é

$$J \geq n(n+1)/2. \tag{5.38}$$

A condição de que a matriz em (5.37) seja não singular é chamada uma condição de excitação (ÅSTRÖM; WITTENMARK, 2008, p. 44).

Para garantir a condição de excitação, é usada a técnica conhecida como revitalização de estados (MURRAY et al., 2002; AL-TAMIMI; ABU-KHALAF; LEWIS, 2007), na qual os estados são reinicializados para o estado inicial $\mathbf{x}_0$. Para resolver recursivamente (5.35) usando os algoritmos RLS ou NLMS, a condição de excitação é substituída pela condição de persistência de excitação (5.20)-(5.21).

5.2 Q-learning

A função valor $V^{\mathbf{h}}(\cdot)$ em (5.6) depende somente do estado. Além disso, mesmo que o Algoritmo 9 é online, não é livre de modelo já que é necessário do conhecimento das matrizes do modelo no aprendizagem do controlador.

Seja $\mathbf{h}$ uma política de controle e seja o valor $Q^{\mathbf{h}}(\mathbf{x}_k, \mathbf{u}_k)$ de seguir a política $\mathbf{h}$ no estado $\mathbf{x}_k$ e na ação de controle $\mathbf{u}_k$ definida como em (2.16)

$$Q^{\mathbf{h}}(\mathbf{x}_k, \mathbf{u}_k) = \frac{1}{2}L(\mathbf{x}_k, \mathbf{u}_k) + V^{\mathbf{h}}(\mathbf{x}_{k+1}). \tag{5.39}$$

Note que função valor $Q^{\mathbf{h}}(\cdot,\cdot)$ depende do estado e da ação de controle enquanto que $V^{\mathbf{h}}(\cdot)$ depende somente do estado.

A função Q ótima é definida como

$$Q^*(\mathbf{x}_k, \mathbf{u}_k) = \frac{1}{2}L(\mathbf{x}_k, \mathbf{u}_k) + V^*(\mathbf{x}_{k+1}). \tag{5.40}$$

Logo, de (4.47) segue-se que

$$V^*(\mathbf{x}_k) = \min_{\mathbf{u}_k}(Q^*(\mathbf{x}_k, \mathbf{u}_k)) \tag{5.41}$$

e o controle ótimo como

$$\mathbf{h}^*(\mathbf{x}_k) = \arg\min_{\mathbf{u}_k}(Q^*(\mathbf{x}_k, \mathbf{u}_k)). \tag{5.42}$$

Na ausência de restrições de controle, obtém-se o valor mínimo resolvendo

$$\frac{\partial}{\partial \mathbf{u}_k} Q^*(\mathbf{x}_k, \mathbf{u}_k) = 0. \tag{5.43}$$

5.2.1 Equação de Bellman para a função Q

A aprendizagem da função valor $Q^{\mathbf{h}}(\cdot,\cdot)$ é baseado em equações de Bellman. Note que

$$Q^{\mathbf{h}}(\mathbf{x}_k, \mathbf{h}(\mathbf{x}_k)) = V^{\mathbf{h}}(\mathbf{x}_k). \tag{5.44}$$

Então, de (5.39), segue-se que uma equação de Bellman para a função $Q^{\mathbf{h}}(\cdot,\cdot)$ é dada por

$$Q^{\mathbf{h}}(\mathbf{x}_k, \mathbf{u}_k) = \frac{1}{2}L(\mathbf{x}_k, \mathbf{u}_k) + Q^{\mathbf{h}}(\mathbf{x}_{k+1}, \mathbf{h}(\mathbf{x}_{k+1})) \tag{5.45}$$

ou

$$Q^{\mathbf{h}}(\mathbf{x}_k, \mathbf{h}(\mathbf{x}_k)) = \frac{1}{2}L(\mathbf{x}_k, \mathbf{h}(\mathbf{x}_k)) + Q^{\mathbf{h}}(\mathbf{x}_{k+1}, \mathbf{h}(\mathbf{x}_{k+1})). \tag{5.46}$$

A função Q ótima satisfaz

$$Q^{*}(\mathbf{x}_k, \mathbf{u}_k) = \frac{1}{2}L(\mathbf{x}_k, \mathbf{u}_k) + Q^{*}(\mathbf{x}_{k+1}, \mathbf{h}^{*}(\mathbf{x}_{k+1})). \tag{5.47}$$

5.2.2 Parametrização da função Q para o LQR

É possível aprender a função $Q^{\mathbf{h}}(\cdot,\cdot)$ em (5.39) usando um aproximador paramétrico. No seguinte, vai-se mostrar um aproximador paramétrio da função $Q^{\mathbf{h}}(\cdot,\cdot)$ para o caso do LQR.

Considere a política de controle

$$\mathbf{u}_k = -\mathbf{K}\mathbf{x}_k. \tag{5.48}$$

Então, de (5.39) e (4.49), obtém-se

$$\begin{aligned} Q^{\mathbf{K}}(\mathbf{x}_k, \mathbf{u}_k) &= \frac{1}{2}\mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k + \frac{1}{2}\mathbf{x}_{k+1}^\top \mathbf{S}\mathbf{x}_{k+1} \\ &= \frac{1}{2}\left(\mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k + (\mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k)^\top \mathbf{S}(\mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k)\right) \end{aligned} \tag{5.49}$$

ou, equivalentemente,

$$Q^{\mathbf{K}}(\mathbf{x}_k, \mathbf{u}_k) = \frac{1}{2}\begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}^\top \begin{bmatrix} \mathbf{Q} + \mathbf{A}^\top \mathbf{S}\mathbf{A} & \mathbf{A}^\top \mathbf{S}\mathbf{B} \\ \mathbf{B}^\top \mathbf{S}\mathbf{A} & \mathbf{R} + \mathbf{B}^\top \mathbf{S}\mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} \equiv \frac{1}{2}\mathbf{z}_k^\top \Theta \mathbf{z}_k. \tag{5.50}$$

Usando a notação em (5.15) obtém-se que

$$Q^{\mathbf{K}}(\mathbf{x}_k, \mathbf{u}_k) = \frac{1}{2}\theta^\top \bar{\mathbf{z}}_k, \tag{5.51}$$

onde θ é uma vetorização da matriz

$$\Theta = \begin{bmatrix} \Theta_{xx} & \Theta_{xu} \\ \Theta_{ux} & \Theta_{uu} \end{bmatrix} = \begin{bmatrix} \mathbf{Q} + \mathbf{A}^\top \mathbf{S}\mathbf{A} & \mathbf{A}^\top \mathbf{S}\mathbf{B} \\ \mathbf{B}^\top \mathbf{S}\mathbf{A} & \mathbf{R} + \mathbf{B}^\top \mathbf{S}\mathbf{B} \end{bmatrix}. \tag{5.52}$$

5.2.3 Iteração de política para a função Q

Substituindo (5.51) na equação de TD (5.5), obtém-se

$$(\bar{\mathbf{z}}_k - \bar{\mathbf{z}}_{k+1})^\top \theta = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k. \tag{5.53}$$

Então, a equação em (5.53) é resolvida online até obter a solução θ. Logo, a política de atualização é dada por

$$\mathbf{u}_k = -\Theta_{uu}^{-1}\Theta_{ux}\mathbf{x}_k. \tag{5.54}$$

O método de iteração de política é baseado em dois passos. Dado uma política de controle admissível $\mathbf{h}(\mathbf{x}_k) = -\mathbf{K}\mathbf{x}_k$, o primeiro passo, chamado avaliação de política, que consiste em resolver a equação em (5.53) e assim obter a função valor $Q^{\mathbf{K}}(\cdot,\cdot)$. O segundo passo é obter uma nova política a partir da função valor obtida usando (5.54).

O Algoritmo 11 mostra o método de iteração de política online e livre de modelo para o problema do LQR.

A diferença entre o Algoritmo 9 e o Algoritmo 11 é que o Algoritmo 9 aprende a função valor de estado $V^{\mathbf{h}}(\cdot)$ e precisa das matrizes do modelo do sistema na aprendizagem do controlador, enquanto que o Algoritmo 11 é baseado na aprendizagem da função valor $Q^{\mathbf{K}}(\cdot,\cdot)$ e não precisa das matrizes do modelo do sistema na aprendizagem do controlador.

Algoritmo 11 Iteração de política Q-learning

1: **Input:** Política de controle admissível $\mathbf{u}_k = -\mathbf{K}(0)\mathbf{x}_k$
2: **for** $j = 0, 1, \ldots$ **do**
3: Resolver $(\bar{\mathbf{z}}_k - \bar{\mathbf{z}}_{k+1})^\top \theta(j) = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k$
4: $\Theta(j)$ = reconstrução de $\theta(j)$
5: $\mathbf{K}(j+1) = (\Theta(j)_{uu})^{-1}\Theta(j)_{ux}$
6: $\mathbf{u}_k = -\mathbf{K}(j+1)\mathbf{x}_k$
7: **end for**

5.2.4 Necessidade do ruido de prova na aprendizagem da função Q

Para obter convergência ao parâmetro verdadeiro nos algoritmos RLS e NLMS o vetor de regressores tem que satisfazer a condição de persistência de excitação (5.20)-(5.21). No entanto, o seguinte resultado mostra que o vetor de regressores $\varphi_k = \bar{\mathbf{z}}_k - \bar{\mathbf{z}}_{k+1}$ em (5.53) não satisfaz a condição de persistência de excitação.

Teorema 5.2.1 *Seja a trajetória $\mathbf{x}_k$ em $\mathbb{R}^n$ gerada pelo sistema (5.1), sendo o sinal de controle dado por $\mathbf{u}_k = \mathbf{K}\mathbf{x}_k$, com $\mathbf{K} \in \mathbb{R}^{m\times n}$. Seja o sinal vetorial $\mathbf{z}_k$ em $\mathbb{R}^{n+m}$ definida por*

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{K}\mathbf{x}_k \end{bmatrix}, \text{ para } k \geq 0, \tag{5.55}$$

Se $l \geq 1$ é fixo, então a matriz

$$\mathbf{Z}_{kl} = \begin{bmatrix} (\mathbf{z}_k - \mathbf{z}_{k+1})^\top \\ (\mathbf{z}_{k+1} - \mathbf{z}_{k+2})^\top \\ \vdots \\ (\mathbf{z}_{k+l-1} - \mathbf{z}_{k+l})^\top \end{bmatrix} \in \mathbb{R}^{l\times(n+m)} \tag{5.56}$$

tem m colunas linearmente dependentes.

Demostração. Tem-se que

$$\begin{aligned} [\mathbf{z}_k - \mathbf{z}_{k+1}]^\top &= [\mathbf{x}_k^\top \ \mathbf{u}_k^\top] - [\mathbf{x}_{k+1}^\top \ \mathbf{u}_{k+1}^\top] && (5.57)\\ &= [(\mathbf{x}_k - \mathbf{x}_{k+1})^\top \ (\mathbf{K}(\mathbf{x}_k - \mathbf{x}_{k+1}))^\top] && (5.58)\\ &= [(\mathbf{x}_k - \mathbf{x}_{k+1})^\top \ (\mathbf{x}_k - \mathbf{x}_{k+1})^\top \mathbf{K}^\top] && (5.59)\\ &= [(\mathbf{x}_k - \mathbf{x}_{k+1})^\top \ \sum_{j=1}^{n} (\mathbf{K}^j)^\top (\mathbf{x}_k - \mathbf{x}_{k+1})^j], && (5.60) \end{aligned}$$

onde $\mathbf{K}^j$ é a j-ésima coluna de $\mathbf{K}$ e $(\mathbf{x}_k - \mathbf{x}_{k+1})^j$ é a j-ésima componente do vetor $\mathbf{x}_k - \mathbf{x}_{k+1}$ com $1 \leq j \leq n$. Então, a matriz $\mathbf{Z}_{kl}$ é igual a

$$\mathbf{Z}_{kl} = \begin{bmatrix} (\mathbf{x}_k - \mathbf{x}_{k+1})^\top & \sum_{j=1}^{n} (\mathbf{K}^j)^\top (\mathbf{x}_k - \mathbf{x}_{k+1})^j] \\ (\mathbf{x}_{k+1} - \mathbf{x}_{k+2})^\top & \sum_{j=1}^{n} (\mathbf{K}^j)^\top (\mathbf{x}_{k+1} - \mathbf{x}_{k+2})^j] \\ \vdots & \vdots \\ (\mathbf{x}_{k+l-1} - \mathbf{x}_{k+l})^\top & \sum_{j=1}^{n} (\mathbf{K}^j)^\top (\mathbf{x}_{k+l-1} - \mathbf{x}_{k+l})^j] \end{bmatrix}. \tag{5.61}$$

Então, a i-ésima coluna, com $1 \leq i \leq m$, de

$$\begin{bmatrix} \sum_{j=1}^{n} (\mathbf{K}^j)^\top (\mathbf{x}_k - \mathbf{x}_{k+1})^j \\ \sum_{j=1}^{n} (\mathbf{K}^j)^\top (\mathbf{x}_{k+1} - \mathbf{x}_{k+2})^j] \\ \vdots \\ \sum_{j=1}^{n} (\mathbf{K}^j)^\top (\mathbf{x}_{k+l-1} - \mathbf{x}_{k+l})^j] \end{bmatrix} \tag{5.62}$$

é uma combinação linear das colunas formadas por

$$\begin{bmatrix} (\mathbf{x}_k - \mathbf{x}_{k+1})^\top \\ (\mathbf{x}_{k+1} - \mathbf{x}_{k+2})^\top \\ \vdots \\ (\mathbf{x}_{k+l-1} - \mathbf{x}_{k+l})^\top \end{bmatrix}. \tag{5.63}$$

Como consequência do Teorema 5.2.1, tem-se que o vetor de regressores em (5.53) não satisfaz a condição de persistência de excitação.

Para obter a condição de persistência de excitação, em (BRADTKE; YDSTIE; BARTO, 1994a) é adicionado um ruído de prova $\mathbf{v}_k$ na ação de controle, de maneira que

$$\mathbf{u}_k = -\mathbf{K}\mathbf{x}_k + \mathbf{v}_k. \tag{5.64}$$

Ruídos de prova $\mathbf{v}_k$ como em (5.64) tem sido usado em esquemas de aprendizagem por reforço para o problema de controle (BRADTKE; YDSTIE; BARTO, 1994a; HAGEN; KRÖSE, 1998; AL-TAMIMI; LEWIS; ABU-KHALAF, 2007; KIM; LEWIS, 2010; CHAKRABARTY et al., 2019)

5.2.5 Simulação computacional

Seja a planta linear invariante no tempo (HAGEN; KRÖSE, 1998)

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \tag{5.65}$$

onde

$$\mathbf{A} = \begin{bmatrix} -0,6 & -0,4 \\ 1,0 & 0,0 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 0,0 \\ 1,0 \end{bmatrix}. \tag{5.66}$$

Além $\mathbf{Q} = \mathbf{I}_2^\top \mathbf{I}_2$ e $\mathbf{R} = \mathbf{I}_1$, sendo $\mathbf{I}_2$ e $\mathbf{I}_1$ as identidades de ordem 2 e 1, respectivamente. O estado inicial é $\mathbf{x}_0 = [1 \ 1]^\top$. Seja a matriz de ganho inicial admissível

$$\mathbf{K}(0) = [0,2 \ \ -0,2]. \tag{5.67}$$

É fácil notar que $(\mathbf{A}, \mathbf{B})$ é controlável e $(\mathbf{A}, \mathbf{I}_2)$ é observável.

Para $\mathbf{S}_N = \mathbf{I}_2$ e $N = 30$, iterando (3.73) para trás no tempo, obtém-se que a solução da DARE é

$$\mathbf{S}^* = \begin{bmatrix} 2,0791 & 0,4108 \\ 0,4108 & 1,3210 \end{bmatrix}. \tag{5.68}$$

Portanto, o ganho de controle ótimo é dado por

$$\mathbf{K}^* = (\mathbf{B}^\top \mathbf{S}^* \mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^\top \mathbf{S}^* \mathbf{A} = [0,4630 \ \ -0,0708]. \tag{5.69}$$

Exemplo 1. Primeiro é usado o Algoritmo 10 para encontrar as matrizes (5.68) e (5.69), onde 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 (5.38). 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 $\mathbf{K}(j)$ no Algoritmo 10. Para os cenários (i) e (ii), simulações serão realizadas usando o algoritmo NLMS e o algoritmo padrão RLS na etapa 7 do Algoritmo 10.

A Figura 16 mostra a importância da revitalização de estados no Algoritmo 10. Note que quando os estados não são revitalizados, o algoritmo RLS não consegue adaptar os parâmetros de maneira certa, mas o algoritmo NLMS consegue convergência para os verdadeiros parâmetros (Figura 16 (a)-(b)). No entanto, se os estados não são revitalizados, não sempre podemos garantir convergência do algoritmo NLMS. Ou seja, se mudamos os dados J e N, é possível que o algoritmo NLMS não consiga adaptar os parâmetros de maneira certa.

Se os estados são revitalizados, ambos algoritmos RLS e NLMS conseguem adaptar os parâmetros, e nesse caso o algoritmo RLS tem um melhor desempenho do que o algoritmo NLMS (Figura 16 (c)-(d)).

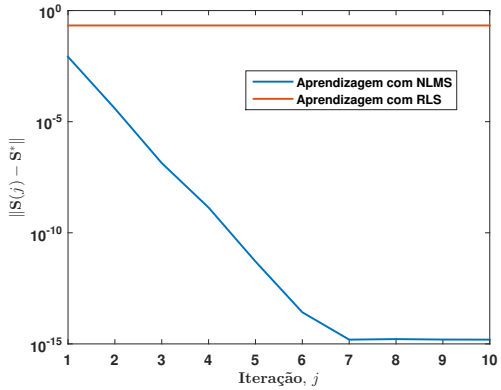
Exemplo 2. Será usado o Algoritmo 11 para encontrar as matrizes (5.69) e

$$\Theta^* = \begin{bmatrix} \mathbf{Q} + \mathbf{A}^T\mathbf{S}^*\mathbf{A} & \mathbf{A}^T\mathbf{S}^*\mathbf{B} \\ \mathbf{B}^T\mathbf{S}^*\mathbf{A} & \mathbf{R} + \mathbf{B}^T\mathbf{S}^*\mathbf{B} \end{bmatrix} = \begin{bmatrix} 2,5766 & 0,3347 & 1,0746 \\ 0,3347 & 1,3327 & -0,1643 \\ 1,0746 & -0,1643 & 2,3210 \end{bmatrix}. \tag{5.70}$$

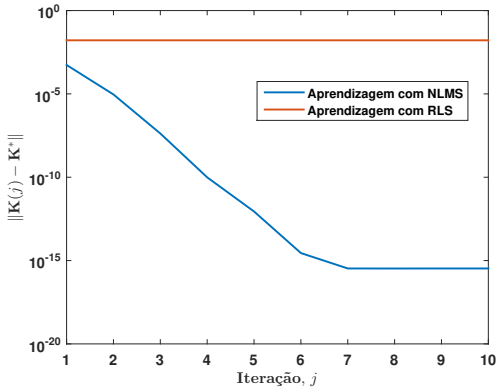
Nesse caso, é necessário adicionar um ruído de prova na ação de controle. Como ruído de prova será usado um processo aleatório com distribuição $N(\mathbf{0}, \sigma^2\mathbf{I})$, sendo a variância $\sigma^2 = 1$. As Figuras 17 (a)-(b) mostram a convergência dos parâmetros $\Theta(j)$ e da matriz de ganho $\mathbf{K}(j)$, respectivamente, no Algoritmo 11 quando é adicionado o ruído de prova v_k na ação de controle u_k em cada instante do tempo k. As Figuras 17 (c)-(d) mostram o comportamento dos parâmetros $\Theta(j)$ e da matriz de ganho $\mathbf{K}(j)$, respectivamente, no Algoritmo 11 quando não é adicionado um ruído de prova na ação de controle. Nesse casso, quando não é adicionado um ruído de prova na ação de controle, os algoritmos de adaptação RLS e NLMS não conseguem adaptar os parâmetros.

5.3 Considerações finais

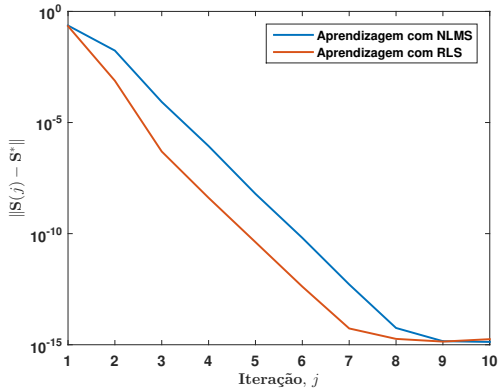
Neste capítulo, a abordagem de filtragem adaptativa para o problema de controle LQR foi apresentada, considerando-se dois casos, um que utiliza o modelo do sistema (Algoritmo 10) e outro livre do modelo (Algoritmo 11). Para o caso da abordagem livre de modelo, foi demonstrado a necessidade de inserir um ruído de prova (artificial) na ação de controle para garantir convergência dos algoritmo adaptativos. Enquanto que



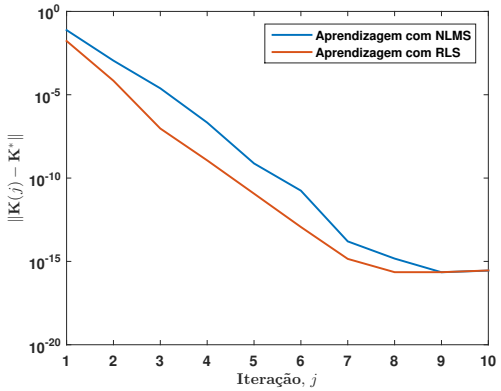
(a) Convergência de $\|\mathbf{S}(j) - \mathbf{S}^*\|$



(b) Convergência de $\|\mathbf{K}(j) - \mathbf{K}^*\|$



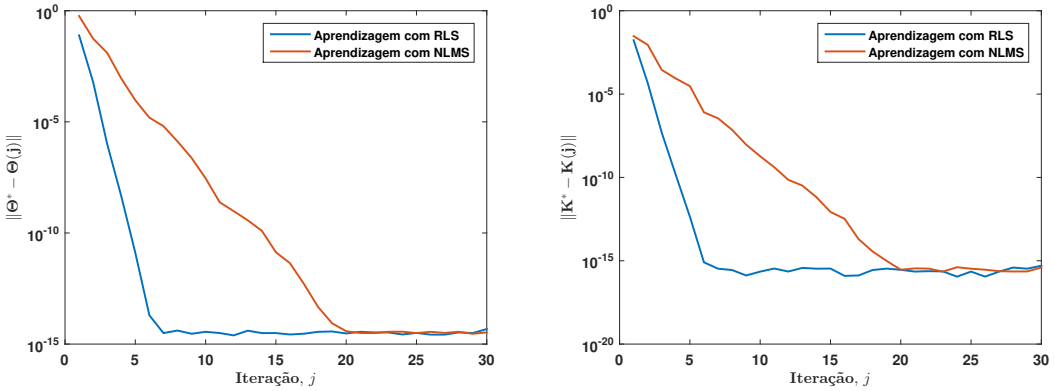
(c) Convergência de $\|\mathbf{S}(j) - \mathbf{S}^*\|$



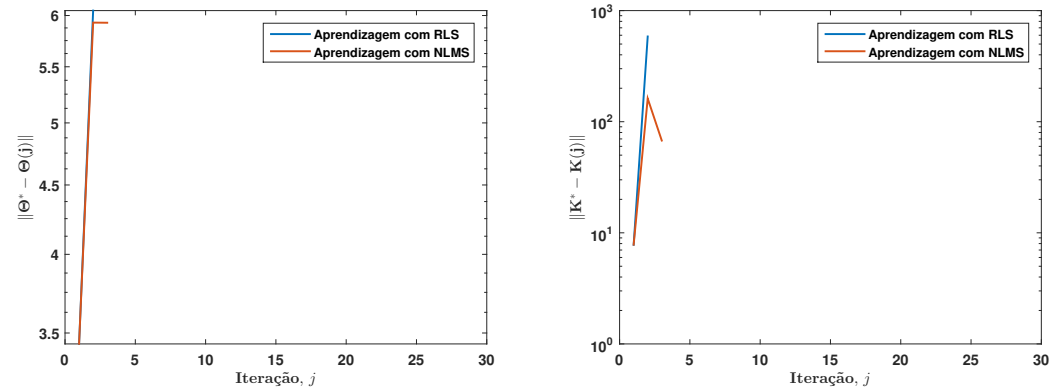
(d) Convergência de $\|\mathbf{K}(j) - \mathbf{K}^*\|$

Figura 16 – Convergência de $\mathbf{S}(j)$ e $\mathbf{K}(j)$ no Algoritmo 10 com $\mathbf{A}$ e $\mathbf{B}$ em (5.66) e $\mathbf{K}(0)$ em (5.67). (a)-(b) mostra o comportamento dos parâmetros $\mathbf{S}(j)$ e $\mathbf{K}(j)$ sem revitalização de estados. Nesse caso, a adaptação RLS não conseguem adaptar os parâmetros. (c)-(d) mostra o comportamento dos parâmetros $\mathbf{S}(j)$ e $\mathbf{K}(j)$ com revitalização de estados. Nesse caso, os algoritmos de adaptação RLS e NLMS conseguem adaptar os parâmetros.

para garantir convergência dos algoritmo adaptativos na abordagem baseada em modelo, a técnica de revitalização de estados foi empregada.



(a) Convergência de $\|\Theta^* - \Theta(j)\|$ (b) Convergência de $\|\mathbf{K}^* - \mathbf{K}(j)\|$



(c) Convergência de $\|\Theta^* - \Theta(j)\|$ (d) Convergência de $\|\mathbf{K}^* - \mathbf{K}(j)\|$

Figura 17 – Convergência de $\Theta(j)$ e $\mathbf{K}(j)$ no Algoritmo 11 com $\mathbf{A}$ e $\mathbf{B}$ em (5.66) e $\mathbf{K}(0)$ em (5.67). (a)-(b) mostra o comportamento dos parâmetros $\Theta(j)$ e $\mathbf{K}(j)$ quando é adicionado um ruído de prova na ação de controle. Nesse caso, obtém-se convergência aos parâmetros verdadeiros. (c)-(d) mostra o comportamento dos parâmetros $\Theta(j)$ e $\mathbf{K}(j)$ quando não é adicionado um ruído de prova na ação de controle. Nesse caso, os algoritmos de adaptação RLS e NLMS não conseguem adaptar os parâmetros.

6 Efeitos devido ao ruído de prova

Para obter a condição de persistência de excitação, o ruído de prova, adicionado à entrada do sistema, durante o processo de aprendizagem do controlador, pode ser considerando como uma perturbação adicional, afetando os estados reais do sistema.

Neste capítulo, é estudado o efeito do ruído de prova nos estados do sistema em termos da média e da matriz de autocovariância do vetor de estados.

Considere o sistema linear

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \tag{6.1a}$$
$$\mathbf{u}_k = -\mathbf{K}\mathbf{x}_k, \tag{6.1b}$$

com $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{K} \in \mathbb{R}^{m\times n}$ e estado inicial $\mathbf{x}_0$. Agora, considere o sistema afetado pelo ruído de prova

$$\tilde{\mathbf{x}}_{k+1} = \mathbf{A}\tilde{\mathbf{x}}_k + \mathbf{B}\tilde{\mathbf{u}}_k \tag{6.2a}$$
$$\tilde{\mathbf{u}}_k = -\mathbf{K}\tilde{\mathbf{x}}_k + \mathbf{v}_k, \tag{6.2b}$$

com estado inicial $\tilde{\mathbf{x}}_0 = \mathbf{x}_0$, sendo $\mathbf{v}_k = \left[v_k^1\ v_k^2\ \cdots\ v_k^m\right]^\top$ um ruído de prova vetorial tal que:

i) Para cada $i = 1, 2, \ldots, m$, tem-se que $\{v_k^i\}_{k=0}^{\infty}$ é um sinal aleatório independente e identicamente distribuído, com média zero e variância σ^2.

ii) Para cada $k = 0, 1, 2, \ldots$, tem-se que v_k^i e v_k^j são não correlacionados, para todo $i, j = 1, 2, \ldots, m$, com $i \neq j$, ou seja

$$\Gamma_{\mathbf{v}^i\mathbf{v}^j}(k) = E\left\{\left(v_k^i - E[v_k^i]\right)\left(v_k^j - E[v_k^j]\right)\right\} = \begin{cases} \sigma^2, & i = j \\ 0, & \text{outro caso,} \end{cases} \tag{6.3}$$

Note que as duas condições anteriores implicam que, para todo k, a esperança matemática $E[\mathbf{v}_k]$ e a matriz de autocovariância do vetor aleatório $\mathbf{v}_k$ são iguais a:

$$E[\mathbf{v}_k] = \mathbf{0} \in \mathbb{R}^m \tag{6.4a}$$
$$\Gamma_{\mathbf{v}}(k) = E\left\{[\mathbf{v}_k - E[\mathbf{v}_k]][\mathbf{v}_k - E[\mathbf{v}_k]]^\top\right\} = \sigma^2\mathbf{I} \in \mathbb{R}^{m\times m}, \tag{6.4b}$$

sendo $\mathbf{I}$ a matriz identidade em $\mathbb{R}^{m\times m}$.

O sistema dado em (6.1) produz um processo de decisão de Markov determinístico, enquanto que (6.2) produz um processo de decisão de Markov estocástico.

Para mostrar o efeito do ruído de prova nos estados e ações de controle, usa-se a seguinte fórmula que pode ser encontrada em Hagen e Kröse (1998).

Proposição 6.0.1 *A solução do sistema em (6.2) é*

$$\tilde{\mathbf{x}}_k = (\mathbf{A} - \mathbf{BK})^k \mathbf{x}_0 + \sum_{j=1}^{k} (\mathbf{A} - \mathbf{BK})^{k-j} \mathbf{B}\mathbf{v}_{j-1}, \tag{6.5}$$

para todo $k \geq 0$, sendo $\mathbf{v}_k$ um ruído de prova vetorial.

Demostração. Será usado indução. Para $k = 0$, é trivial que (6.5) é verdadeira. Para o caso $k = 1$, tem-se

$$\begin{aligned} \tilde{\mathbf{x}}_1 &= \mathbf{A}\tilde{\mathbf{x}}_0 + \mathbf{B}\tilde{\mathbf{u}}_0 && (6.6)\\ &= \mathbf{A}\tilde{\mathbf{x}}_0 + \mathbf{B}(-\mathbf{K}\tilde{\mathbf{x}}_0 + \mathbf{v}_0) && (6.7)\\ &= (\mathbf{A} - \mathbf{BK})\mathbf{x}_0 + \mathbf{B}\mathbf{v}_0. && (6.8) \end{aligned}$$

Então, de (6.8) segue-se que (6.5) é verdadeira para $k = 1$.

Suponha que (6.5) é verdadeira para $k = n$. De (6.2a) e (6.2b) segue-se que

$$\tilde{\mathbf{x}}_{n+1} = (\mathbf{A} - \mathbf{BK})\tilde{\mathbf{x}}_n + \mathbf{B}\mathbf{v}_n \tag{6.9}$$

Como (6.5) é verdadeira para $k = n$, então de (6.9), segue-se que

$$\begin{aligned} \tilde{\mathbf{x}}_{n+1} &= (\mathbf{A} - \mathbf{BK})\left[(\mathbf{A} - \mathbf{BK})^n \mathbf{x}_0 + \sum_{j=1}^{n} (\mathbf{A} - \mathbf{BK})^{n-j} \mathbf{B}\mathbf{v}_{j-1}\right] + \mathbf{B}\mathbf{v}_n && (6.10)\\ &= (\mathbf{A} - \mathbf{BK})^{n+1} \mathbf{x}_0 + \sum_{j=1}^{n} (\mathbf{A} - \mathbf{BK})^{n+1-j} \mathbf{B}\mathbf{v}_{j-1} + \mathbf{B}\mathbf{v}_n && (6.11)\\ &= (\mathbf{A} - \mathbf{BK})^{n+1} \mathbf{x}_0 + \sum_{j=1}^{n+1} (\mathbf{A} - \mathbf{BK})^{n+1-j} \mathbf{B}\mathbf{v}_{j-1}. && (6.12) \end{aligned}$$

Portanto, de (6.12) segue-se que (6.5) é verdadeira para $k = n + 1$.

É óbvio que a solução de (6.1), que nesse caso é (CHEN, 1999, p. 131)

$$\mathbf{x}_k = (\mathbf{A} - \mathbf{BK})^k \mathbf{x}_0, \tag{6.13}$$

que é diferente da solução de (6.2) devido ao somatório em (6.5).

Os seguintes dois Teoremas mostram uma fórmula fechada para as matrizes de autocovariância

$$\begin{aligned} \Gamma_{\tilde{\mathbf{x}}}(k) &= E\left\{[\tilde{\mathbf{x}}_k - E[\tilde{\mathbf{x}}_k]][\tilde{\mathbf{x}}_k - E[\tilde{\mathbf{x}}_k]]^\top\right\} && (6.14)\\ \Gamma_{\tilde{\mathbf{u}}}(k) &= E\left\{[\tilde{\mathbf{u}}_k - E[\tilde{\mathbf{u}}_k]][\tilde{\mathbf{u}}_k - E[\tilde{\mathbf{u}}_k]]^\top\right\} && (6.15) \end{aligned}$$

do estado $\tilde{\mathbf{x}}_k$ e da ação $\tilde{\mathbf{u}}_k$, respectivamente, e que dita sequência de matrizes $\Gamma_{\tilde{\mathbf{x}}}(k), \Gamma_{\tilde{\mathbf{u}}}(k)$ convergem quando a matriz de ganho de controle é admissível.

Teorema 6.0.1 *(YÁNEZ; SOUZA, 2022) Considere o sistema em (6.2), onde o ruído de prova* $\mathbf{v}_k$ *satisfaz as condições em (6.4). Então, as seguintes condições são satisfeitas:*

(a) Para todo $k \geq 0$, $E[\tilde{\mathbf{x}}_k] = \mathbf{x}_k$.

(b) Para todo k, $\Gamma_{\tilde{\mathbf{x}}}(k) = \sigma^2 \mathbf{S}_k$, *sendo*

$$\mathbf{S}_k = \sum_{j=1}^{k} \left((\mathbf{A} - \mathbf{BK})^{k-j} \mathbf{B}\mathbf{B}^\top ((\mathbf{A} - \mathbf{BK})^{k-j})^\top \right). \tag{6.16}$$

(c) Para todo k,

$$\tilde{\mathbf{u}}_k = \mathbf{u}_k + \sum_{j=1}^{k} \mathbf{K}(\mathbf{A} - \mathbf{BK})^{k-j} \mathbf{B}\mathbf{v}_{j-1} + \mathbf{v}_k, \tag{6.17}$$

sendo $\mathbf{u}_k = \mathbf{K}\mathbf{x}_k$.

(d) Para todo k, $\Gamma_{\tilde{\mathbf{u}}}(k)$ *é igual a* $\sigma^2 \mathbf{U}_k$, *no qual*

$$\mathbf{U}_k = \mathbf{K} \left(\sum_{j=1}^{k} (\mathbf{A} - \mathbf{BK})^{k-j} \mathbf{B}\mathbf{B}^\top ((\mathbf{A} - \mathbf{BK})^{k-j})^\top \right) \mathbf{K}^\top + \mathbf{I}. \tag{6.18}$$

Demostração. (a) Segue-se imediato de (6.5) que

$$\begin{aligned} E[\tilde{\mathbf{x}}_k] &= E[(\mathbf{A} - \mathbf{BK})^k \mathbf{x}_0] + E\left[\sum_{j=1}^{k} (\mathbf{A} - \mathbf{BK})^{k-j} \mathbf{B}\mathbf{v}_{j-1} \right] && (6.19) \\ &= (\mathbf{A} - \mathbf{BK})^k \mathbf{x}_0 + \sum_{j=1}^{k} (\mathbf{A} - \mathbf{BK})^{k-j} \mathbf{B} E[\mathbf{v}_{j-1}] && (6.20) \\ &= (\mathbf{A} - \mathbf{BK})^k \mathbf{x}_0 + \sum_{j=1}^{k} (\mathbf{A} - \mathbf{BK})^{k-j} \mathbf{B}\mathbf{0} && (6.21) \\ &= \mathbf{x}_k. && (6.22) \end{aligned}$$

(b) Para cada $k \geq 1$, seja o vetor

$$\mathbf{s}(j) = (\mathbf{A} - \mathbf{BK})^{k-j} \mathbf{B}\mathbf{v}_{j-1} \in \mathbb{R}^n, \tag{6.23}$$

com $j = 1, 2, \ldots, k$. Como os vetores $\mathbf{v}_{j-1}$ são mutuamente não correlacionados, então de

(6.5) segue-se que

$$\begin{aligned}
\Gamma_{\tilde{\mathbf{x}}}(k) &= \Gamma_{\sum_{j=1}^{k}\mathbf{s}(j)}(k) && (6.24)\\
&= \sum_{j=1}^{k}\Gamma_{\mathbf{s}(j)}(k) && (6.25)\\
&= \sum_{j=1}^{k}(\mathbf{A}-\mathbf{BK})^{k-j}\mathbf{B}\Gamma_{\mathbf{v}}(j-1)((\mathbf{A}-\mathbf{BK})^{k-j}\mathbf{B})^{\top} && (6.26)\\
&= \sum_{j=1}^{k}(\mathbf{A}-\mathbf{BK})^{k-j}\mathbf{B}(\sigma^2\mathbf{I})((\mathbf{A}-\mathbf{BK})^{k-j}\mathbf{B})^{\top} && (6.27)\\
&= \sigma^2\sum_{j=1}^{k}\left((\mathbf{A}-\mathbf{BK})^{k-j}\mathbf{B}\mathbf{B}^{\top}((\mathbf{A}-\mathbf{BK})^{k-j})^{\top}\right) && (6.28)\\
&= \sigma^2\mathbf{S}_k. && (6.29)
\end{aligned}$$

(c) De (6.2b) e (6.5) segue-se que

$$\begin{aligned}
\tilde{\mathbf{u}}_k &= \mathbf{K}\left(\mathbf{x}_k+\sum_{j=1}^{k}(\mathbf{A}-\mathbf{BK})^{k-j}\mathbf{B}\mathbf{v}_{j-1}\right)+\mathbf{v}_k\\
&\quad \mathbf{u}_k+\sum_{j=1}^{k}\mathbf{K}(\mathbf{A}-\mathbf{BK})^{k-j}\mathbf{B}\mathbf{v}_{j-1}+\mathbf{v}_k. && (6.30)
\end{aligned}$$

(d) Usando a notação em (6.23), (6.17) implica que

$$\begin{aligned}
\Gamma_{\tilde{\mathbf{u}}}(k) &= \Gamma_{\sum_{j=1}^{k}\mathbf{K}\mathbf{s}(j)}(k)+\Gamma_{\mathbf{v}}(k)\\
&= \sum_{j=1}^{k}\Gamma_{\mathbf{K}s(j)}(k)+\sigma^2\mathbf{I}\\
&= \sum_{j=1}^{k}\mathbf{K}(\mathbf{A}-\mathbf{BK})^{k-j}\mathbf{B}\Gamma_{\mathbf{v}}(j-1)(\mathbf{K}(\mathbf{A}-\mathbf{BK})^{k-j}\mathbf{B})^{\top}+\sigma^2\mathbf{I}\\
&= \sum_{j=1}^{k}\mathbf{K}(\mathbf{A}-\mathbf{BK})^{k-j}\mathbf{B}\sigma^2\mathbf{I}\mathbf{B}^{\top}((\mathbf{A}-\mathbf{BK})^{k-j})^{\top}\mathbf{K}^{\top}+\sigma^2\mathbf{I}\\
&= \sigma^2\mathbf{U}_k. && (6.31)
\end{aligned}$$

Teorema 6.0.2 *(YÁNEZ; SOUZA, 2022) Considere o sistema em (6.2) com estado inicial* $\mathbf{x}_0$*, onde o ruído de prova* $\mathbf{v}_k$ *satisfaz as condições em (6.4).*

(a) Então, para cada matriz admissível $\mathbf{K}\in\mathbb{R}^{m\times n}$

$$\lim_{k\to\infty}\Gamma_{\tilde{\mathbf{x}}}(k)=\lim_{k\to\infty}E[\tilde{\mathbf{x}}_k\tilde{\mathbf{x}}_k^{\top}]=\sigma^2\mathbf{S}, \tag{6.32}$$

sendo $\mathbf{S}$ *a única solução da equação de Lyapunov*

$$(\mathbf{A}-\mathbf{BK})\mathbf{S}(\mathbf{A}-\mathbf{BK})^{\top}-\mathbf{S}+\mathbf{B}\mathbf{B}^{\top}=\mathbf{0}. \tag{6.33}$$

(b) A matriz de autocovariância $\Gamma_{\tilde{\mathbf{u}}}(k)$ *satisfaz*

$$\lim_{k\to\infty} \Gamma_{\tilde{\mathbf{u}}}(k) = \sigma^2(\mathbf{KSK}^\top + \mathbf{I}), \tag{6.34}$$

sendo $\mathbf{S}$ *a única solução da equação de Lyapunov (6.33).*

Demostração. (a) Como $\mathbf{K}$ é admissível, então a equação de Lyapunov (6.33) tem uma única solução, a série convergente (CHEN, 1999, p. 136)

$$\mathbf{S} = \sum_{l=0}^{\infty} (\mathbf{A}-\mathbf{BK})^l \mathbf{B}\mathbf{B}^\top ((\mathbf{A}-\mathbf{BK})^l)^\top. \tag{6.35}$$

Logo, do Teorema 6.0.1 (b)

$$\lim_{k\to\infty} \Gamma_{\tilde{\mathbf{x}}}(k) = \sigma^2 \left(\lim_{k\to\infty} \mathbf{S}_k \right) = \sigma^2 \mathbf{S}. \tag{6.36}$$

De (6.14) e do Teorema 6.0.1 (a) e (b) obtém-se

$$\Gamma_{\tilde{\mathbf{x}}}(k) = E[\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k^\top] - \mathbf{x}_k \mathbf{x}_k^\top = \sigma^2 \mathbf{S}_k. \tag{6.37}$$

Como $\mathbf{K}$ é admissível então $\mathbf{x}_k \to \mathbf{0}$ quando $k \to \infty$, e como $\mathbf{S}_k \to \mathbf{S}$ quando $k \to \infty$, então de (6.37) pode-se concluir (6.32).
(b) De (6.16) e (6.18) obtemos

$$\Gamma_{\tilde{\mathbf{u}}}(k) = \sigma^2(\mathbf{KS}_k\mathbf{K}^\top + \mathbf{I}). \tag{6.38}$$

Então o resultado segue imediatamente de (6.38) e (6.35).

Observação 6.0.1 *O Teorema 6.0.1 (b) e (d) e o Teorema 6.0.2 afirmam que a norma das matrizes de autocovariância dos estados e entradas de controle são proporcionais à variância* σ^2 *do ruído de prova.*

6.1 Simulação computacional

Exemplo 1. Seja a planta escalar

$$\tilde{x}_{k+1} = a\tilde{x}_k + b\tilde{u}_k, \tag{6.39}$$

com $a = 0,5$, $b = 2$, estado inicial $\tilde{x}_0 = 10$ e lei de controle

$$\tilde{u}_k = 0,1\tilde{x}_k + v_k, \tag{6.40}$$

onde e v_k é um processo aleatório normal de média 0 e variância σ^2.

Pelo Teorema 6.0.1 (b), tem-se que a variância dos estados em (6.39) é

$$\Gamma_{\tilde{x}}(k) = 4\sigma^2 \sum_{j=1}^{k} (0,7)^{2(k-j)}. \tag{6.41}$$

O limite em (6.41) quando $k \to \infty$ é a série geométrica convergente

$$\Gamma_{\tilde{x}}(\infty) = 4\sigma^2 \sum_{l=0}^{\infty} (0,7)^{2l} = \frac{4\sigma^2}{1-(0,7)^2}, \tag{6.42}$$

que coincide com o Teorema 6.0.2 (a), já que

$$s = \frac{4}{1-(0,7)^2} \tag{6.43}$$

é solução da equação de Lyapunov

$$s(0,7)^2 - s + 4 = 0. \tag{6.44}$$

A Tabela 9 mostra $\Gamma_{\tilde{x}}(\infty)$ em (6.42) para três diferentes variâncias σ^2. A Figura 18 mostra a convergência da variância $\Gamma_{\tilde{x}}(k)$ para diferentes ruídos de prova.

Tabela 9 – $\Gamma_{\tilde{x}}(\infty)$ em (6.42).

	$\sigma^2 = (0,1)^2$	$\sigma^2 = (0,5)^2$	$\sigma^2 = 1$
$\Gamma_{\tilde{x}}(\infty)$	0,0784	1,9608	7,8431

A Figura 19 mostra várias realizações do sistemas (6.39) para diferentes variâncias σ^2. Na Figura 19 (b), uma realização com variância $\sigma^2 = (0,1)^2$ foi feita. Note que nesse caso, o comportamento dos estados é consistente com o resultado teórico dado na Tabela 9, ou seja, a variância dos estados em regime permanente é igual a $\Gamma_{\tilde{x}}(\infty) = 0,0784$. As Figuras 19 (c)-(d) mostram uma realização com variâncias $\sigma^2 = (0,5)^5$ e $\sigma^2 = 1$, respectivamente. Nesses casos, os comportamento dos estados também coincidem com o resultado teórico dado na Tabela 9.

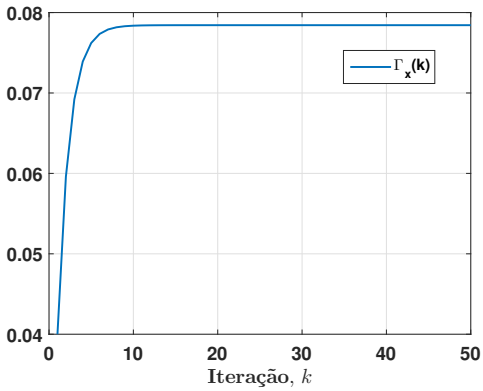
A Figura 20 mostra simulações de Monte Carlo de 1000 realizações de (6.39) para diferentes variâncias σ^2. Note que as Figuras 20 (b)-(d) são consistentes com o resultado teórico dado no Teorema 6.0.1 (a), ou seja, independentemente do valor da variância σ^2, tem-se que $E[\tilde{x}_k] = x_k$.

Exemplo 2. Seja a planta como na Seção 5.2.5

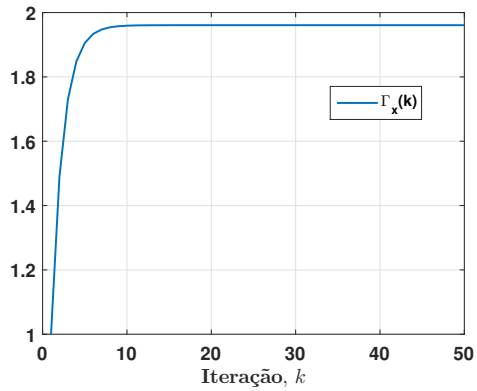
$$\tilde{\mathbf{x}}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\tilde{\mathbf{u}}_k, \tag{6.45}$$

onde

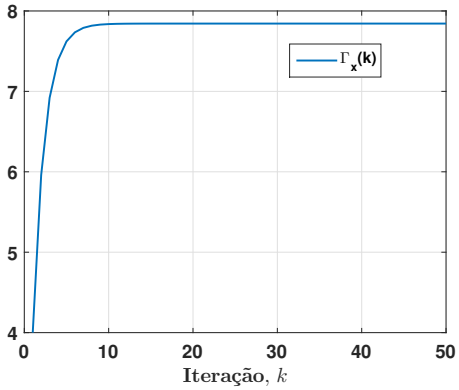
$$\mathbf{A} = \begin{bmatrix} -0,6 & -0,4 \\ 1,0 & 0,0 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 0,0 \\ 1,0 \end{bmatrix}. \tag{6.46}$$



(a) $\sigma^2 = (0,1)^2$



(b) $\sigma^2 = (0,5)^2$



(c) $\sigma^2 = 1$

Figura 18 – Convergência da variância $\Gamma_{\tilde{x}}(k)$.

O estado inicial é $\mathbf{x}_0 = [1\ 1]^T$. Seja a lei de controle

$$\tilde{\mathbf{u}}_k = -[0,2 \quad -0,2]\tilde{\mathbf{x}}_k + \mathbf{v}_k \tag{6.47}$$

com $E[\mathbf{v}_k] = \mathbf{0}$ e $\Gamma_{\mathbf{v}}(k) = \sigma^2\mathbf{I}$.

A matriz de autocovariância dos estado em regime permanente para o ganho $\mathbf{K} = [0,2 \quad -0,2]$ e variância σ^2 é (Teorema 6.0.2 (a))

$$\Gamma_{\tilde{\mathbf{x}}}(\infty) = \sigma^2\mathbf{S}, \tag{6.48}$$

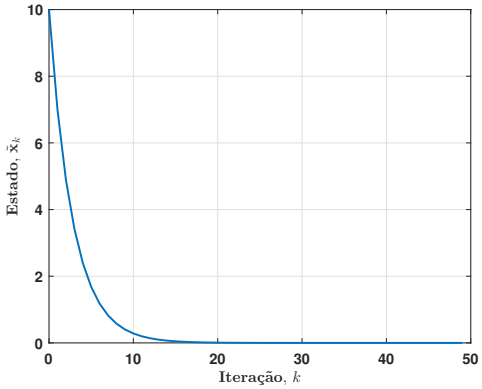
onde

$$\mathbf{S} = \begin{bmatrix} 0,1875 & -0,1250 \\ -0,1250 & 1,1250 \end{bmatrix} \tag{6.49}$$

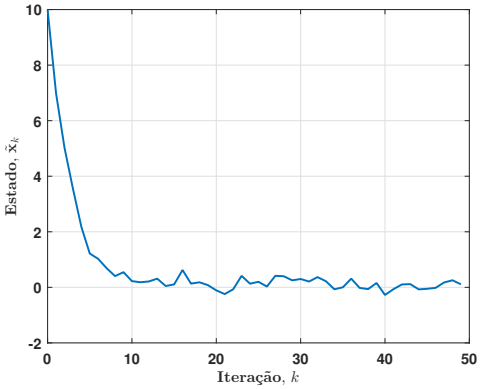
é a solução da equação de Lyapunov

$$(\mathbf{A}-\mathbf{BK})\mathbf{S}(\mathbf{A}-\mathbf{BK})^\top - \mathbf{S} + \mathbf{B}\mathbf{B}^\top = \mathbf{0}. \tag{6.50}$$

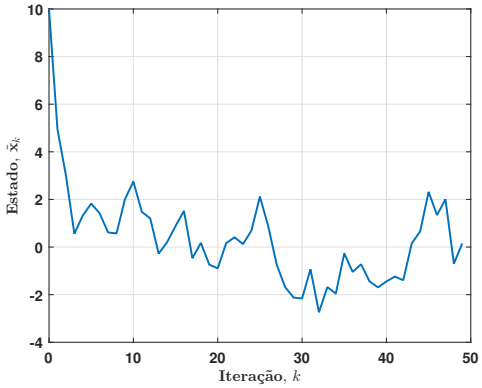
A Figura 21 mostra o comportamento dos estados do sistema (6.45). As Figuras 21 (a)-(b) mostram uma realização do comportamento dos estados $\tilde{\mathbf{x}}_{1k}$ e $\tilde{\mathbf{x}}_{2k}$, respectivamente,



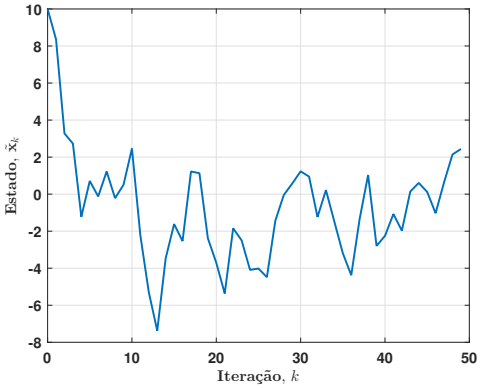
(a) Uma realização sem ruído de prova.



(b) Realização com $\sigma^2 = (0,1)^2$.



(c) Realização com $\sigma^2 = (0,5)^2$.



(d) Realização com $\sigma^2 = 1$.

Figura 19 – Realizações do sistema em (6.39).

quando não é adicionado um ruído de prova ao sistema. As Figuras 21 (c)-(d) mostram uma realização do comportamento dos estados $\tilde{\mathbf{x}}_{1k}$ e $\tilde{\mathbf{x}}_{2k}$, respectivamente, quando é adicionado um ruído de prova de variância $\sigma^2 = 1$. As Figuras 21 (e)-(f) mostram uma simulação de Monte Carlo (MC) de 1000 realizações dos estados $\tilde{\mathbf{x}}_{1k}$ e $\tilde{\mathbf{x}}_{2k}$, respectivamente, quando é adicionado um ruído de prova de variância $\sigma^2 = 1$.

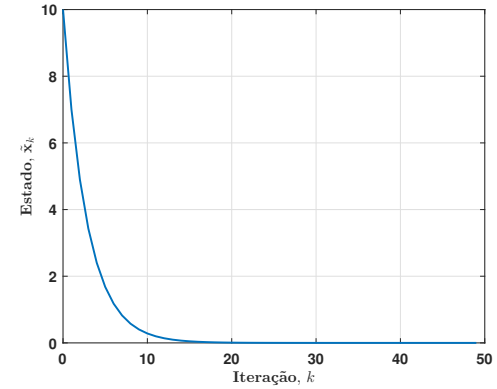
6.2 Efeito na estimação da função Q devido ao ruído de prova

É claro que o ruído de prova afeta os estados e entradas do sistema real. Embora, o ruído de prova não produz bias na estimação da função valor $Q^{\mathbf{h}}(\cdot,\cdot)$ (LEWIS; VRABIE, 2009), porém pode afetar a velocidade de convergência do algoritmo de estimação de parâmetros que resolve a equação de Bellman (5.53).

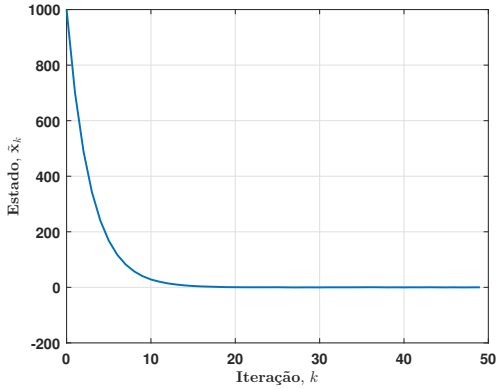
Seja

$$\varphi_k = \bar{\mathbf{z}}_k - \bar{\mathbf{z}}_{k+1} \tag{6.51}$$

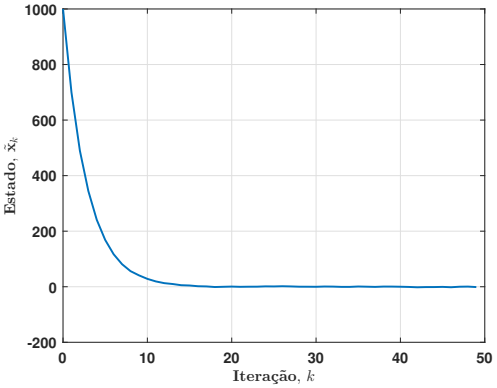
$$r_k = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k. \tag{6.52}$$



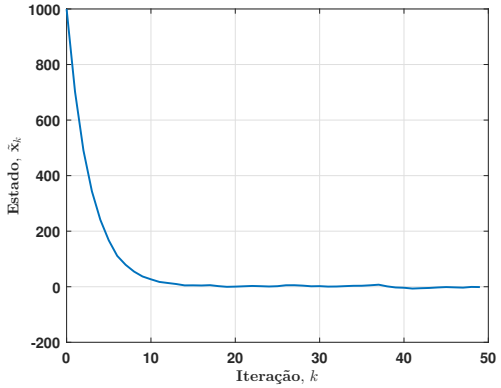
(a) Uma realização sem ruído de prova.



(b) Simulação de MC com $\sigma^2 = (0,1)^2$.



(c) Simulação de MC com $\sigma^2 = (0,5)^2$.



(d) Simulação de MC com $\sigma^2 = 1$.

Figura 20 – Simulação de Monte Carlo (MC) de 1000 realizações do sistema em (6.39). Note que os resultados experimentais em (b)-(d) coincidem com o resultado teórico no Teorema 6.0.1 (a) no sentido de que $E[\tilde{x}_k] = x_k$.

A recursão para o RLS é dado por (HAYKIN, 1986)

$$\theta(0) = \mathbf{0} \tag{6.53}$$

$$\mathbf{P}(0) = \beta\mathbf{I} \tag{6.54}$$

$$\epsilon(i) = r_k - \varphi_k^\top \theta(i-1) \tag{6.55}$$

$$\theta(i) = \theta(i-1) + \frac{\mathbf{P}(i-1)\varphi_k \epsilon(i)}{\lambda + \varphi_k^\top \mathbf{P}(i-1)\varphi_k} \tag{6.56}$$

$$\mathbf{P}(i) = \lambda^{-1}\left[\mathbf{P}(i-1) - \frac{\mathbf{P}(i-1)\varphi_k\varphi_k^\top \mathbf{P}(i-1)}{\lambda + \varphi_k^\top \mathbf{P}(i-1)\varphi_k}\right], \tag{6.57}$$

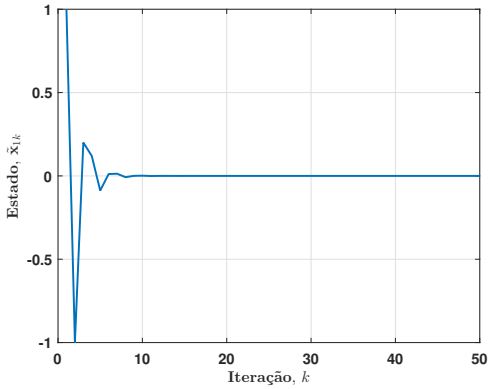
onde $0 < \lambda \leq 1$ é o fator de esquecimento.

A recursão para o NLMS é dado por (HAYKIN, 1986)

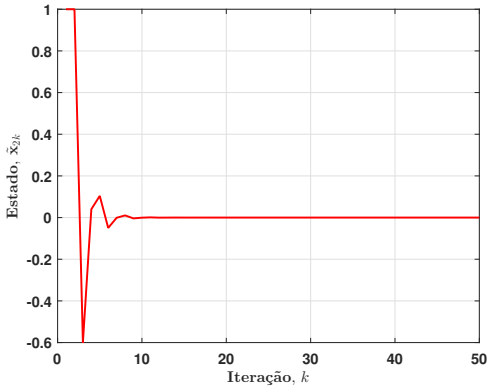
$$\theta(0) = \mathbf{0} \tag{6.58}$$

$$\epsilon(i) = r_k - \varphi_k^\top \theta(i-1) \tag{6.59}$$

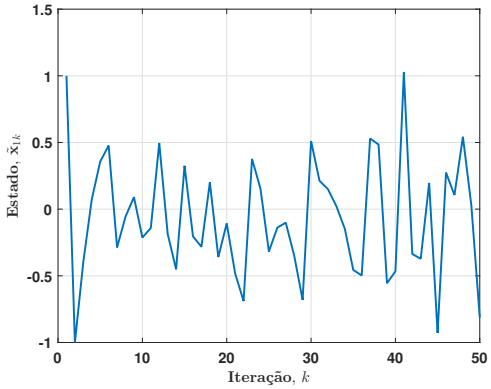
$$\theta(i) = \theta(i-1) + \frac{\mu}{\psi + \varphi_k^\top \varphi_k}\epsilon(i)\varphi_k, \tag{6.60}$$



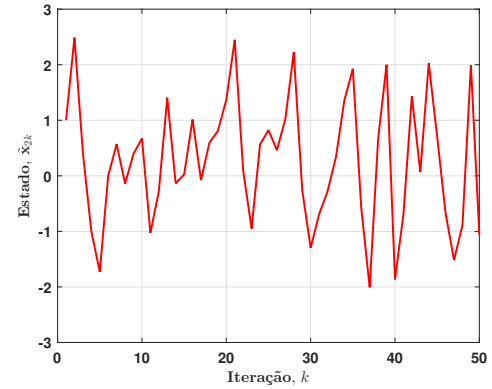
(a) Uma realização sem ruído de prova.



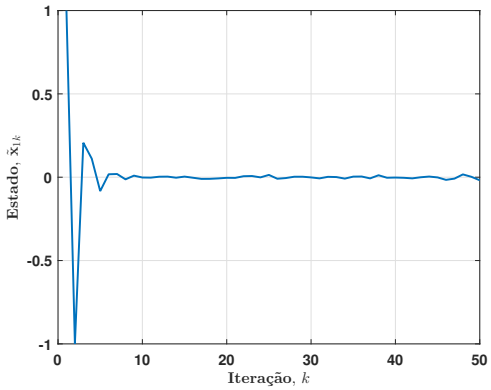
(b) Uma realização sem ruído de prova.



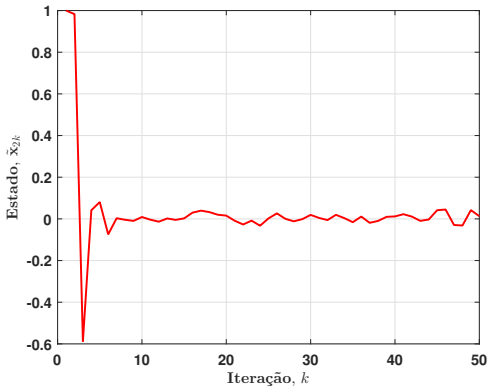
(c) Uma realização com $\sigma^2 = 1$.



(d) Uma realização com $\sigma^2 = 1$.



(e) Simulação de MC com $\sigma^2 = 1$.



(f) Simulação de MC com $\sigma^2 = 1$.

Figura 21 – (a)-(b) Mostra uma realização do sistema (6.45) sem ruído de prova. (c)-(d) Mostra uma realização do sistema (6.45) com ruído de prova de variância $\sigma^2 = 1$. Nesse caso, devido ao ruído de prova, a sequência do vetor de estado não converge para o vetor nulo. (e)-(f) Mostra uma simulação de Monte Carlo (MC) de 1000 realizações do sistema em (6.45). Nesse caso, a sequência da média do vetor de estado converge para o vetor nulo, coincidindo com o resultado teórico no Teorema 6.0.1 (a).

sendo $0 < \mu < 2$ a taxa de aprendizagem e $\psi \geq 0$ um parâmetro para evitar divisão por zero. Os algoritmos RLS e NLMS convergem para os verdadeiros parâmetros θ if φ_k satisfaz a condição de persistência de excitação (5.20)-(5.21), respectivamente. Como foi mostrado no Capítulo 5, um ruído de prova é adicionado à ação de controle para obter a condição de persistência de excitação.

Algoritmo 12 Iteração de política Q learning

1: Parâmetros $\theta_0(0,0)$ e $\mathbf{K}(0)$ matriz de ganho admissível
2: $k = 0$ (Tempo)
3: **for** $j = 0, 1, \ldots, M$ **do**
4: **for** $i = 1, \ldots, N$ **do**
5: $\mathbf{u}_k = \mathbf{K}(j)\mathbf{x}_k + \mathbf{v}_k$
6: $\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k$
7: Atualização $\theta_k(j,i) = f\left(\theta_k(j,i-1)\right)$
8: $k = k + 1$
9: **end for**
10: $\Theta_k(j)$ = reconstrução de $\theta_k(j,N)$
11: $\mathbf{K}(j+1) = -(\Theta(i)_{uu})^{-1}\Theta(i)_{ux}$
12: Inicialização de parâmetros $\theta_k(j+1,0) = \theta_k(j,N)$
13: **end for**

Para cada $j = 0, 1, 2, \ldots,$ o Algoritmo 12 resolve uma equação como

$$(\bar{\mathbf{z}}_k - \bar{\mathbf{z}}_{k+1})^\top \theta(j) = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k \tag{6.61}$$

e encontra uma nova matriz

$$\mathbf{K}(j+1) = -(\Theta(i)_{uu})^{-1}\Theta(i)_{ux} = -(\mathbf{R} + \mathbf{B}^\top \mathbf{P}_j \mathbf{B})^{-1}\mathbf{B}^\top \mathbf{P}_j \mathbf{A}. \tag{6.62}$$

Para resolver (6.61), o Algoritmo 12 usa um filtro adaptativo como na Figura 3 no Capítulo 1, sendo $f(\cdot)$ a atualização dos parâmetros via algoritmo NLMS ou RLS. Os parâmetros do filtro converge para os verdadeiros parâmetros θ se o sinal $\bar{\mathbf{z}}_k - \bar{\mathbf{z}}_{k+1}$ satisfaz a condição de persistência de excitação (6.56)-(6.60), quando é usado RLS ou NLMS, respectivamente. Para obter a condição de persistência de excitação, é adicionado um ruído de prova $\mathbf{v}_k$ à ação de controle.

Portanto, se $\mathbf{K}(0)$ é uma matriz de ganho admissível e $\bar{\mathbf{z}}_k - \bar{\mathbf{z}}_{k+1}$ satisfaz a condição de persistência de excitação (6.56) ou (6.60), então o algoritmo de Hewer (HEWER, 1971) garante que o Algoritmo 12 gera uma sequência de matrizes admissíveis $\mathbf{K}(i)$ tal que $\mathbf{K}(i) \to \mathbf{K}^*$ quando $i \to \infty$.

6.2.1 Métricas para grau de persistência

Os desempenhos dos estimadores RLS e NLMS no Algoritmo 12 podem ser comparado numericamente usando a métrica RMS (*root mean square*)

$$RMS(\theta_k) = \frac{1}{k}\sqrt{\sum_{t=1}^{k}\left(\sum_{i=1}^{n_\theta}[(\theta_t - \theta^*)^i]^2\right)} \tag{6.63}$$

onde $\theta_k = \theta_k(i, j-1)$ é dado no Algoritmo 12, θ^* é a vetorização da matriz

$$\Theta^* = \begin{bmatrix} \Theta^*_{xx} & \Theta^*_{xu} \\ \Theta^*_{ux} & \Theta^*_{uu} \end{bmatrix} = \begin{bmatrix} \mathbf{Q} + \mathbf{A}^\top\mathbf{S}^*\mathbf{A} & \mathbf{A}^\top\mathbf{S}^*\mathbf{B} \\ \mathbf{B}^\top\mathbf{S}^*\mathbf{A} & \mathbf{R} + \mathbf{B}^\top\mathbf{S}^*\mathbf{B} \end{bmatrix} \tag{6.64}$$

e $\mathbf{S}^*$ é solução da DARE (3.85).

Para quantificar o efeito do ruído de prova nos estados, será usado

$$RMS(\tilde{\mathbf{x}}_k) = \frac{1}{k}\sqrt{\sum_{t=1}^{k}\left(\sum_{i=1}^{n}(\tilde{\mathbf{x}}_t^i)^2\right)} \tag{6.65}$$

onde $\tilde{\mathbf{x}}_k$ é gerado por (6.2a).

Para quantificar o grau de persistência de excitação, será usado

$$RMS(\varphi_k) = \frac{1}{k}\sqrt{\sum_{t=1}^{k}\left(\sum_{i=1}^{n}(\varphi_t^i)^2\right)} \tag{6.66}$$

onde φ_t é dado em (6.51).

6.3 Simulação computacional

Exemplo 1. Seja a planta como na Seção 5.2.5

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \tag{6.67}$$

onde

$$\mathbf{A} = \begin{bmatrix} -0,6 & -0,4 \\ 1,0 & 0,0 \end{bmatrix},\ \mathbf{B} = \begin{bmatrix} 0,0 \\ 1,0 \end{bmatrix}. \tag{6.68}$$

O estado inicial é $\mathbf{x}_0 = [1\ 1]^\top$. Seja a matriz de ganho inicial admissível

$$\mathbf{K}(0) = [0,2 \quad -0,2]. \tag{6.69}$$

Nesse caso, a matriz de ganho ótima é

$$\mathbf{K}^* = [0,4630 \quad -0,0708]. \tag{6.70}$$

A matriz de autocovariância dos estado em regime permanente para o ganho ótimo $\mathbf{K}^*$ em (6.70) e variância σ^2 é (Teorema 6.0.2 (a))

$$\Gamma[\tilde{\mathbf{x}}_\infty] = \sigma^2\mathbf{S}, \tag{6.71}$$

onde

$$\mathbf{S} = \begin{bmatrix} 0,2071 & -0,0769 \\ -0,0769 & 1,0592 \end{bmatrix} \tag{6.72}$$

é a solução da equação de Lyapunov

$$(\mathbf{A} - \mathbf{B}\mathbf{K}^*)\mathbf{S}(\mathbf{A} - \mathbf{B}\mathbf{K}^*)^\top - \mathbf{S} + \mathbf{B}\mathbf{B}^\top = \mathbf{0}. \tag{6.73}$$

A Tabela 10 mostra os parâmetros usados em (6.54), (6.56) e (6.57) para o algoritmo RLS e (6.60) para o algoritmo NLMS no Exemplo 1.

Tabela 10 – Parâmetros para RLS e NLMS no Exemplo 1.

Algoritmos	Parâmetros			
	λ	μ	ψ	$\mathbf{P}(0)$
RLS	1	-	-	$100\mathbf{I}$
NLMS	-	1	0	-

A Figura 22 mostra o comportamento dos estados e convergência dos parâmetros da função valor de ação $Q^{\mathbf{K}^*}(\cdot,\cdot)$ e da matriz de ganho ótimo $\mathbf{K}^*$ quando a variância do ruído de prova é $\sigma^2 = (0,01)^2$. Nesse caso, o algoritmo RLS falha na convergência. Note na Figura 22 (a) que os estados do sistema não estão sendo regulados, ou seja, não estão convergindo para zero.

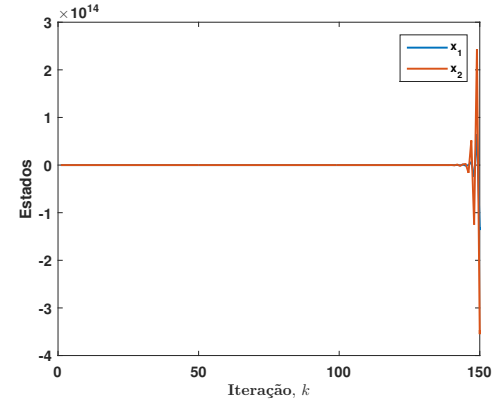
A Figura 23 mostra o comportamento dos estados do sistema para as variâncias $\sigma^2 = (0,1)^2, \sigma^2 = (0,5)^2$ e $\sigma^2 = 1$. Note que quando a variância aumenta, os estados do sistema oscilam mais longe do vetor nulo. Como um dos objetivos do problema do LQR é levar os estados do sistema assintoticamente para o vetor nulo, é desejável que a variância seja o menor possível.

As Figuras 24, 25 e 26 mostram a convergência dos parâmetros da função valor de ação $Q^{\mathbf{K}^*}(\cdot,\cdot)$ e da matriz de ganho ótimo $\mathbf{K}^*$ para as variâncias $\sigma^2 = (0,1)^2, \sigma^2 = (0,5)^2$ e $\sigma^2 = 1$. Nesse caso, quando a variância é menor, o algoritmo NLMS tem o melhor desempenho do que do algoritmo RLS.

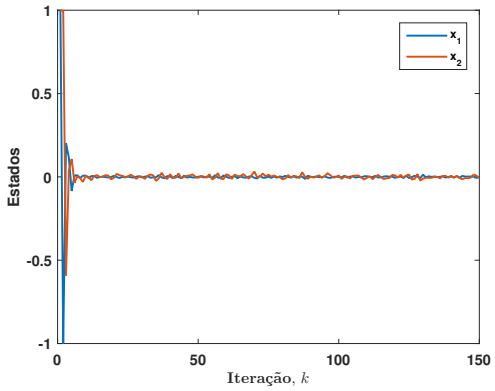
A Tabela 11 mostra o RMS para $\tilde{\mathbf{x}}_k, \theta_k$ e φ_k em regime permanente.

Exemplo 2. Seja a planta linear invariante como na Seção 3.2.6

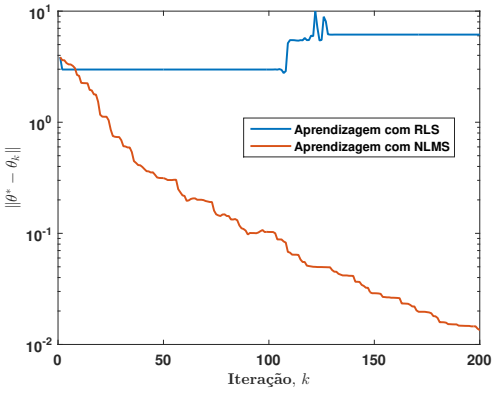
$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \tag{6.74}$$



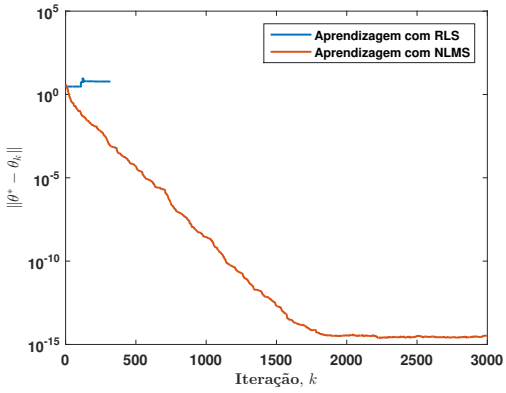
(a) Comportamento dos estados com RLS.



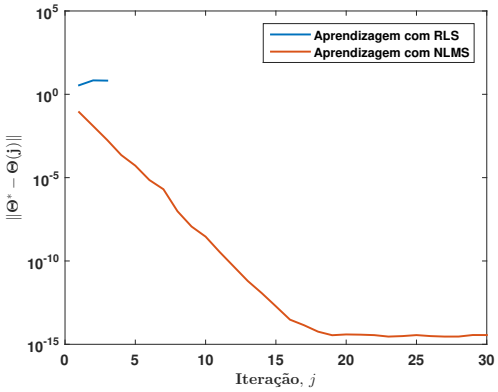
(b) Comportamento dos estados com NLMS.



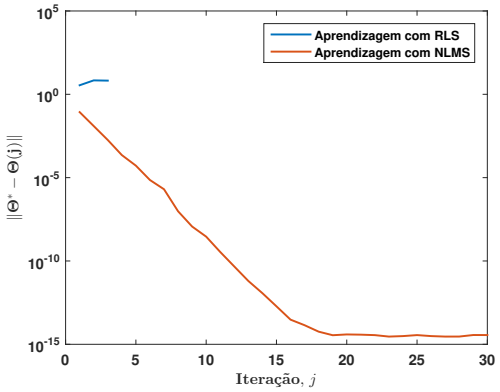
(c) Convergência de $\|\theta^* - \theta_k\|$.



(d) Convergência de $\|\theta^* - \theta_k\|$.

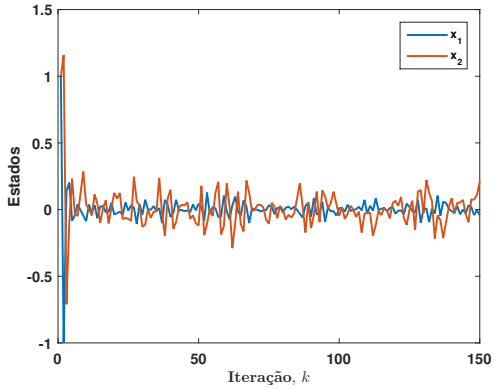


(e) Convergência de $\|\Theta^* - \Theta(j)\|$.

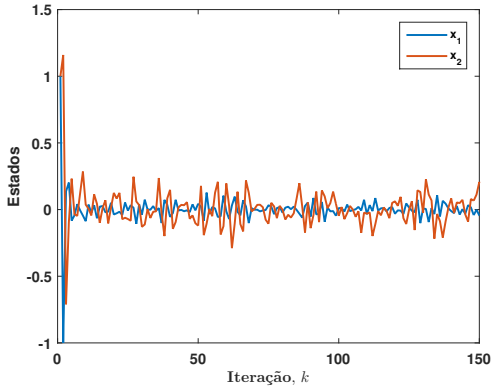


(f) Convergência de $\|\mathbf{K}^* - \mathbf{K}(j)\|$.

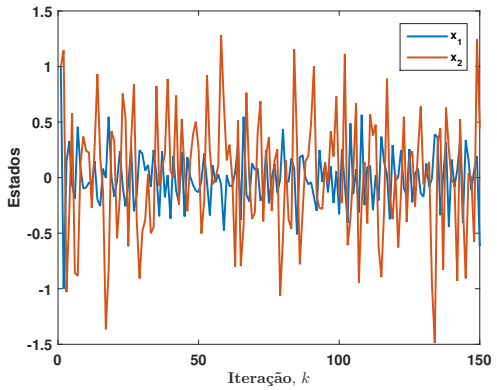
Figura 22 – Aprendizagem com $\sigma^2 = (0,01)^2$ no Exemplo 1. Note que o algoritmo RLS falha na convergência dos parâmetros e portanto o controlador não estabiliza a planta.



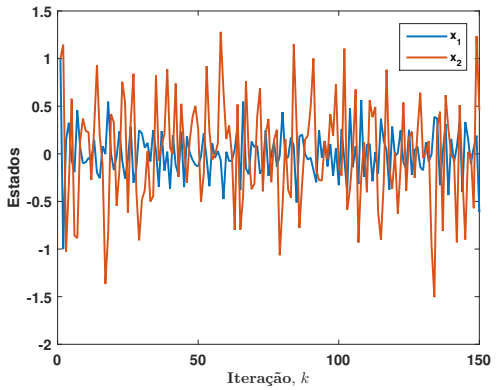
(a) Aprendizagem com RLS e $\sigma^2 = (0,1)^2$.



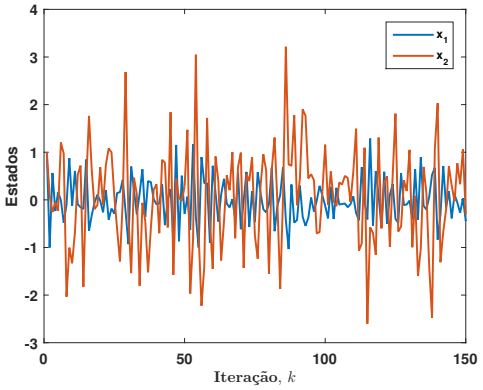
(b) Aprendizagem com NLMS e $\sigma^2 = (0,1)^2$.



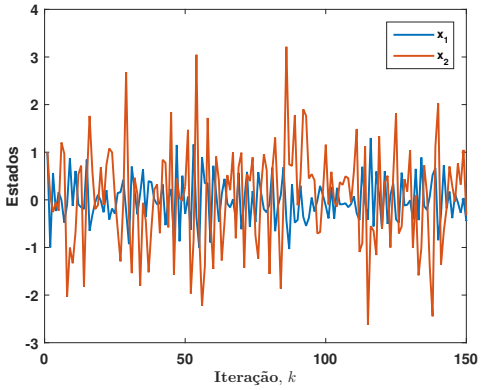
(c) Aprendizagem com RLS e $\sigma^2 = (0,5)^2$.



(d) Aprendizagem com NLMS e $\sigma^2 = (0,5)^2$.



(e) Aprendizagem com RLS e $\sigma^2 = 1$.

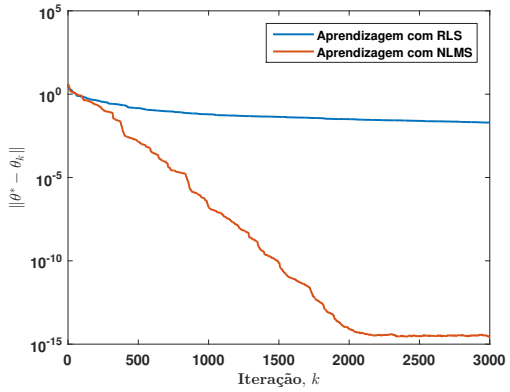


(f) Aprendizagem com NLMS e $\sigma^2 = 1$.

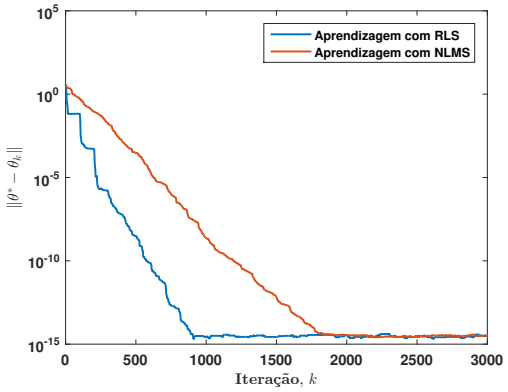
Figura 23 – Comportamento dos estados x_1 e x_2 durante o processo de aprendizagem usando os algoritmos RLS e NLMS para diferentes variâncias σ^2 no Exemplo 1.

Tabela 11 – RMS para $\tilde{\mathbf{x}}_k$, θ_k e φ_k para o Exemplo 1 em regime permanente.

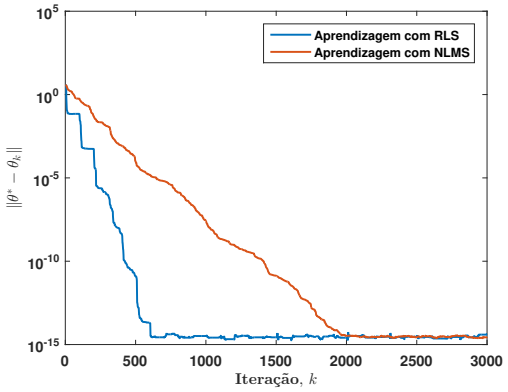
σ^2	$RMS(\tilde{\mathbf{x}}_k)$		$RMS(\theta_k)$		$RMS(\varphi_k)$	
	RLS	NLMS	RLS	NLMS	RLS	NLMS
$(0,01)^2$	Não converge	0,000182	Não converge	0,002389	Não converge	0,000298
$(0,1)^2$	0,012681	0,012675	0,016092	0,004116	0,001830	0,001830
$(0,5)^2$	0,111413	0,111412	0,003311	0,014800	0,113890	0,113892
1	1,286409	1,286553	0,005070	0,023974	14,845539	14,847075



(a) Aprendizagem com $\sigma^2 = (0,1)^2$.



(b) Aprendizagem com $\sigma^2 = (0,5)^2$.

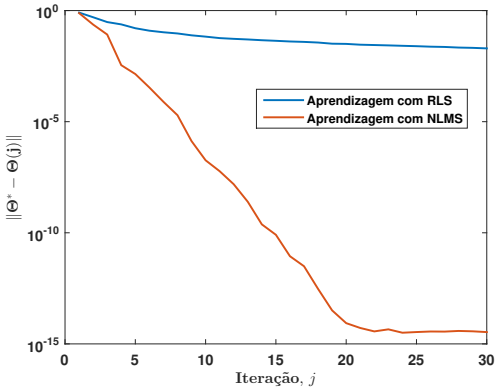


(c) Aprendizagem com $\sigma^2 = 1$.

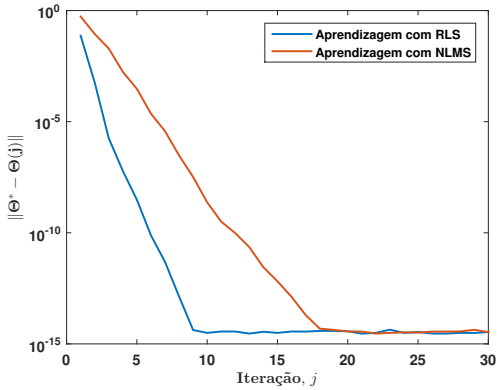
Figura 24 – Convergência dos parâmetros θ_k durante o processo de aprendizagem para diferentes variâncias σ^2 no Exemplo 1.

onde

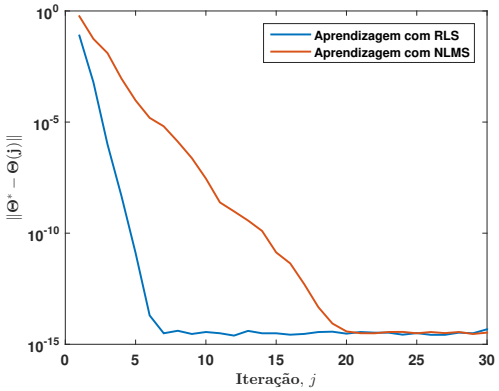
$$\mathbf{A} = \begin{bmatrix} 0,912 & 0,083 & -0,001 \\ 0,372 & 0,943 & -0,010 \\ 0,000 & 0,000 & 0,368 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0,000 & 0,043 \\ -0,006 & 0,968 \\ 0,632 & 0,000 \end{bmatrix}. \tag{6.75}$$



(a) Aprendizagem com $\sigma^2 = (0,1)^2$.



(b) Aprendizagem com $\sigma^2 = (0,5)^2$.



(c) Aprendizagem com $\sigma^2 = 1$.

Figura 25 – Convergência dos parâmetros $\Theta(j)$ durante o processo de aprendizagem para diferentes variâncias σ^2 no Exemplo 1.

O estado inicial é $\mathbf{x}_0 = [3\ 5\ 2]^T$. Seja a matriz de ganho admissível

$$\mathbf{K}(0) = \begin{bmatrix} -0,0005 & -0,0011 & 0,4655 \\ 0,3843 & 0,8817 & -0,0070 \end{bmatrix}. \tag{6.76}$$

Nesse caso, a matriz de ganho ótima é

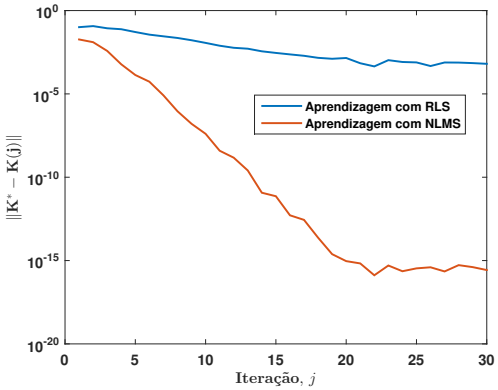
$$\mathbf{K}^* = \begin{bmatrix} -0,0049 & -0,0038 & 0,1782 \\ 0,5633 & 0,6130 & -0,0066 \end{bmatrix}. \tag{6.77}$$

A matriz de autocovariância dos estado em regime permanente para o ganho ótimo (6.77) e variância σ^2 é (Teorema 6.0.2 (a))

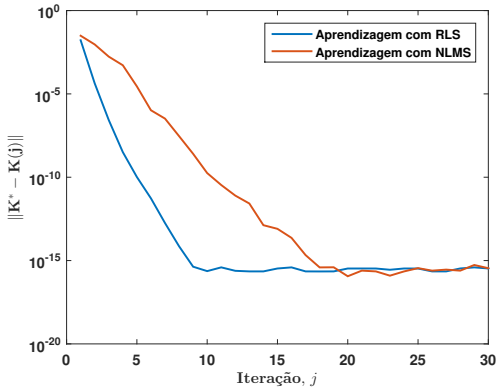
$$\Gamma[\tilde{\mathbf{x}}_\infty] = \sigma^2\mathbf{S} \tag{6.78}$$

onde

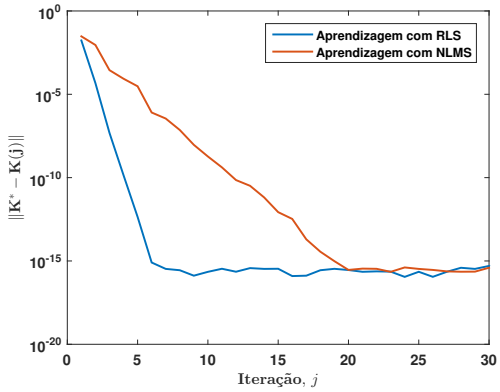
$$\mathbf{S} = \begin{bmatrix} 0,0609 & 0,0761 & 0,0005 \\ 0,0761 & 1,0591 & -0,0035 \\ 0,0005 & -0,0035 & 0,4273 \end{bmatrix} \tag{6.79}$$



(a) Aprendizagem com $\sigma^2 = (0,1)^2$.



(b) Aprendizagem com $\sigma^2 = (0,5)^2$.



(c) Aprendizagem com $\sigma^2 = 1$.

Figura 26 – Convergência da matriz de ganho $\mathbf{K}(j)$ durante o processo de aprendizagem para diferentes variâncias σ^2 no Exemplo 1.

Tabela 12 – Parâmetros para RLS e NLMS no Exemplo 2.

Algoritmos	Parâmetros			
	λ	μ	ψ	$\mathbf{P}(0)$
RLS	1	-	-	$100\mathbf{I}$
NLMS	-	1,7	0	-

é a solução da equação de Lyapunov

$$(\mathbf{A} - \mathbf{B}\mathbf{K}^*)\mathbf{S}(\mathbf{A} - \mathbf{B}\mathbf{K}^*)^\top - \mathbf{S} + \mathbf{B}\mathbf{B}^\top = \mathbf{0}. \tag{6.80}$$

A Tabela 12 mostra os parâmetros usados em (6.54) e (6.60) dos algoritmos RLS e NLMS, respectivamente, para o Exemplo 2.

A Figura 27 mostra o comportamento dos estados e convergência dos parâmetros da função valor de ação $Q^{\mathbf{K}^*}(\cdot,\cdot)$ e da matriz de ganho ótimo $\mathbf{K}^*$ quando a variância do ruído de prova é $\sigma^2 = (0,01)^2$. Nesse caso, o algoritmo RLS falha na convergência. Note

na Figura 27 (a) que os estados do sistema não estão sendo regulados, ou seja, não estão convergindo para zero.

A Figura 28 mostra o comportamento dos estados do sistema para as variâncias $\sigma^2 = (0,05)^2, \sigma^2 = (0,5)^2$ e $\sigma^2 = 1$. Como no exemplo anterior, quando a variância aumenta, os estados do sistema tendem a estar longe do vetor nulo. Como um dos objetivos do problema do LQR é levar os estados do sistema assintoticamente para o vetor nulo, é desejável que a variância seja o menor possível.

As Figuras 29, 30 e 31 mostram o comportamento dos estados e convergência dos parâmetros da função valor de ação $Q^{\mathbf{K}^*}(\cdot,\cdot)$ e da matriz de ganho ótimo $\mathbf{K}^*$ para as variâncias $\sigma^2 = (0,05)^2, \sigma^2 = (0,5)^2$ e $\sigma^2 = 1$. Nesse caso, quando a variância é menor, o algoritmo NLMS tem o melhor desempenho do que do algoritmo RLS.

A Tabela 13 mostra o RMS para $\tilde{\mathbf{x}}_k, \theta_k$ e φ_k em regime permanente.

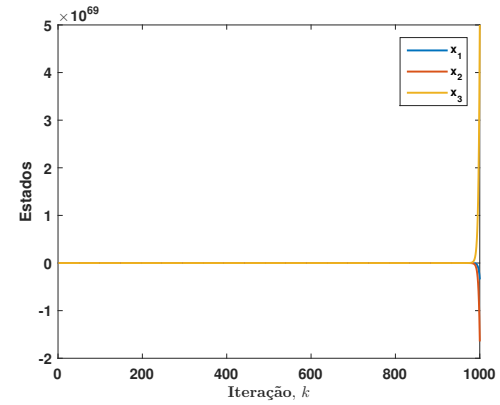
Tabela 13 – RMS para $\tilde{x}_k$, θ_k e φ_k para o Exemplo 2 em regime permanente.

σ^2	$RMS(\tilde{\mathbf{x}}_k)$		$RMS(\theta_k)$		$RMS(\varphi_k)$	
	RLS	NLMS	RLS	NLMS	RLS	NLMS
$(0,01)^2$	Não converge	0,000381	Não converge	0,012313	Não converge	0,010577
$(0,05)^2$	0,015854	0,015857	0,024086	0,010884	0,022702	0,022702
$(0,5)^2$	0,387164	0,387170	0,001801	0,009378	3,051238	3,051306
1	1,544832	1,545338	0,001672	0,104938	48,225908	48,230389

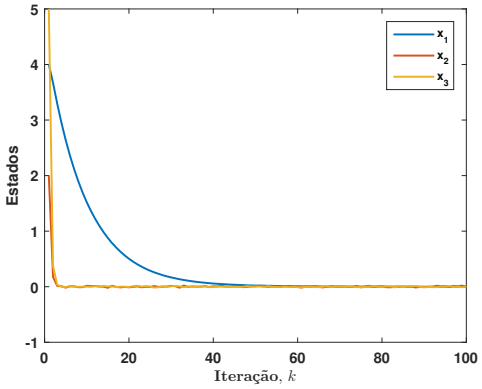
Exemplo 3. Como foi mostrado nos dois exemplos anteriores, o algoritmo RLS falha na adaptação dos parâmetros quando a variância é $\sigma^2 = (0,01)^2$. No entanto, nesses exemplos foi usado como fator de esquecimento $\lambda = 1$. A seguir, será usado um fator de esquecimento $\lambda = 0,5$, em (6.56) e (6.57), nesses caso quando o algoritmo RLS falha na adaptação dos parâmetros.

Seja o sistema de ordem 2 como no Exemplo 1 da presente seção. Ao contrário do caso quando o fator de esquecimento é igual a $\lambda = 1$, o RLS consegue adaptar os parâmetros quando $\lambda = 0,5$ e o Q-learning baseado em RLS tem um melhor desempenho do que do Q-learning baseado em NLMS (Figura 32).

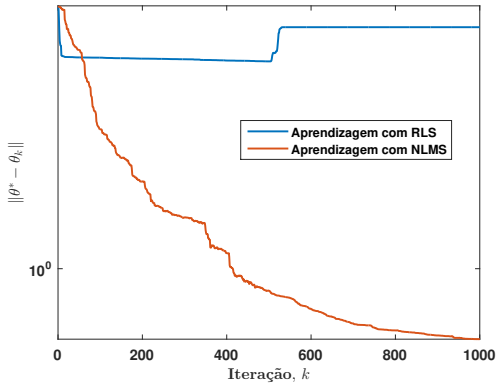
Seja o sistema de ordem 3 como no Exemplo 2 da presente seção. Ao contrário do caso quando o fator de esquecimento é igual a $\lambda = 1$, o RLS consegue adaptar os parâmetros quando $\lambda = 0,5$ e o Q-learning baseado em RLS tem um melhor desempenho do que do Q-learning baseado em NLMS (Figura 33).



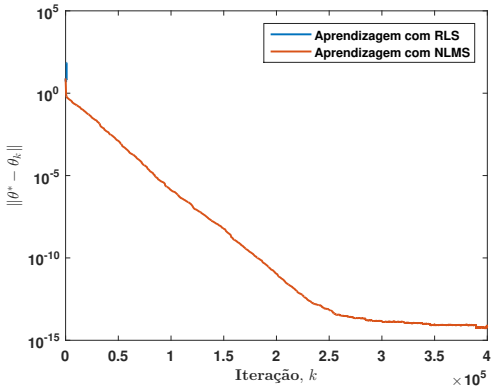
(a) Estados com RLS.



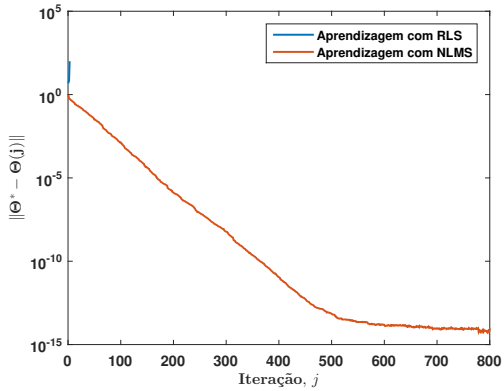
(b) Estados NLMS.



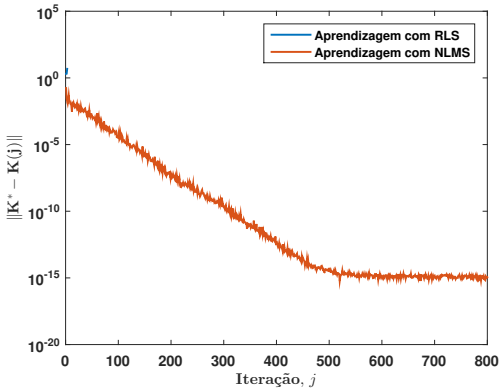
(c) Aprendizagem com $\sigma^2 = (0,01)^2$.



(d) Aprendizagem com $\sigma^2 = (0,01)^2$.

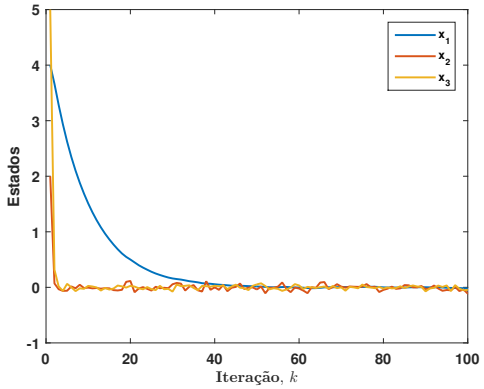


(e) Aprendizagem com $\sigma^2 = (0,01)^2$.

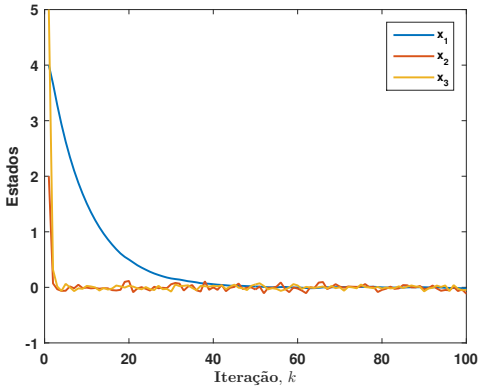


(f) Aprendizagem com $\sigma^2 = (0,01)^2$.

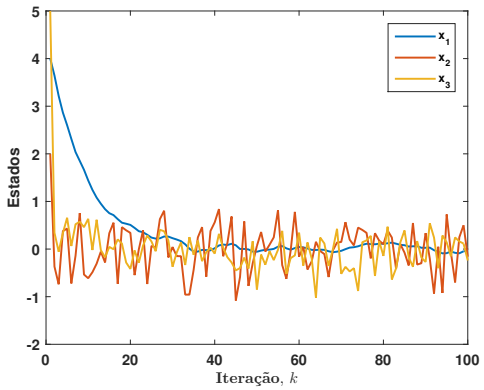
Figura 27 – Aprendizagem com $\sigma^2 = (0,01)^2$ no Exemplo 2. Note que o algoritmo RLS falha na convergência



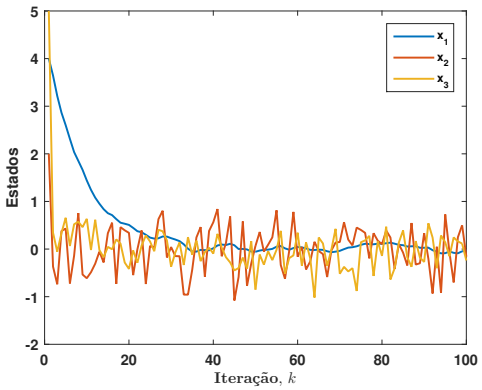
(a) Estados com RLS e $\sigma^2 = (0,05)^2$.



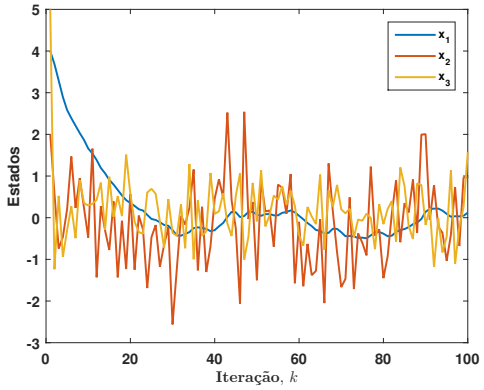
(b) Estados com NLMS e $\sigma^2 = (0,05)^2$.



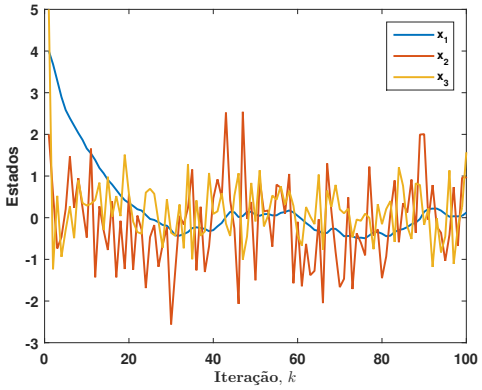
(c) Estados com RLS e $\sigma^2 = (0,5)^2$.



(d) Estados com NLMS e $\sigma^2 = (0,5)^2$.

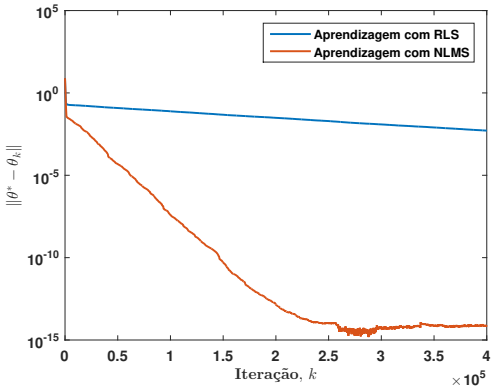


(e) Estados com RLS e $\sigma^2 = 1$.

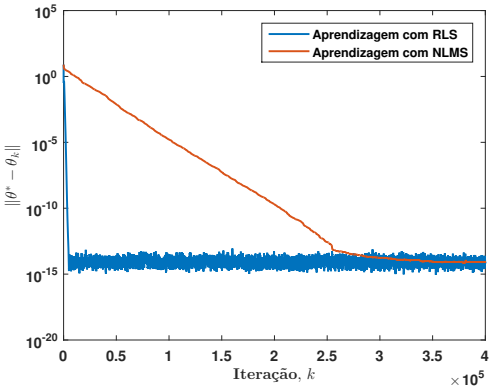


(f) Estados com NLMS e $\sigma^2 = 1$.

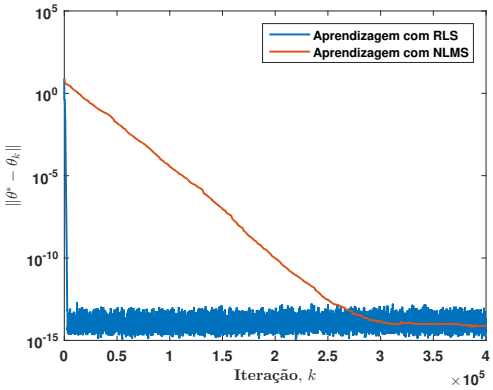
Figura 28 – Compórtamento dos estados x_1, x_2 e x_3 durante o processo de aprendizagem usando os algoritmos RLS e NLMS para diferentes variâncias σ^2 no Exemplo 2.



(a) Aprendizagem com $\sigma^2 = (0,05)^2$.



(b) Aprendizagem com $\sigma^2 = (0,5)^2$.

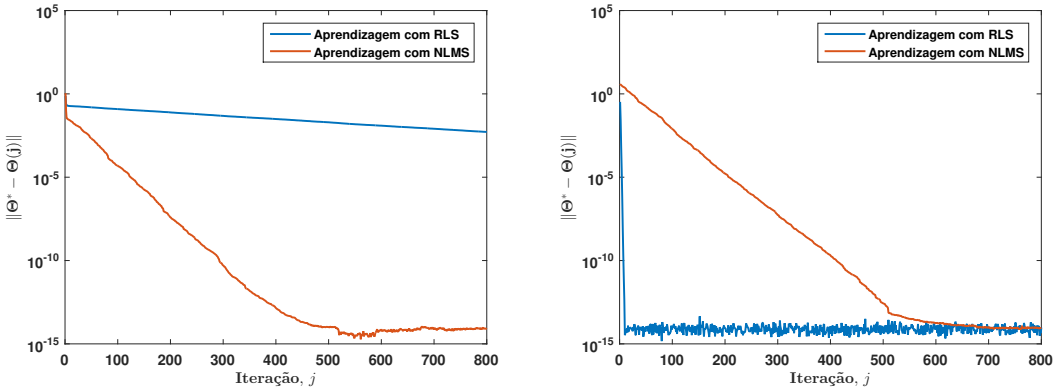


(c) Aprendizagem com $\sigma^2 = 1$.

Figura 29 – Convergência dos parâmetros θ_k durante o processo de aprendizagem para diferentes variâncias σ^2 no Exemplo 2.

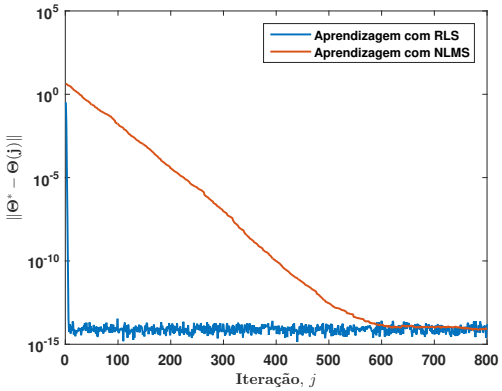
6.4 Considerações finais

Neste capítulo, o efeito do ruído de prova necessário para obter a condição de persistência de excitação foi apresentado. Como era de se esperar, o ruído de prova não produz *bias* nos estados do sistema nem na estimação dos parâmetros da função valor. No entanto, como foi mostrado nas simulações, o ruído de prova afeta o desempenho numérico dos algoritmos de adaptação de parâmetros RLS e NLMS. Além do mais, a norma da matriz de autocovarância dos estados do sistema é proporcional à variância do ruído de prova, o que comprova que quando a variância do ruído de prova aumenta, os estados do sistema tendem a apresentar oscilações de maior amplitude em torno do valor zero.



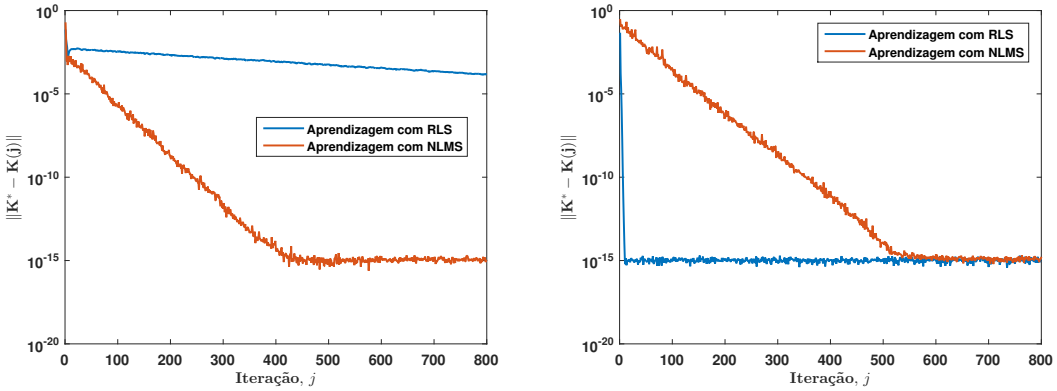
(a) Aprendizagem com $\sigma^2 = (0,05)^2$.

(b) Aprendizagem com $\sigma^2 = (0,5)^2$.



(c) Aprendizagem com $\sigma^2 = 1$.

Figura 30 – Convergência dos parâmetros $\Theta(j)$ durante o processo de aprendizagem para diferentes variâncias σ^2 no Exemplo 2.

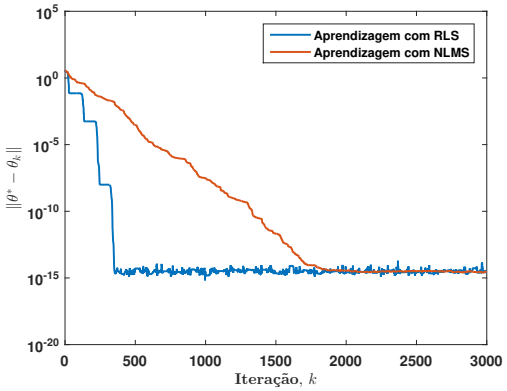


(a) Aprendizagem com $\sigma^2 = (0,05)^2$.

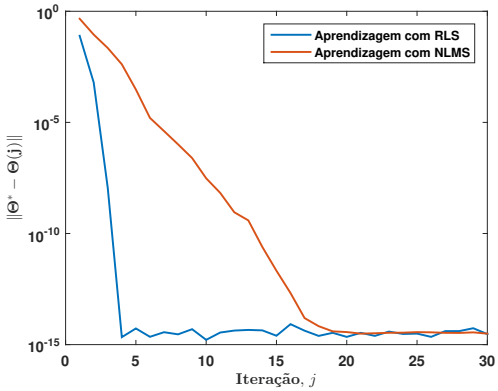
(b) Aprendizagem com $\sigma^2 = (0,5)^2$.

(c) Aprendizagem com $\sigma^2 = 1$.

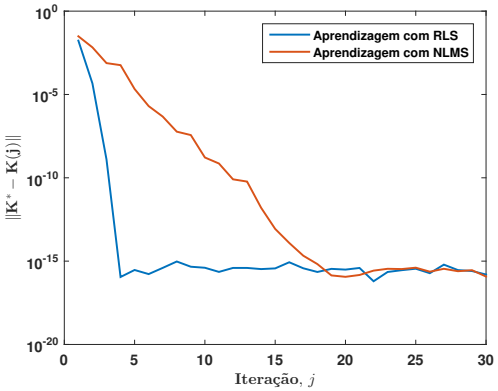
Figura 31 – Convergência da matriz de ganho $\mathbf{K}(j)$ durante o processo de aprendizagem para diferentes variâncias σ^2 no Exemplo 2.



(a) Convergência de $\|\theta^* - \theta_k\|$.

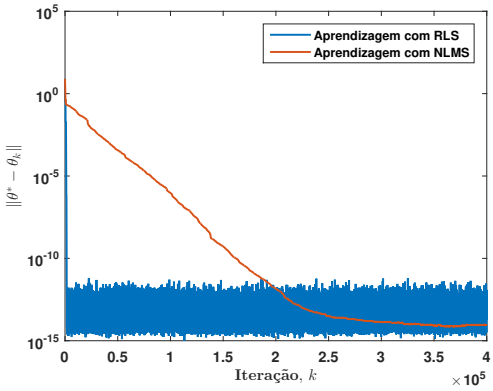


(b) Convergência de $\|\Theta^* - \Theta_k\|$.

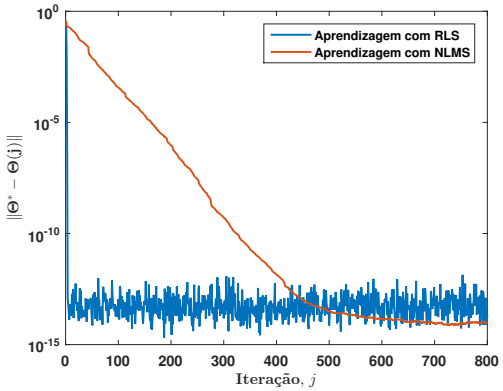


(c) Convergência de $\|\mathbf{K}^* - \mathbf{K}(j)\|$.

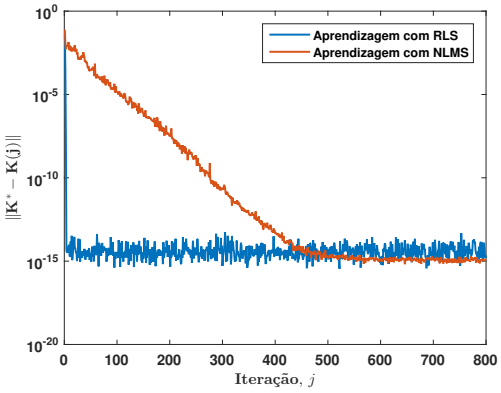
Figura 32 – Convergência dos parâmetros durante o processo de aprendizagem com ruído de prova de variância $\sigma^2 = (0,01)^2$ e fator de esquecimento $\lambda = 0,5$ no RLS. Nesse caso, o algoritmo RLS consegue adaptar os parâmetros e o Q-learning baseado em RLS tem um melhor desempenho do que do Q-learning baseado em NLMS.



(a) Convergência de $\|\theta^* - \theta_k\|$.



(b) Convergência de $\|\Theta^* - \Theta_k\|$.



(c) Convergência de $\|\mathbf{K}^* - \mathbf{K}(j)\|$.

Figura 33 – Convergência dos parâmetros durante o processo de aprendizagem com ruído de prova de variância $\sigma^2 = (0,01)^2$ e fator de esquecimento $\lambda = 0,5$ no RLS. Nesse caso, o algoritmo RLS consegue adaptar os parâmetros e o Q-learning baseado em RLS tem um melhor desempenho do que do Q-learning baseado em NLMS.

7 Realimentação de Saída

O projeto de realimentação de saída é um problema teoricamente desafiador (SADABADI; PEAUCELLE, 2016). Em termos simples, o problema é o seguinte: dado um sistema dinâmico, encontre uma realimentação de saída estática de modo que o sistema em malha fechada tenha alguma característica desejável ou determine que tal realimentação não existe (VESELỲ, 2001; SYRMOS et al., 1997). O problema é frequentemente citado como um dos problemas difíceis abertos em sistemas de controle (BLONDEL; TSITSIKLIS, 2000).

O problema da realimentação de saída em tempo discreto tem sido investigado em Crusius e Trofino (1999), Veselỳ (2001), Garcia, Pradin e Zeng (2001), Rosinová, Veselỳ e Kučera (2003), e para o caso em tempo contínuo tem sido estudado em Levine e Athans (1970), Choi e Sirisena (1977), Trofino-Neto e Kucera (1993), Iwasaki, Skelton e Geromel (1994), Kučera e Souza (1995), Cao, Lam e Sun (1998), Crusius e Trofino (1999), Veselỳ (2001).

Em relação ao problema do LQR com realimentação de saída, Lewis e Vamvoudakis (2011), Rizvi e Lin (2019) utilizam aprendizagem por reforço para resolver o problema do LQR usando dados de entrada-saída. Por outro lado, em Valadbeigi, Sedigh e Lewis (2019), aprendizagem por reforço para resolver o problema de controle H_∞ com realimentação de saída estática é considerado.

7.1 O problema do LQR

Seja a planta linear invariante no

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \tag{7.1a}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k, \tag{7.1b}$$

com $\mathbf{A} \in \mathbb{R}^{n\times n}, \mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$, sendo $\mathbf{C}$ de posto completo.

O sistema (7.1) é estabilizável com realimentação de saída (VESELỲ, 2001; ROSINOVÁ; VESELỲ; KUČERA, 2003) se existe $\mathbf{F} \in \mathbb{R}^{m\times p}$ tal que a matriz $\mathbf{A} + \mathbf{BFC}$ é estável, ou seja se todos seus autovalores estão dentro do círculo unitário.

Seja o índice de desempenho associado à planta (7.1)

$$J(\mathbf{x}_0) = \sum_{k=0}^{\infty}[\mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k] \tag{7.2a}$$

$$= \sum_{k=0}^{\infty}[\mathbf{x}_k^\top \mathbf{C}^\top \mathbf{C}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k] \tag{7.2b}$$

$$= \sum_{k=0}^{\infty}[\mathbf{y}_k^\top \mathbf{y}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k], \tag{7.2c}$$

onde $\mathbf{Q} = \mathbf{C}^\top\mathbf{C}$, $\mathbf{R} = \mathbf{R}^\top > 0$, com $(\mathbf{A}, \mathbf{B})$ estabilizável e $(\mathbf{A}, \mathbf{C})$ detectável.

O problema do LQR com realimentação de saída consiste em encontrar a lei de controle

$$\mathbf{u}_k = \mathbf{F}\mathbf{C}\mathbf{x}_k, \tag{7.3}$$

com $\mathbf{F} \in \mathbb{R}^{m\times p}$ tal que minimize (7.2) e tal que o sistema em malha fechada (7.1) é assintoticamente estável, ou seja tal que a matriz $\mathbf{A} + \mathbf{B}\mathbf{F}\mathbf{C}$ é estável.

7.1.1 Equação de Bellman

Seja $\mathbf{u}_k$ como em (7.3) tal que $\mathbf{A} + \mathbf{B}\mathbf{F}\mathbf{C}$ é estável e seja a função valor

$$V^{\mathbf{F}}(\mathbf{x}_k) = \sum_{i=k}^{\infty}[\mathbf{x}_i^\top \mathbf{Q}\mathbf{x}_i + \mathbf{u}_i^\top \mathbf{R}\mathbf{u}_i]. \tag{7.4}$$

Então

$$V^{\mathbf{F}}(\mathbf{x}_k) = \mathbf{x}_k^\top \mathbf{P}\mathbf{x}_k, \tag{7.5}$$

para alguma matriz $\mathbf{P} = \mathbf{P}^\top > 0$. Logo, de (7.4) e (7.5), obtém-se a equação de Bellman

$$\mathbf{x}_k^\top \mathbf{P}\mathbf{x}_k = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k + \mathbf{x}_{k+1}^\top \mathbf{P}\mathbf{x}_{k+1}. \tag{7.6}$$

Substituindo (7.1a) e (7.3) em (7.6) tem-se

$$\mathbf{x}_k^\top \mathbf{P}\mathbf{x}_k = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k + (\mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k)^\top \mathbf{P}(\mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k) \tag{7.7}$$

$$= \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{x}_k^\top \mathbf{C}^\top \mathbf{F}^\top \mathbf{R}\mathbf{F}\mathbf{C}\mathbf{x}_k + \mathbf{x}_k^\top (\mathbf{A} + \mathbf{B}\mathbf{F}\mathbf{C})^\top \mathbf{P}(\mathbf{A} + \mathbf{B}\mathbf{F}\mathbf{C})\mathbf{x}_k. \tag{7.8}$$

Como (7.8) tem que ser satisfeito para todos os estados $\mathbf{x}_k$, então tem-se a equação de Lyapunov

$$(\mathbf{A} + \mathbf{B}\mathbf{F}\mathbf{C})^\top \mathbf{P}(\mathbf{A} + \mathbf{B}\mathbf{F}\mathbf{C}) - \mathbf{P} + \mathbf{Q} + \mathbf{C}^\top \mathbf{F}^\top \mathbf{R}\mathbf{F}\mathbf{C} = \mathbf{0}. \tag{7.9}$$

Portanto, as equações de Bellman (7.6) e Lyapunov (7.9) são equivalentes.

Derivando (7.7) com relação a $\mathbf{u}_k$, tem-se

$$0 = \mathbf{R}\mathbf{u}_k + \mathbf{B}^\top \mathbf{P}(\mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k) \tag{7.10}$$

$$= \mathbf{R}\mathbf{u}_k + \mathbf{B}^\top \mathbf{P}\mathbf{A}\mathbf{x}_k + \mathbf{B}^\top \mathbf{P}\mathbf{B}\mathbf{u}_k \tag{7.11}$$

$$= (\mathbf{R} + \mathbf{B}^\top \mathbf{P}\mathbf{B})\mathbf{u}_k + \mathbf{B}^\top \mathbf{P}\mathbf{A}\mathbf{x}_k. \tag{7.12}$$

Então,

$$\mathbf{u}_k = -(\mathbf{R} + \mathbf{B}^\top \mathbf{P}\mathbf{B})^{-1}\mathbf{B}^\top \mathbf{P}\mathbf{A}\mathbf{x}_k, \tag{7.13}$$

ou seja

$$\mathbf{F}\mathbf{C} = -(\mathbf{R} + \mathbf{B}^\top \mathbf{P}\mathbf{B})^{-1}\mathbf{B}^\top \mathbf{P}\mathbf{A}, \tag{7.14}$$

e assim

$$\mathbf{F} = -(\mathbf{R} + \mathbf{B}^\top \mathbf{P}\mathbf{B})^{-1}\mathbf{B}^\top \mathbf{P}\mathbf{A}\mathbf{C}^\top (\mathbf{C}\mathbf{C}^T)^{-1}. \tag{7.15}$$

Note que a matriz inversa $(\mathbf{C}\mathbf{C}^\top)^{-1}$ existe já que $\mathbf{C}$ é de posto completo.

Por outro lado, tem-se

$$\begin{aligned}
\mathbf{C}^\top \mathbf{F}^\top \mathbf{B}^T \mathbf{P}\mathbf{A} + \mathbf{C}^\top \mathbf{F}^\top \mathbf{B}^\top \mathbf{P}\mathbf{B}\mathbf{F}\mathbf{C} + \mathbf{C}^\top \mathbf{F}^\top \mathbf{R}\mathbf{F}\mathbf{C} &= && (7.16)\\
\mathbf{C}^\top \mathbf{F}^\top \mathbf{B}^\top \mathbf{P}\mathbf{A} + \mathbf{C}^\top \mathbf{F}^\top (\mathbf{R} + \mathbf{B}^\top \mathbf{P}\mathbf{B})\mathbf{F}\mathbf{C} &= && (7.17)\\
\mathbf{C}^\top \mathbf{F}^\top \mathbf{B}^\top \mathbf{P}\mathbf{A} - \mathbf{C}^\top \mathbf{F}^\top (\mathbf{R} + \mathbf{B}^\top \mathbf{P}\mathbf{B})(\mathbf{R} + \mathbf{B}^\top \mathbf{P}\mathbf{B})^{-1}\mathbf{B}^\top \mathbf{P}\mathbf{A}\mathbf{C}^\top (\mathbf{C}\mathbf{C}^\top)^{-1}\mathbf{C} &= && (7.18)\\
\mathbf{C}^\top \mathbf{F}^\top \mathbf{B}^\top \mathbf{P}\mathbf{A} - \mathbf{C}^\top \mathbf{F}^\top \mathbf{B}^\top \mathbf{P}\mathbf{A}\mathbf{C}^\top (\mathbf{C}\mathbf{C}^\top)^{-1}\mathbf{C} &= && (7.19)\\
\mathbf{C}^\top \mathbf{F}^\top \mathbf{B}^\top \mathbf{P}\mathbf{A}(\mathbf{I} - \mathbf{C}^\top (\mathbf{C}\mathbf{C}^\top)^{-1}\mathbf{C}) &= && (7.20)\\
-\mathbf{C}^\top (\mathbf{C}\mathbf{C}^\top)^{-1}\mathbf{C}\mathbf{A}^\top \mathbf{P}\mathbf{B}(\mathbf{R} + \mathbf{B}^\top \mathbf{P}\mathbf{B})^{-1}\mathbf{B}^\top \mathbf{P}\mathbf{A}(\mathbf{I} - \mathbf{C}^\top (\mathbf{C}\mathbf{C}^\top)^{-1}\mathbf{C}). && (7.21)
\end{aligned}$$

Portanto, substituindo (7.15) na equação de Lyapunov (7.9), obtém-se a equação de Riccati

$$\begin{aligned}
\mathbf{A}^\top \mathbf{P}\mathbf{A} - \mathbf{P} + \mathbf{Q} - \mathbf{A}^\top \mathbf{P}\mathbf{B}(\mathbf{R} + \mathbf{B}^\top \mathbf{P}\mathbf{B})^{-1}\mathbf{B}^\top \mathbf{P}\mathbf{A}\mathbf{C}^\top (\mathbf{C}\mathbf{C}^\top)^{-1}\mathbf{C} && (7.22a)\\
-\mathbf{C}^\top (\mathbf{C}\mathbf{C}^\top)^{-1}\mathbf{C}\mathbf{A}^\top \mathbf{P}\mathbf{B}(\mathbf{R} + \mathbf{B}^\top \mathbf{P}\mathbf{B})^{-1}\mathbf{B}^\top \mathbf{P}\mathbf{A}(\mathbf{I} - \mathbf{C}^\top (\mathbf{C}\mathbf{C}^\top)^{-1}\mathbf{C}) &= 0. && (7.22b)
\end{aligned}$$

Observação 7.1.1 *Note que se* $\mathbf{C}$ *é uma matriz quadrada em* $\mathbb{R}^{n\times n}$*, então* $\mathbf{C}^\top (\mathbf{C}\mathbf{C}^\top)^{-1}\mathbf{C} = \mathbf{I}$*. Logo, o termo esquerdo em (7.22b) é igual à matriz nula e a equação de Riccati em (7.22) torna-se na DARE para o caso de realimentação de estados*

$$\mathbf{A}^\top \mathbf{P}\mathbf{A} - \mathbf{P} + \mathbf{Q} - \mathbf{A}^\top \mathbf{P}\mathbf{B}(\mathbf{R} + \mathbf{B}^\top \mathbf{P}\mathbf{B})^{-1}\mathbf{B}^\top \mathbf{P}\mathbf{A} = 0. \tag{7.23}$$

Também, as equações em (7.14) e (7.15) são equivalentes.

Proposição 7.1.1 *Se* $\mathbf{C}$ *é uma matriz quadrada em* $\mathbb{R}^{n\times n}$*, então o sistema em malha fechada (7.1), com realimentação de saída* $\mathbf{F}$ *em (7.15), é estável.*

Demostração. Como $\mathbf{C}$ é quadrada, então a matriz $\mathbf{F}$ em (7.15) satisfaz

$$0 = \mathbf{F}\mathbf{C} + (\mathbf{R} + \mathbf{B}^\top \mathbf{P}\mathbf{B})^{-1}\mathbf{B}^\top \mathbf{P}\mathbf{A}, \tag{7.24}$$

sendo $\mathbf{P}$ a solução da DARE (7.23). Logo, o resultado segue-se do Teorema 1 em Rosinová, Veselỳ e Kučera (2003).

Algoritmo 13 Método de Hewer para realimentação de saída

1: **Input:** $\mathbf{F}(0)$ tal que $\mathbf{A} + \mathbf{BF}(j)\mathbf{C}$ é estável
2: **for** $j = 0, 1, \ldots$ **do**
3: $\quad (\mathbf{A} + \mathbf{BF}(j)\mathbf{C})^\top \mathbf{P}(j)(\mathbf{A} + \mathbf{BF}(j)\mathbf{C}) - \mathbf{P}(j) + \mathbf{Q} + \mathbf{C}^\top \mathbf{F}^\top(j)\mathbf{RF}(j)\mathbf{C} = 0$
4: $\quad \mathbf{F}(j+1) = -(\mathbf{R} + \mathbf{B}^\top \mathbf{P}(j)\mathbf{B})^{-1}\mathbf{B}^\top \mathbf{P}(j)\mathbf{AC}^\top(\mathbf{CC}^\top)^{-1}$
5: **end for**
6: Se $\mathbf{P}(j) \to \mathbf{P}$, então $\mathbf{F} = -(\mathbf{R} + \mathbf{B}^\top \mathbf{PB})^{-1}\mathbf{B}^\top \mathbf{PAC}^\top(\mathbf{CC}^\top)^{-1}$

Baseados em (7.9) e (7.15), tem-se a seguinte versão do método de Hewer (1971) no Algoritmo 13.

Se a matriz $\mathbf{P}(j)$ converge para $\mathbf{P}$, então a matriz $\mathbf{F}$ no Algoritmo 13 é uma matriz de ganho de saída estabilizável (DEMIR; ÖZBAY, 2020).

Proposição 7.1.2 *Se* $\mathbf{C}$ *é uma matriz quadrada em* $\mathbb{R}^{n\times n}$*, de posto completo, então a sequência de matrizes* $\mathbf{P}_j$ *no Algoritmo 13 converge para a solução da DARE (7.23).*

Demostração. Para todo $j \geq 0$, seja

$$\mathbf{K}_j = \mathbf{F}_j\mathbf{C}. \tag{7.25}$$

Como a matriz $\mathbf{C}$ é invertível (já que é de posto completo), então

$$\mathbf{F}_{j+1} = -(\mathbf{R} + \mathbf{B}^\top \mathbf{P}_j\mathbf{B})^{-1}\mathbf{B}^\top \mathbf{P}_j\mathbf{AC}^\top(\mathbf{CC}^\top)^{-1} \iff \mathbf{K}_{j+1} = -(\mathbf{R} + \mathbf{B}^\top \mathbf{P}_j\mathbf{B})^{-1}\mathbf{B}^\top \mathbf{P}_j\mathbf{A} \tag{7.26}$$

Portanto, o Algoritmo 13 produz uma sequência $\mathbf{P}_j$, $j \geq 0$, de soluções da equação

$$(\mathbf{A} + \mathbf{BK}_j)^\top \mathbf{P}_j(\mathbf{A} + \mathbf{BK}_j) - \mathbf{P}_j + \mathbf{Q} + \mathbf{K}_j^\top \mathbf{RK}_j = 0, \tag{7.27}$$

onde

$$\mathbf{K}_{j+1} = -(\mathbf{R} + \mathbf{B}^\top \mathbf{P}_j\mathbf{B})^{-1}\mathbf{B}^\top \mathbf{P}_j\mathbf{A}. \tag{7.28}$$

Logo, segue-se do método de Hewer (1971), que a sequência $\mathbf{P}_j$ em (7.27) converge para a solução da DARE.

7.2 Simulação computacional

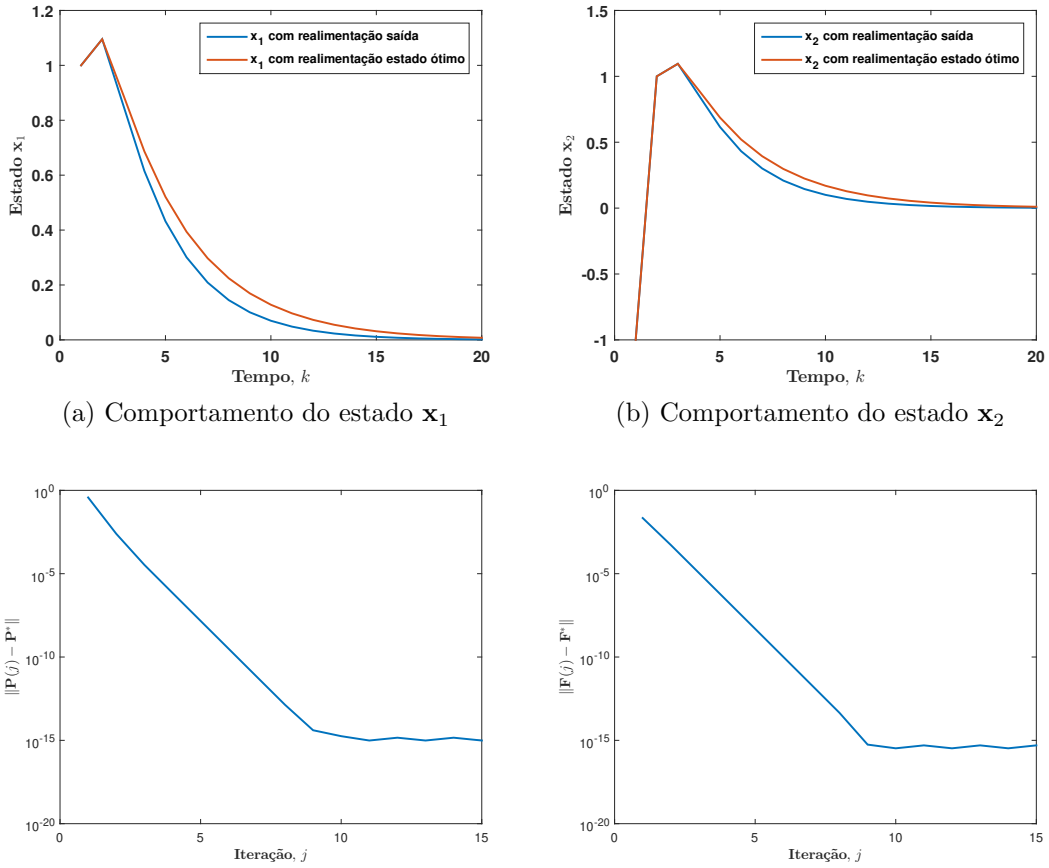
Exemplo 1. Sejam as matrizes como em Rizvi e Lin (2019)

$$\mathbf{A} = \begin{bmatrix} 1,1 & -0,3 \\ 1 & 0 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ e } \mathbf{C} = \begin{bmatrix} 1 & -0,8 \end{bmatrix}. \tag{7.29}$$

Será usado o Algoritmo 14 em Rosinová, Veselỳ e Kučera (2003) para encontrar uma matriz de realimentação de saída $\mathbf{F}^*$

Algoritmo 14 Realimentação de saída

1: **Input:** $\mathbf{G}(j) = 0$
2: **for** $j = 0, 1, \ldots$ **do**
3: $\quad \mathbf{A}^\top\mathbf{P}(j+1)\mathbf{B} - \mathbf{P}(j+1) - \mathbf{A}^\top\mathbf{P}(j+1)\mathbf{B}(\mathbf{R} + \mathbf{B}^\top\mathbf{P}(j+1)\mathbf{B})^{-1}\mathbf{B}^\top\mathbf{P}(j+1)\mathbf{A} + \mathbf{C}^\top\mathbf{C} + \mathbf{G}^\top(j)(\mathbf{R} + \mathbf{B}^\top\mathbf{P}(j+1)\mathbf{B})\mathbf{G}(j) = 0$
4: $\quad \mathbf{G}(j+1) = (\mathbf{R} + \mathbf{B}^\top\mathbf{P}(j+1)\mathbf{B})^{-1}\mathbf{B}^\top\mathbf{P}(j+1)\mathbf{A}[\mathbf{I} - \mathbf{C}^T(\mathbf{C}\mathbf{C}^\top)^{-1}\mathbf{C}]$
5: **end for**
6: Se $\mathbf{P}(j) \to \mathbf{P}$, então $\mathbf{F} = -(\mathbf{R} + \mathbf{B}^\top\mathbf{P}\mathbf{B})^{-1}\mathbf{B}^\top\mathbf{P}\mathbf{A}\mathbf{C}^\top(\mathbf{C}\mathbf{C}^\top)^{-1}$



(a) Comportamento do estado $\mathbf{x}_1$

(b) Comportamento do estado $\mathbf{x}_2$

(c) Convergência: $\mathbf{P}(j) \to \mathbf{P}^*$

(d) Convergência: $\mathbf{F}(j) \to \mathbf{F}^*$

Figura 34 – Exemplo 1. (a) Comportamento do estado $\mathbf{x}_1$. (b) Comportamento do estado $\mathbf{x}_2$. (c) Convergência de $\mathbf{P}(j)$ no Algoritmo 13 à matriz $\mathbf{P}^*$ em (7.30). (d) Convergência de $\mathbf{F}(j)$ no Algoritmo 13 à matriz $\mathbf{F}^*$ em (7.30).

O Algoritmo 14 converge às seguintes matrizes

$$\mathbf{P}^* = \begin{bmatrix} 1,0573 & -0,8457 \\ -0,8457 & 0,6870 \end{bmatrix}, \quad \mathbf{F}^* = [0,1693]. \tag{7.30}$$

Por outro lado, as Figuras 34 (c)-(d) mostram a convergência das sequências $\mathbf{P}(j)$ e $\mathbf{F}(j)$ no Algoritmo 13 às matrizes $\mathbf{P}^*$ e $\mathbf{F}^*$ em (7.30), respectivamente.

Exemplo 2. Sejam as matrizes como na Seção 3.2.6

$$\mathbf{A} = \begin{bmatrix} 0,912 & 0,083 & -0,001 \\ 0,372 & 0,943 & -0,010 \\ 0,000 & 0,000 & 0,368 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 0,000 & 0,043 \\ -0,006 & 0,968 \\ 0,632 & 0,000 \end{bmatrix} \text{ e } \ \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \end{bmatrix} \tag{7.31}$$

O Algoritmo 14 converge para as seguintes matrizes

$$\mathbf{P}^* = \begin{bmatrix} 5,4566 & 0,7121 & -0,0432 \\ 0,7121 & 2,5364 & 1,6270 \\ -0,0432 & 1,6270 & 4,4164 \end{bmatrix}, \ \mathbf{F}^* = \begin{bmatrix} -0,0623 & 0,1599 \\ 0,5306 & 0,1593 \end{bmatrix}. \tag{7.32}$$

Por outro lado, as Figuras 35 (d)-(e) mostram a convergência das sequências $\mathbf{P}(j)$ e $\mathbf{F}(j)$ no Algoritmo 13 às matrizes $\mathbf{P}^*$ e $\mathbf{F}^*$ em (7.32), respectivamente.

7.3 Q-learning com realimentação de saída

O Algoritmo 13 tem a ventagem de resolver uma sequência de equações de Lyapunov (equações lineares em $\mathbf{P}$), enquanto que o Algoritmo 14 resolve uma sequência de equações de Riccati (equações não lineares em $\mathbf{P}$). No entanto, O Algoritmo 13 depende das matrizes do modelo do sistema.

Baseado na equivalência das equações de Bellman (7.6) e Lyapunov (7.9), é proposto o seguinte algoritmo de iteração de política (Algoritmo 15).

Algoritmo 15 Iteração de política para realimentação de saída

1: **Input:** $\mathbf{u}_k = \mathbf{F}(0)\mathbf{y}_k$ estabilizável
2: **for** $j = 0, 1, \ldots$ **do**
3: Resolver a equação de Bellman
$\mathbf{x}_k^\top \mathbf{P}(j)\mathbf{x}_k = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k + \mathbf{x}_{k+1}^\top \mathbf{P}(j)\mathbf{x}_{k+1}$
4: $\mathbf{u}_k = -(\mathbf{R} + \mathbf{B}^\top \mathbf{P}(j)\mathbf{B})^{-1}\mathbf{B}^\top \mathbf{P}(j)\mathbf{A}\mathbf{C}^\top(\mathbf{C}\mathbf{C}^\top)^{-1}\mathbf{y}_k$
5: **end for**
6: Se $\mathbf{P}(j) \to \mathbf{P}$, então $\mathbf{F} = -(\mathbf{R} + \mathbf{B}^\top \mathbf{P}\mathbf{B})^{-1}\mathbf{B}^\top \mathbf{P}\mathbf{A}\mathbf{C}^\top(\mathbf{C}\mathbf{C}^\top)^{-1}$

Para a política de controle em (7.3), a equação de Bellman no Algoritmo 15 será resolvida usando a função $Q^{\mathbf{F}}(\cdot, \cdot)$

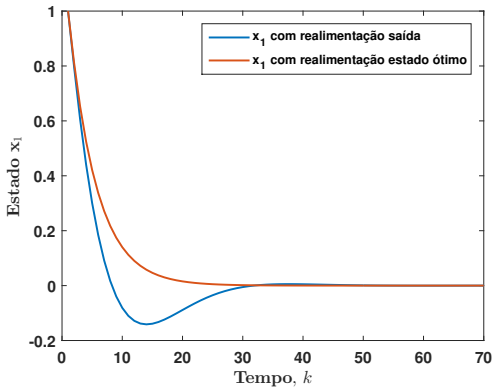
$$Q^{\mathbf{F}}(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k + V^{\mathbf{F}}(\mathbf{x}_{k+1}). \tag{7.33}$$

Note que

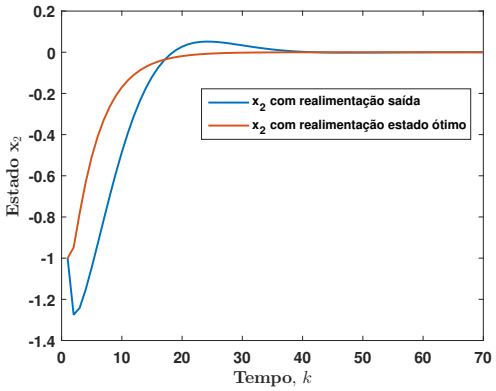
$$V^{\mathbf{F}}(\mathbf{x}_k) = Q^{\mathbf{F}}(\mathbf{x}_k, \mathbf{F}\mathbf{C}\mathbf{x}_k). \tag{7.34}$$

Seguindo as mesmas ideias da Seção 5.2.2, obtém-se

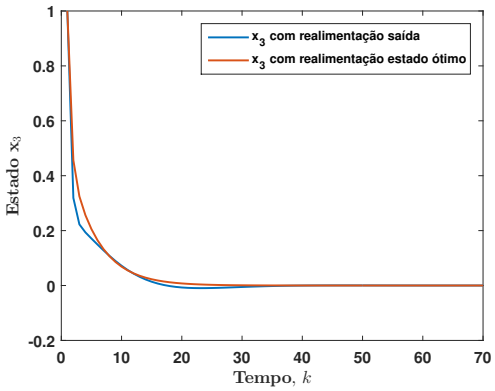
$$Q^{\mathbf{F}}(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}^\top \begin{bmatrix} \mathbf{Q} + \mathbf{A}^\top \mathbf{P}\mathbf{A} & \mathbf{A}^\top \mathbf{P}\mathbf{B} \\ \mathbf{B}^\top \mathbf{P}\mathbf{A} & \mathbf{R} + \mathbf{B}^\top \mathbf{P}\mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} \equiv \mathbf{z}_k^\top \Theta \mathbf{z}_k, \tag{7.35}$$



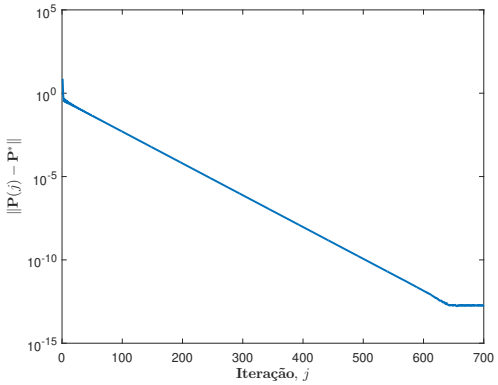
(a) Comportamento do estado $\mathbf{x}_1$



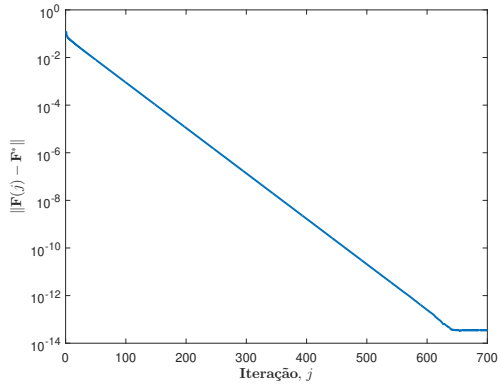
(b) Comportamento do estado $\mathbf{x}_2$



(c) Comportamento do estado $\mathbf{x}_3$



(d) Convergência: $\mathbf{P}(j) \to \mathbf{P}^*$



(e) Convergência: $\mathbf{F}(j) \to \mathbf{F}^*$

Figura 35 – Exemplo 2. (a) Comportamento do estado $\mathbf{x}_1$. (b) Comportamento do estado $\mathbf{x}_2$. (c) Comportamento do estado $\mathbf{x}_3$. (d) Convergência de $\mathbf{P}(j)$ no Algoritmo 13 à matriz $\mathbf{P}^*$ em (7.32). (e) Convergência de $\mathbf{F}(j)$ no Algoritmo 13 à matriz $\mathbf{F}^*$ em (7.32).

com

$$\Theta = \begin{bmatrix} \Theta_{xx} & \Theta_{xu} \\ \Theta_{ux} & \Theta_{uu} \end{bmatrix} = \begin{bmatrix} \mathbf{Q} + \mathbf{A}^\top \mathbf{P}\mathbf{A} & \mathbf{A}^\top \mathbf{P}\mathbf{B} \\ \mathbf{B}^\top \mathbf{P}\mathbf{A} & \mathbf{R} + \mathbf{B}^\top \mathbf{P}\mathbf{B} \end{bmatrix} \text{ e } \mathbf{z}_k = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}. \tag{7.36}$$

Então, usando a notação em (5.15) obtém-se que

$$Q^{\mathbf{F}}(\mathbf{x}_k, \mathbf{u}_k) = \theta^\top \bar{\mathbf{z}}_k, \tag{7.37}$$

no qual θ é uma vetorização da matriz Θ em (7.36). Logo, de (7.37), (7.34) e (7.33), tem-se a equação

$$(\bar{\mathbf{z}}_k - \bar{\mathbf{z}}_{k+1})^\top \theta = \mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k. \tag{7.38}$$

A equação em (7.38) é resolvida em tempo real usando algoritmos de adaptação de parâmetros. O Algoritmo 16 mostra a iteração de política em tempo real para realimentação de saída, sendo $f(\cdot)$ a adaptação do algoritmo NLMS ou RLS.

Algoritmo 16 Iteração de política em tempo real para realimentação de saída

1: Parâmetros $\theta_0(0,0)$ e $\mathbf{F}(0)$ matriz de ganho admissível
2: $k = 0$ (Tempo)
3: **for** $j = 0, 1, \ldots, M$ **do**
4: **for** $i = 1, \ldots, N$ **do**
5: $\mathbf{y}_k = \mathbf{C}\mathbf{x}_k$
6: $\mathbf{u}_k = \mathbf{F}(j)\mathbf{y}_k + \mathbf{v}_k$
7: $\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k$
8: Atualização $\theta_k(j,i) = f\left(\theta_k(j,i-1)\right)$
9: $k = k + 1$
10: **end for**
11: $\Theta_k(j)$ = reconstrução de $\theta_k(j,N)$
12: $\mathbf{F}(j+1) = -(\Theta(j)_{uu})^{-1}\Theta(j)_{ux}\mathbf{C}^\top(\mathbf{C}\mathbf{C}^\top)^{-1}$
13: Inicialização de parâmetros $\theta_k(j+1,0) = \theta_k(j,N)$
14: **end for**

7.3.1 Efeito do ruído de prova

Para resolver a equação de Bellman em (7.38) é adicionado um ruído de prova $\mathbf{v}_k = [v_k^1\ v_k^2\ \cdots\ v_k^m]^T$ na realimentação de saída, de maneira que

$$\tilde{\mathbf{u}}_k = \mathbf{K}\tilde{\mathbf{y}}_k + \mathbf{v}_k, \tag{7.39}$$

no qual

$$\tilde{\mathbf{x}}_{k+1} = \mathbf{A}\tilde{\mathbf{x}}_k + \mathbf{B}\tilde{\mathbf{u}}_k \tag{7.40a}$$

$$\tilde{\mathbf{y}}_k = \mathbf{C}\tilde{\mathbf{x}}_k, \tag{7.40b}$$

com $E[\mathbf{v}_k] = \mathbf{0}$ e $\Gamma_{\mathbf{v}}(k) = \sigma^2\mathbf{I}$, para todo k.

Proposição 7.3.1 *A solução de (7.40) é*

$$\tilde{\mathbf{x}}_k = (\mathbf{A}+\mathbf{BFC})^k \mathbf{x}_0 + \sum_{j=1}^{k} (\mathbf{A}+\mathbf{BFC})^{k-j}\mathbf{B}\mathbf{v}_{j-1}, \tag{7.41}$$

para todo $k \geq 0$.

Demostração. Tem-se que

$$\begin{aligned} \tilde{\mathbf{u}}_k &= \mathbf{F}\tilde{\mathbf{y}}_k + \mathbf{v}_k & (7.42)\\ &= (\mathbf{FC})\tilde{\mathbf{x}}_k + \mathbf{v}_k. & (7.43) \end{aligned}$$

Logo, da Proposição 6.0.1, podemos concluir (7.41).

Teorema 7.3.1 *Considere o sistema em (7.40), onde o ruído de prova* $\mathbf{v}_k$ *satisfaz as condições em (6.4). Então, as seguintes condições são satisfeitas:*

(a) Para todo $k \geq 0$, $E[\tilde{\mathbf{y}}_k] = \mathbf{y}_k$.

(b) Para todo k, $\Gamma_{\tilde{\mathbf{y}}}(k) = \sigma^2 \mathbf{C}\mathbf{S}_k\mathbf{C}^\top$, *onde*

$$\mathbf{S}_k = \sum_{j=1}^{k} \left((\mathbf{A}+\mathbf{BFC})^{k-j}\mathbf{B}\mathbf{B}^\top ((\mathbf{A}+\mathbf{BFC})^{k-j})^\top \right). \tag{7.44}$$

Demostração. (a) Segue-se imediato de (7.41) que

$$\begin{aligned} E[\tilde{\mathbf{y}}_k] &= E[\mathbf{C}\tilde{\mathbf{x}}_k] & (7.45)\\ &= \mathbf{C}\left\{ E[(\mathbf{A}+\mathbf{BFC})^k \mathbf{x}_0] + E\left[\sum_{j=1}^{k} (\mathbf{A}+\mathbf{BFC})^{k-j}\mathbf{B}\mathbf{v}_{j-1} \right] \right\} & (7.46)\\ &= \mathbf{C}\left\{ (\mathbf{A}+\mathbf{BFC})^k \mathbf{x}_0 + \sum_{j=1}^{k} (\mathbf{A}+\mathbf{BFC})^{k-j}\mathbf{B}E[\mathbf{v}_{j-1}] \right\} & (7.47)\\ &= \mathbf{C}\left\{ \mathbf{x}_k + \sum_{j=1}^{k} (\mathbf{A}+\mathbf{BFC})^{k-j}\mathbf{B}\mathbf{0} \right\} & (7.48)\\ &= \mathbf{y}_k. & (7.49) \end{aligned}$$

(b) Para cada $k \geq 1$, seja o vetor

$$\mathbf{s}(j) = (\mathbf{A}+\mathbf{BFC})^{k-j}\mathbf{B}\mathbf{v}_{j-1} \in \mathbb{R}^n, \tag{7.50}$$

com $j = 1, 2, \ldots, k$. Como os vetores $\mathbf{v}_{j-1}$ são mutuamente não correlacionados, então de (7.41) e da notação em (7.50), segue-se que

$$\begin{aligned}
\Gamma_{\tilde{\mathbf{y}}}(k) &= \mathbf{C}\Gamma_{\tilde{\mathbf{x}}}(k)\mathbf{C}^\top && (7.51)\\
&= \mathbf{C}\left(\Gamma_{\sum_{j=1}^{k}\mathbf{s}(j)}(k)\right)\mathbf{C}^\top && (7.52)\\
&= \mathbf{C}\left(\sum_{j=1}^{k}\Gamma_{\mathbf{s}(j)}(k)\right)\mathbf{C}^\top && (7.53)\\
&= \mathbf{C}\left(\sum_{j=1}^{k}(\mathbf{A}+\mathbf{BFC})^{k-j}\mathbf{B}\Gamma_{\mathbf{v}}(j-1)((\mathbf{A}+\mathbf{BFC})^{k-j}\mathbf{B})^\top\right)\mathbf{C}^\top && (7.54)\\
&= \mathbf{C}\left(\sum_{j=1}^{k}(\mathbf{A}+\mathbf{BFC})^{k-j}\mathbf{B}(\sigma^2\mathbf{I})((\mathbf{A}+\mathbf{BFC})^{k-j}\mathbf{B})^\top\right)\mathbf{C}^\top && (7.55)\\
&= \sigma^2\mathbf{C}\left(\sum_{j=1}^{k}\left((\mathbf{A}+\mathbf{BFC})^{k-j}\mathbf{B}\mathbf{B}^\top((\mathbf{A}+\mathbf{BFC})^{k-j})^\top\right)\right)\mathbf{C}^\top && (7.56)\\
&= \sigma^2\mathbf{C}\mathbf{S}_k\mathbf{C}^\top. && (7.57)
\end{aligned}$$

Teorema 7.3.2 *(a) Seja $\mathbf{x}_0 \in \mathbb{R}^n$. Então, para cada matriz admissível $\mathbf{F} \in \mathbb{R}^{m\times p}$, $\tilde{\mathbf{y}}_k$ em (7.40b) satisfaz*

$$\lim_{k\to\infty}\Gamma_{\tilde{\mathbf{y}}}(k) = \lim_{k\to\infty}E[\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k^\top] = \sigma^2\mathbf{C}\mathbf{S}\mathbf{C}^\top, \tag{7.58}$$

sendo $\mathbf{S}$ a única solução da equação de Lyapunov

$$(\mathbf{A}+\mathbf{BFC})\mathbf{S}(\mathbf{A}+\mathbf{BFC})^\top - \mathbf{S} + \mathbf{B}\mathbf{B}^\top = \mathbf{0}. \tag{7.59}$$

(b) Seja $\mathbf{F} \in \mathbb{R}^{m\times p}$ tal que $\|\mathbf{A}+\mathbf{BFC}\| < 1$. Então existe um operador $\mathbf{T}$ em $\mathbb{R}^{n\times n}$ e um único $\mathbf{S}$ tal que $\mathbf{T}(\mathbf{S}) = \mathbf{S}$, sendo $\mathbf{S}$ a única solução de (7.59) e satisfaz (7.58).

Demostração. (a) Como $\mathbf{F}$ é admissível, então a equação de Lyapunov (7.59) tem uma única solução, a série convergente (CHEN, 1999, p. 136)

$$\mathbf{S} = \sum_{l=0}^{\infty}(\mathbf{A}+\mathbf{BFC})^l\mathbf{B}\mathbf{B}^\top((\mathbf{A}+\mathbf{BFC})^l)^\top. \tag{7.60}$$

Logo, do Teorema 7.3.1 (b) e (7.60) segue-se que

$$\lim_{k\to\infty}\Gamma_{\tilde{\mathbf{y}}}(k) = \sigma^2\mathbf{C}\left(\lim_{k\to\infty}\mathbf{S}_k\right)\mathbf{C}^\top = \sigma^2\mathbf{C}\mathbf{S}\mathbf{C}^\top. \tag{7.61}$$

Do Teorema (b) 7.3.1 obtemos

$$\Gamma_{\tilde{\mathbf{y}}}(k) = E[\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k^\top] - \mathbf{y}_k\mathbf{y}_k^\top = \sigma^2\mathbf{C}\mathbf{S}_k\mathbf{C}^\top. \tag{7.62}$$

Como $\mathbf{F}$ é admissível então $\mathbf{y}_k \to \mathbf{0}$ quando $k \to \infty$, e como $\mathbf{S}_k \to \mathbf{S}$ quando $k \to \infty$, então de (7.62) podemos concluir (7.58).
(b) Seja o operador $\mathbf{T}$ em $\mathbb{R}^{n\times n}$

$$\mathbf{T}(\mathbf{X}) = (\mathbf{A}+\mathbf{BFC})\mathbf{X}(\mathbf{A}+\mathbf{BFC})^{\top} + \mathbf{B}\mathbf{B}^{\top}. \tag{7.63}$$

Então, para todo $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times n}$:

$$\|\mathbf{T}(\mathbf{X}) - \mathbf{T}(\mathbf{Y})\| \leq \|\mathbf{A}+\mathbf{BFC}\|^2 \|\mathbf{X}-\mathbf{Y}\|. \tag{7.64}$$

Como $\mathbb{R}^{n\times n}$ é completo e $\|\mathbf{A}+\mathbf{BFC}\|^2 < 1$, então (KREYSZIG, 1978, p. 300) existe um único $\mathbf{S} \in \mathbb{R}^{n\times n}$ tal que $\mathbf{T}(\mathbf{S}) = \mathbf{S}$. Portanto, $\mathbf{S}$ é a única solução de (7.59).

Por outro lado, note que $\mathbf{S}_k$ em (7.44) satisfaz

$$\mathbf{S}_{k+1} = \mathbf{T}(\mathbf{S}_k), \tag{7.65}$$

para todo $k \geq 0$. Então, $\mathbf{S}_k \to \mathbf{S}$ quando $k \to \infty$. Logo, de (7.62) segue-se que $\mathbf{S}$ satisfaz (7.58).

Observação 7.3.1 *Lembre-se que o espaço das matrizes $\mathbb{R}^{n\times n}$ é completo, e em geral todo espaço vetorial normado de dimensão finita é completo (KREYSZIG, 1978, p. 73), no sentido de que toda sequência de Cauchy em $\mathbb{R}^{n\times n}$ é convergente. Uma sequência de Cauchy $\{\mathbf{x}_k\}$ é definida da seguinte forma: para cada $\epsilon > 0$, existe um inteiro positivo k_0 tal que se $k_1, k_2 > k_0$, então*

$$\|\mathbf{x}_{k_1} - \mathbf{x}_{k_2}\| < \epsilon. \tag{7.66}$$

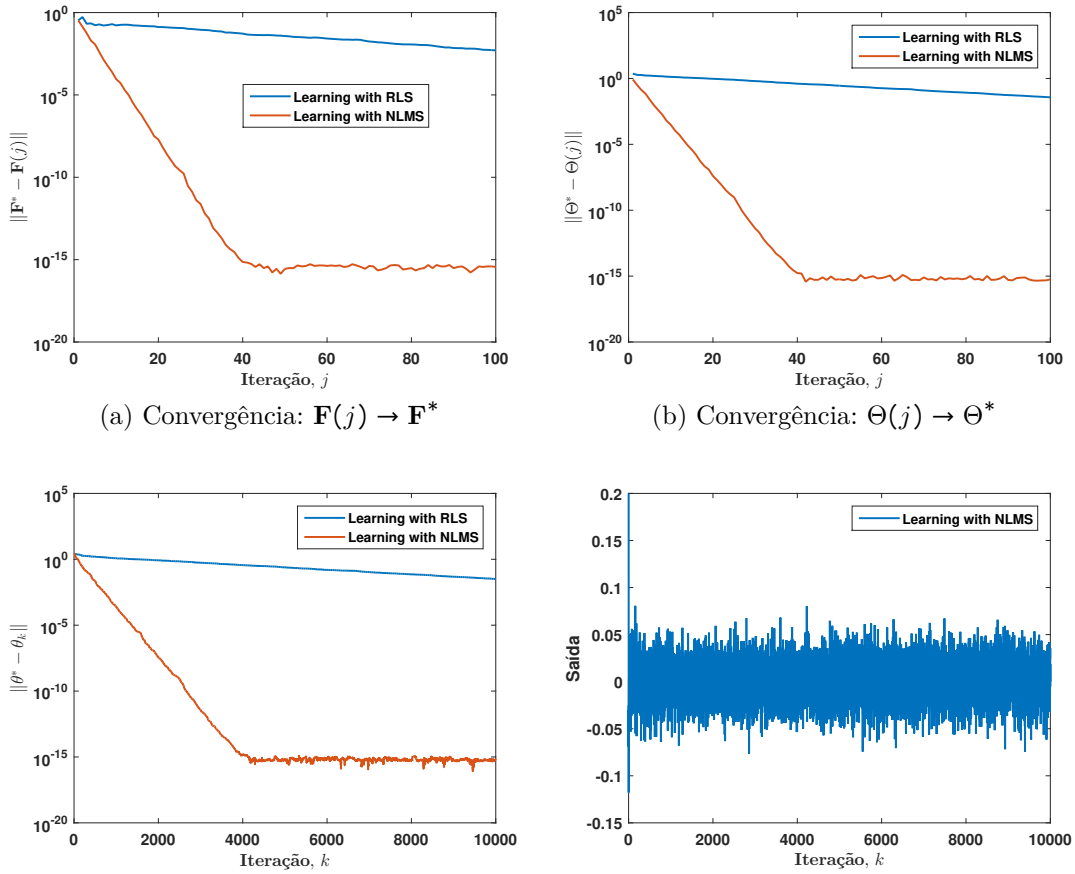
7.4 Simulação computacional

Nesta seção, será usado o algoritmo de iteração de política no Algoritmo 16 para as matrizes em (7.29).

Seja a matriz de ganho inicial $\mathbf{F}(0) = [0,5] \in \mathbb{R}^{1\times 1}$ e variância do ruído de prova $\sigma^2 = (0,02)^2$. A Figura 36 mostra a convergência da matriz de ganho $\mathbf{F}(j)$ (Figura 36 (a)), da matriz da função de ação $\Theta(j)$ (Figura 36 (b)), dos parâmetros $\theta(k)$ (Figura 36 (c)) e da saída do sistema $\mathbf{y}_k$ (Figura 36 (e)). Nesse caso, o algoritmo iteração de política tem um melhor desempenho quando é usado o algoritmo NLMS ao invés do algoritmo RLS.

Para um ruído de prova com variância $\sigma^2 = (0,06)^2$, o algoritmo iteração de política tem um melhor desempenho quando é usado o algoritmo RLS ao invés do algoritmo NLMS (Figura 37).

Igualmente ao caso anterior, para um ruído de prova com variância $\sigma^2 = (0,5)^2$, o algoritmo iteração de política baseado em RLS tem um melhor desempenho quando é comparado com o algoritmo iteração de política baseado em NLMS (Figura 38).



(a) Convergência: $\mathbf{F}(j) \to \mathbf{F}^*$

(b) Convergência: $\Theta(j) \to \Theta^*$

(c) Convergência: $\theta_k \to \theta^*$

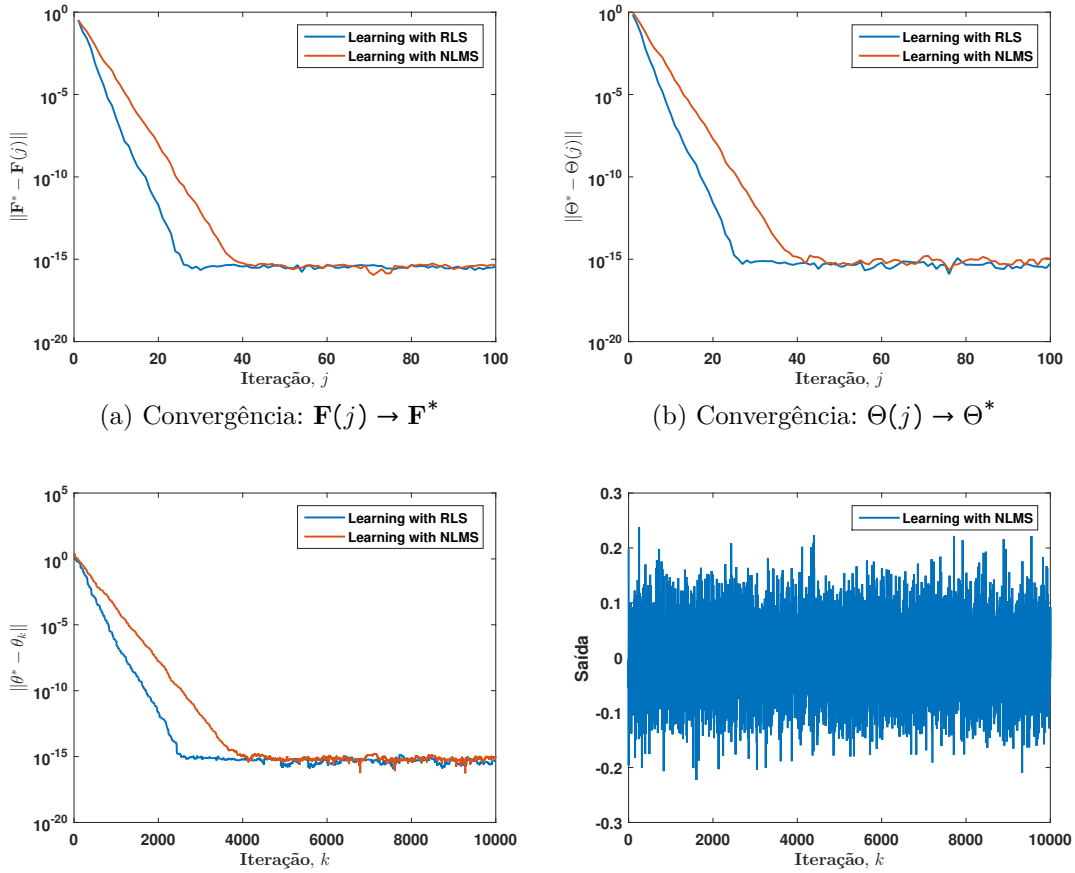
(d) Saída do sistema

Figura 36 – Aprendizagem com variância $\sigma^2 = (0,02)^2$ para o Algoritmo 16. (a) Convergência de $\mathbf{F}(j)$. (b) Convergência de $\Theta(j)$. (c) Convergência de θ_k. (d) Comportamento da saída.

7.5 Considerações finais

Similarmente ao capítulo anterior, neste capítulo, o efeito do ruído de prova para obter a condição de persistência de excitação foi apresentado para o caso do problema do LQR com realimentação de saída, considerando o Algoritmo 16. Já que o Algoritmo 16 é baseado na versão do método de Hewer para realimentação da saída (Algoritmo 13), resta por pesquisar condições necessárias e/ou suficientes de convergência quando a matriz de saída não é quadrada.

Igualmente ao caso da realimentação de estados, o ruído de prova afeta o desempenho numérico dos algoritmos de adaptação NLMS e RLS na aprendizagem da função valor Q. Além disso, em cada tempo de amostragem (e em regime permanente) a norma da matriz de autocovarância das saídas do sistema é proporcional à variância do ruído de prova.



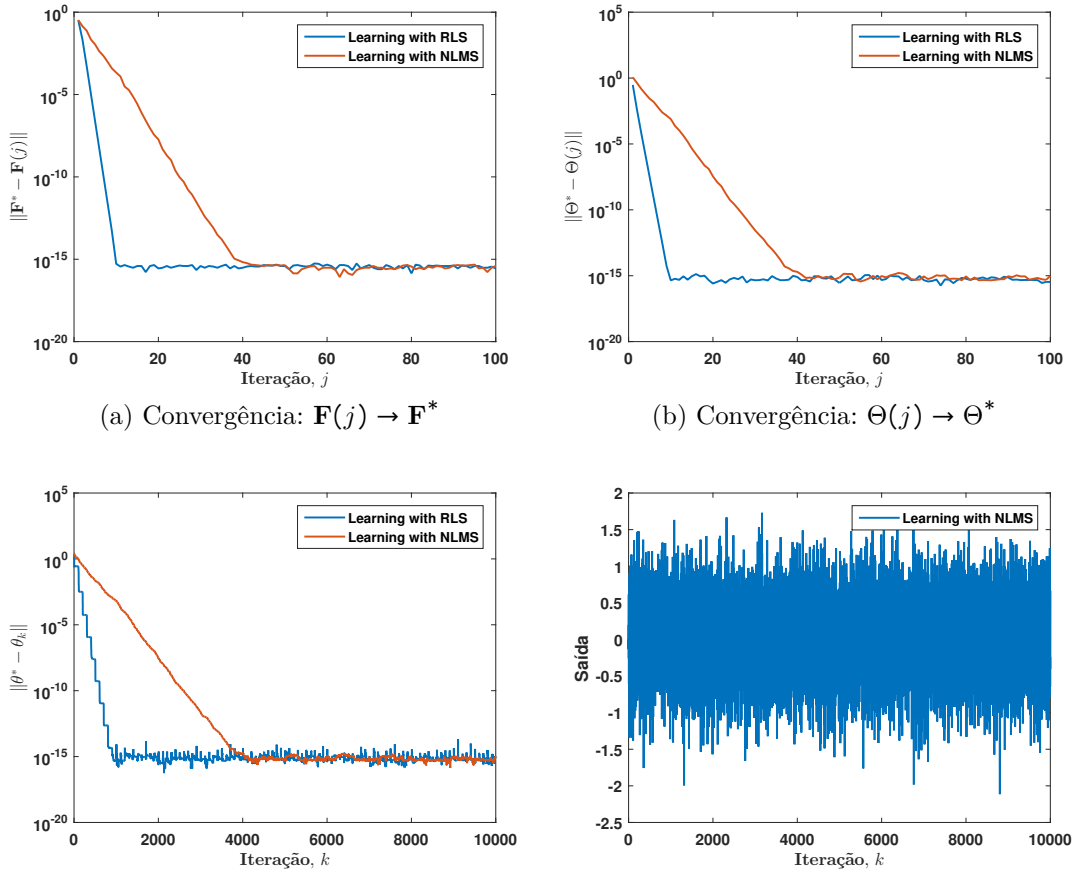
(a) Convergência: $\mathbf{F}(j) \to \mathbf{F}^*$

(b) Convergência: $\Theta(j) \to \Theta^*$

(c) Convergência: $\theta_k \to \theta^*$

(d) Saída do sistema

Figura 37 – Aprendizagem com variância $\sigma^2 = (0,06)^2$ para o Algoritmo 16. (a) Convergência de $\mathbf{F}(j)$. (b) Convergência de $\Theta(j)$. (c) Convergência de θ_k. (d) Comportamento da saída.



(a) Convergência: $\mathbf{F}(j) \to \mathbf{F}^*$

(b) Convergência: $\Theta(j) \to \Theta^*$

(c) Convergência: $\theta_k \to \theta^*$

(d) Saída do sistema

Figura 38 – Aprendizagem com variância $\sigma^2 = (0,5)^2$ para o Algoritmo 16. (a) Convergência de $\mathbf{F}(j)$. (b) Convergência de $\Theta(j)$. (c) Convergência de θ_k. (d) Comportamento da saída.

8 Conclusões

A necessidade da persistência de excitação é um problema clássico em controle ótimo adaptativo usando Q-learning via iteração de política. Como foi mostrado no Teorema 5.2.1, o sinal usado para identificar a função Q de uma política (que é equivalente a resolver uma equação de Bellman) gera uma matriz com colunas linearmente dependentes e como consequência qualquer algoritmo de estimação de parâmetros em tempo real não consegue aprender os parâmetros da função Q (veja Figuras 17 (c)-(d)). Portanto, para obter a condição persistência de excitação, é necessário adicionar um ruído de prova à ação de controle.

Apesar do ruído de prova, o método Q-learning via iteração de política converge para a função Q ótima. No entanto, o ruído de prova afeta duas coisas: (i) a entrada de controle (e, como consequência, afeta a trajetória do sistema durante o processo de aprendizagem) e (ii) o desempenho dos algoritmos de estimação de parâmetros durante o processo de aprendizagem.

Para mostrar o efeito do ruído de prova, foram estabelecidas fórmulas fechadas para as matrizes de autocovariância dos vetores de estado e de saída para cada tempo de amostra. Além disso, em estado estacionário, demonstrou-se a convergência das sequências de matrizes de autocovariância dos vetores de estado e de saída, que dependem de uma solução da equação de Lyapunov. As matrizes de autocovariância em cada tempo de amostra, e em estado estacionário, são proporcionais à variância do ruído de prova.

Experimentos numéricos mostram que o ruído de prova afeta o desempenho dos algoritmos de adaptação no método Q-learning via iteração de política. O algoritmo RLS precisa de uma alta variância de ruído de prova para convergir quando comparado com o algoritmo NLMS; no entanto, nesse caso (RLS), o vetor de estados do sistema fica longe do vetor nulo. Isso é consistente com o Teorema 6.0.1 (b) e o Teorema 6.0.2, já que a norma da matriz de autocovariância do vetor de estados é proporcional à variância do ruído de prova. Portanto, se o objetivo é manter o vetor de estados perto do vetor nulo, ruído de prova com baixa variância no algoritmo Q-learning baseado em NLMS é a melhor opção, já que para baixa variância o clássico algoritmo Q-learning baseado em RLS diverge ou converge para os parâmetros ótimos muito lentamente.

Por outro lado, baseado na Proposição 7.1.2, o Algoritmo dada na Tabela 15 (ou equivalentemente o Algoritmo 16), converge para uma matriz de ganho $\mathbf{F}$ quando a matriz de saída $\mathbf{C}$ é quadrada. No entanto, com foi mostrado na simulação computacional, esses algoritmos convergem (empiricamente) quando a matriz $\mathbf{C}$ não é quadrada.

Os resultados obtidos para realimentação de estados são estendidos para realimen-

tação de saída, no sentido de que para baixas variâncias, o algoritmo Q-learning baseado em RLS não consegue aprender os parâmetros ótimos ou pode ter uma convergência lenta.

Como trabalho futuro pode ser estudado limites da variância do ruído de prova que garantem convergência dos algoritmos de adaptação de parâmetros NLMS e RLS na aprendizagem da função valor de ação. Além do mais, pode ser pesquisado o efeito com diferentes tipos de ruído de prova, tais como soma de sinais sinusoidais com diferentes frequências ou ruídos exponencialmente decrescentes. Em relação à realimentação de saída, estabelecer condições necessárias e/ou suficientes de convergência quando a matriz de saída não é quadrada.

Referências

AL-TAMIMI, A.; ABU-KHALAF, M.; LEWIS, F. L. Adaptive critic designs for discrete-time zero-sum games with application to H_∞ control. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, IEEE, v. 37, n. 1, p. 240–247, 2007. Citado na página 71.

AL-TAMIMI, A.; LEWIS, F. L.; ABU-KHALAF, M. Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control. *Automatica*, Elsevier, v. 43, n. 3, p. 473–481, 2007. Citado 3 vezes nas páginas 17, 18 e 75.

ÅSTRÖM, K. J.; WITTENMARK, B. *Adaptive control*. Second edition. [S.l.]: Dover Publications, 2008. Citado na página 71.

BELLMAN, R. E. *Dynamic Programming*. [S.l.]: NJ: Princenton Univ. Press, 1957. Citado 3 vezes nas páginas 14, 22 e 58.

BITMEAD, R. Persistence of excitation conditions and the convergence of adaptive schemes. *IEEE Trans. Inform. Theory*, IEEE, v. 30, n. 2, p. 183–191, 1984. Citado 2 vezes nas páginas 17 e 68.

BLONDEL, V. D.; TSITSIKLIS, J. N. A survey of computational complexity results in systems and control. *Automatica*, Elsevier, v. 36, n. 9, p. 1249–1274, 2000. Citado na página 105.

BRADTKE, S. J.; YDSTIE, B. E.; BARTO, A. G. Adaptive linear quadratic control using policy iterartion. In: *Proc. of the American Control Conference*. Baltimore, Maryland: [s.n.], 1994. p. 3475–3479. Citado 3 vezes nas páginas 17, 65 e 75.

BRADTKE, S. J.; YDSTIE, B. E.; BARTO, A. G. *Adaptive linear quadratic control using policy iteration*. [S.l.], 1994. Citado 2 vezes nas páginas 17 e 18.

BRYSON, A. E.; HO, Y.-C. *Applied optimal control: optimization, estimation and control*. [S.l.]: Taylor & Francis Group, 1975. Citado na página 44.

BUSONIU, L. et al. *Reinforcement learning and dynamic programming using function approximators*. [S.l.]: Taylor & Francis Group, 2010. Citado 6 vezes nas páginas 15, 22, 23, 24, 25 e 33.

CAO, Y.-Y.; LAM, J.; SUN, Y.-X. Static output feedback stabilization: an ilmi approach. *Automatica*, Elsevier, v. 34, n. 12, p. 1641–1645, 1998. Citado na página 105.

CHAKRABARTY, A. et al. Appoximate dynamic programming for linear systems with state and input constraints. In: *European Control Conference*. [S.l.: s.n.], 2019. p. 1–17. Citado na página 75.

CHEN, C.-T. *Linear System Theory and Desing*. Third edition. [S.l.]: Oxford University Press, 1999. Citado 6 vezes nas páginas 42, 50, 55, 80, 83 e 114.

CHOI, S.; SIRISENA, H. Computation of optimal output feedback controls for unstable linear multivariable systems. *IEEE Transactions on Automatic Control*, IEEE, v. 22, n. 1, p. 134–136, 1977. Citado na página 105.

CRUSIUS, C. A.; TROFINO, A. Sufficient LMI conditions for output feedback control problems. *IEEE Transactions on Automatic Control*, IEEE, v. 44, n. 5, p. 1053–1057, 1999. Citado na página 105.

DAS, S. et al. Fitted Q-learning for relational domains. *arXiv:2006.05595v1 [cs.LG]*, 2020. Citado na página 17.

DEMIR, O.; ÖZBAY, H. Static output feedback stabilization of discrete time linear time invariant systems based on approximate dynamic programming. *Transactions of the Institute of Measurement and Control*, SAGE Publications Sage UK: London, England, v. 42, n. 16, p. 3168–3182, 2020. Citado na página 108.

FARJADNASAB, M.; BABAZADEH, M. Model-free LQR design by Q-function learning. *Automatica*, Elsevier, v. 137, p. 110060, 2022. Citado na página 65.

GARCIA, G.; PRADIN, B.; ZENG, F. Stabilization of discrete time linear systems by static output feedback. *IEEE Transactions on Automatic Control*, IEEE, v. 46, n. 12, p. 1954–1958, 2001. Citado na página 105.

GLOVERA, F.; LAGUNA, M. *Tabu Search*. [S.l.]: Kluwer, 1977. Citado na página 37.

GOLDBER, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. [S.l.]: Addison-Wesley, 1989. Citado na página 37.

GOODWIN, G. C.; SIN, K. S. *Adaptive filtering prediction and control*. [S.l.]: Dover Publications, 2009. Citado 2 vezes nas páginas 17 e 68.

HAGEN, S. ten; KRÖSE, B. Linear Quadratic Regulation using Reinforcement learning. In: *In: Proc. BENELEARN-98, 8th Belgian-Dutch Conference on Machine Learning*. [S.l.: s.n.], 1998. p. 39–46. Citado 4 vezes nas páginas 17, 18, 75 e 79.

HAYKIN, S. Adaptive filter theory (book). *Englewood Cliffs, NJ, Prentice-Hall, 1986. 607*, 1986. Citado na página 87.

HAYKIN, S. *Neural Networks and Learning Machines*. Third edition. [S.l.]: Prentice Hall, 2009. Citado na página 15.

HEWER, G. An iterative technique for the computation of steady state gains for the discrete optimal regulator. *IEEE Transactions on Automatic Control*, v. 16, n. 4, p. 382–384, 1971. Citado 3 vezes nas páginas 53, 89 e 108.

HUANG, Z.; BALAKRISHNAN, S. Robust adaptive critic based neurocontrollers for systems with input uncertainties. In: IEEE. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*. [S.l.], 2000. v. 3, p. 67–72. Citado na página 14.

HUANG, Z.; BALAKRISHNAN, S. Robust adaptive critic based neurocontrollers for missiles with model uncertainties. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. [S.l.: s.n.], 2001. Citado na página 14.

IWASAKI, T.; SKELTON, R.; GEROMEL, J. Linear quadratic suboptimal control with static output feedback. *Systems & Control Letters*, Elsevier, v. 23, n. 6, p. 421–430, 1994. Citado na página 105.

JIANG, Y.; JIANG, Z.-P. Robust approximate dynamic programming and global stabilization with nonlinear dynamic uncertainties. In: IEEE. *2011 50th IEEE Conference on Decision and Control and European Control Conference*. [S.l.], 2011. p. 115–120. Citado na página 18.

JIANG, Y.; JIANG, Z.-P. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica*, Elsevier, v. 48, n. 10, p. 2699–2704, 2012. Citado na página 18.

JIANG, Y.; JIANG, Z.-P. *Robust Adaptive Dynamic Programming*. [S.l.]: IEE Press, Wiley, 2017. Citado 2 vezes nas páginas 17 e 18.

KHAN, S. G. et al. Reinforcement learning and optimal adaptive control: An overview and implementation examples. *Annual reviews in control*, Elsevier, v. 36, n. 1, p. 42–59, 2012. Citado 2 vezes nas páginas 14 e 15.

KIM, J.-H.; LEWIS, F. L. Model-free H_∞ control desing for unknown linear discrete-time systems via Q-learning with LMI. *Automatica*, v. 46, p. 1320–1326, 2010. Citado na página 75.

KIUMARSI, B.; LEWIS, F.; JIANG, Z.-P. H_∞ control of linear discrete-time systems: Off-policy reinforcement learning. *Automatica*, v. 78, p. 144–152, 2017. Citado 2 vezes nas páginas 16 e 18.

KIUMARSI, B. et al. Optimal and Autonomous Control Using Reinforcement Learning: A Survey. *IEEE Trans. on Neural Networks and Learning Systems*, v. 29, n. 6, p. 2042–2062, 2018. Citado na página 14.

KREYSZIG, E. *Introductory Functional Analysis with Aplicatins*. [S.l.]: Jonh Wiley & Sons, 1978. Citado 2 vezes nas páginas 31 e 115.

KRISHNAKUMAR, K.; NEIDHOEFER, J. Immunized adaptive critics. In: IEEE. *Proceedings of International Conference on Neural Networks (ICNN'97)*. [S.l.], 1997. v. 4, p. 2283–2287. Citado na página 14.

KUČERA, V.; SOUZA, C. E. D. A necessary and sufficient condition for output feedback stabilizability. *Automatica*, Elsevier, v. 31, n. 9, p. 1357–1359, 1995. Citado na página 105.

LAUB, A. J. A schur method for solving algebraic riccati equations. *IEEE Transactions on Automatic Control*, v. 24, n. 6, p. 913–921, 1979. Citado na página 53.

LENDARIS, G. G. A retrospective on adaptive dynamic programming for control. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS. Atlanta, GA, USA, 2009. p. 1750–1757. Citado na página 14.

LENDARIS, G. G. et al. Dual heuristic programming for fuzzy control. In: IEEE. *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569)*. [S.l.], 2001. v. 1, p. 551–556. Citado na página 14.

LEVINE, W.; ATHANS, M. On the determination of the optimal constant output feedback gains for linear multivariable systems. *IEEE Transactions on Automatic control*, IEEE, v. 15, n. 1, p. 44–48, 1970. Citado na página 105.

LEWIS, F.; VRABIE, D. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits and System Magazine*, v. 9, n. 3, p. 32–50, 2009. Citado 6 vezes nas páginas 14, 15, 17, 26, 53 e 86.

LEWIS, F. L.; VAMVOUDAKIS, K. G. Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, IEEE, v. 41, n. 1, p. 14–25, 2011. Citado na página 105.

LEWIS, F. L.; VRABIE, D.; SYRMOS, V. *Optimal Control*. [S.l.]: John Wiley & Sons, Inc., 2012. Citado 10 vezes nas páginas 8, 14, 40, 42, 43, 46, 49, 50, 51 e 59.

LEWIS, R. M.; TORCZON, V. Pattern search algorithms for linearly constrained minimization. *SIAM Journal on Optimization*, v. 10, n. 3, p. 917–941, 2000. Citado na página 37.

LI, Y. et al. Data-driven approximate Q-learning stabilization with optimality error bound analysis. *Automatica*, Elsevier, v. 103, p. 435–442, 2019. Citado na página 17.

LIU, C.; MURPHEY, Y. L. Optimal power management based on Q-learning and neuro-dynamic programming for plug-in hybrid electric vehicles. *IEEE Trans. Neural Netw. Learn. Syst.*, IEEE, v. 31 (6), p. 1942–1954, 2020. Citado na página 17.

LIU, X.; BALAKRISHNAN, S. Convergence analysis of adaptive critic based neural networks. In: *Proceedings of 2000 American Control Conference*. [S.l.: s.n.], 2000. Citado na página 14.

LOPEZ, V. G.; ALSALTI, M.; MÜLLER, M. A. Efficient off-policy Q-learning for data-based discrete-time LQR problems. *arXiv preprint arXiv:2105.07761*, 2021. Citado na página 65.

LOW, E. S.; ONG, P.; CHEAH, K. C. Solving the optimal path planning of a mobile robot using improved Q-learning. *Robot. Auton. Syst.*, Elsevier, v. 115, p. 143–161, 2019. Citado na página 17.

LUUS, R. *Iterative dynamic programming*. [S.l.]: CRC Press, 2019. Citado na página 64.

MANNOR, S.; RUBINSTEIN, R.; GAT, Y. The cross entropy method for fast policy search. In: *In: Proc. 20th International Conference on Machine Learning (ICML 03)*. [S.l.: s.n.], 2003. p. 512–519. Citado na página 37.

MU, C. et al. Q-learning solution for optimal consensus control of discrete-time multiagent systems using reinforcement learning. *J. Franklin Inst.*, Elsevier, v. 356, p. 6946–6967, 2019. Citado na página 17.

MURRAY, J. J. et al. Adaptive dynamic programming. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, IEEE, v. 32, n. 2, p. 140–153, 2002. Citado na página 71.

NETO, J. V. da F.; RêGO, P. H. QR-Tuning and approximate-LS solutions of the HJB equation for online DLQR desing via state and action-dependent heuristic dynamic programming. *International Journal of Innovative Computing, Information and Control*, v. 10, n. 3, p. 1071–1094, 2014. Citado na página 54.

PADHI, R.; BALAKISHNAN, S. A systematic synthesis of optimal process control with neural networks. In: IEEE. *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)*. [S.l.], 2001. v. 3, p. 1910–1915. Citado na página 14.

PADHI, R.; BALAKRISHNAN, S. Adaptive critic based optimal control for distributed parameter systems. In: *Proc. Intnl. Conference on Information, Communication and Signal Processing*. [S.l.: s.n.], 1999. Citado na página 14.

PANG, B.; BIAN, T.; JIANG, Z.-P. Adaptive dynamic programming for finite-horizon optimal control of linear time-varying discrete system. *Control Theory and Technology*, v. 17, n. 1, p. 73–84, 2019. Citado na página 46.

PONTRYAGIN, L. S. et al. *Mathematical theory of optimal processes*. [S.l.]: Gordon and Breach Science Publishers, 1986. Citado na página 14.

PUTERMAN, M. L. *Markov Decision Processes-Discrete Stochastic Dynamic Programming*. [S.l.]: Wiley, 1994. Citado 2 vezes nas páginas 8 e 22.

RIZVI, A. A. A.; LIN, Z. Output Feedback Q-learning Control for the Discrete-Time Linear Quadratic Regulator Problem. *IEEE Transactions on Neural Networks and Learning Systems*, v. 30, n. 5, p. 1523–1536, 2019. Citado 3 vezes nas páginas 18, 105 e 108.

ROSINOVÁ, D.; VESELỲ, V.; KUČERA, V. A necessary and sufficient condition for static output feedback stabilizability of linear discrete-time systems. *Kybernetika*, Institute of Information Theory and Automation AS CR, v. 39, n. 4, p. 447–459, 2003. Citado 3 vezes nas páginas 105, 107 e 108.

RUBINSTEIN, R. Y.; KROESE, D. P. *The Cross Entropy Method: A unified approach to combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*. [S.l.]: Springer, 1989. Citado na página 37.

RUMMERY, G. A.; NIRANJAN, M. *On-line Q-learning using connectionist systems*. [S.l.], 1994. Citado 2 vezes nas páginas 16 e 65.

SAADATMAND, S. et al. Autonomous control of a line follower robot using a Q-learning controller. In: IEEE. *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*. [S.l.], 2020. p. 0556–0561. Citado na página 17.

SADABADI, M. S.; PEAUCELLE, D. From static output feedback to structured robust static output feedback: A survey. *Annual reviews in control*, Elsevier, v. 42, p. 11–26, 2016. Citado na página 105.

SAHOO, P. P.; VAMVOUDAKIS, K. G. On-Off Adversarially Robust Q-learning. *IEEE Control Syst. Lett.*, IEEE, v. 4, n. 3, p. 749–754, 2020. Citado na página 17.

SILVA, M. E. G. *Algoritmos da Família LMS para a Solução Aproximada da HJB em Projetos Online de Controle Ótimo Discreto Multivariável e Aprendizado por Reforço*. Dissertação (Mestrado) — Universidade Federal do Maranhão, 2014. Citado 2 vezes nas páginas 17 e 18.

SUSSMANN, H. J.; WILLEMS, J. C. 300 years of optimal control: from the brachystochrone to the maximum principle. *IEEE Control Systems Magazine*, IEEE, v. 17, n. 3, p. 32–44, 1997. Citado na página 14.

SUTTON, R. S. *Temporal credit assignament in reinforcement learning*. Tese (Doutorado) — University of Massachusetts, 1984. Citado 2 vezes nas páginas 14 e 16.

SUTTON, R. S. Learning to predict by the methods of temporal differences. *Machine learning*, Springer, v. 3, n. 1, p. 9–44, 1988. Citado 2 vezes nas páginas 14 e 16.

SUTTON, R. S.; BARTO, A. G. *Reinforcement Learning, An Introduction*. Second edition. [S.l.]: Cambridge, MA: MIT Press, 2018. Citado 4 vezes nas páginas 14, 15, 16 e 23.

SYRMOS, V. L. et al. Static output feedback—a survey. *Automatica*, Elsevier, v. 33, n. 2, p. 125–137, 1997. Citado na página 105.

TORCZON, V. On the convergence of pattern search algorithm. *SIAM Journal on Optimization*, v. 7, n. 1, p. 1–25, 1997. Citado na página 37.

TREFETHEN, L. N.; III, D. B. *Numerical linear algebra*. [S.l.]: SIAM, 1997. v. 50. Citado na página 56.

TROFINO-NETO, A.; KUCERA, V. Stabilization via static output feedback. *IEEE Transactions on Automatic Control*, IEEE, v. 38, n. 5, p. 764–765, 1993. Citado na página 105.

VALADBEIGI, A. P.; SEDIGH, A. K.; LEWIS, F. L. H_∞ static output-feedback control design for discrete-time systems using reinforcement learning. *IEEE transactions on neural networks and learning systems*, IEEE, v. 31, n. 2, p. 396–406, 2019. Citado na página 105.

VAMVOUDAKIS, K.; VRABIE, D.; LEWIS, F. Online policy iteration based algorithms to solve the continuous-time infinite horizon optimal control problem. In: IEEE. *2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*. [S.l.], 2009. p. 36–41. Citado na página 18.

VAMVOUDAKIS, K. G. Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach. *Systems Control Lett.*, Elsevier, v. 100, p. 14–20, 2017. Citado na página 18.

VAMVOUDAKIS, K. G.; LEWIS, F. L. Multi-player non-zero-sum games: Online adaptive learning solution of coupled Hamilton–Jacobi equations. *Automatica*, Elsevier, v. 47, n. 8, p. 1556–1569, 2011. Citado na página 18.

VESELỲ, V. Static output feedback controller design. *Kybernetika*, Institute of Information Theory and Automation AS CR, v. 37, n. 2, p. 205–221, 2001. Citado na página 105.

WANG, Y. et al. Target transfer Q-learning and its convergence analysis. *Neurocomputing*, Elsevier, v. 392, p. 11–22, 2020. Citado na página 17.

WATKINS, C. *Learning from delayed rewards*. Tese (Doutorado) — Cambridge University, 1989. Citado 2 vezes nas páginas 16 e 65.

WATKINS, C. J. C. H.; DAYAN, P. Q-learning. *Machine Learning*, v. 8, p. 279–292, 1992. Citado 2 vezes nas páginas 16 e 65.

WERBOS, P. J. *Beyond Regression: New tools for prediction and analysis in the behavior sciences*. Tese (Doutorado) — Harvard University, 1974. Citado 2 vezes nas páginas 14 e 15.

WERBOS, P. J. Neural networks for control and system identification. In: *Proc. IEEE Conf. Decision and Control*. Tampa, FL: [s.n.], 1989. p. 260–265. Citado 2 vezes nas páginas 14 e 15.

WERBOS, P. J. A menu of designs for reinforcement learning over time. In: _____. *Neural networks for control*. Eds. Cambridge, MA: MIT Press, 1991. p. 67–95. Citado 2 vezes nas páginas 14 e 15.

WERBOS, P. J. Approximate dynamic programming for real-time control and neural modeling. In: _____. *Handbook of Intelligent Control*. Eds. New York: Van Nostrand Reinhold, 1992. Citado 2 vezes nas páginas 14 e 15.

YAGHMAIE, F. A.; GUSTAFSSON, F.; LJUNG, L. Linear quadratic control using model-free reinforcement learning. *IEEE Transactions on Automatic Control*, IEEE, 2022. Citado na página 65.

YÁNEZ, W. J. L.; SOUZA, F. d. C. de. Considerações sobre o controle ótimo LQR online com solução da equação algébrica de Riccati baseada em algoritmos de filtragem adaptativa. In: *Simpósio Brasileiro de Automação Inteligente-SBAI*. [S.l.: s.n.], 2021. v. 1, n. 1. Citado na página 69.

YÁNEZ, W. J. L.; SOUZA, F. d. C. de. On the effect of probing noise in optimal control LQR via Q-learning using adaptive filtering algorithms. *European Journal of Control*, Elsevier, p. 100633, 2022. Citado 3 vezes nas páginas 18, 81 e 82.

ZHAO, Y.; LEE, J.; CHEN, W. Q-greedy UCB: a new exploration policy for adaptive and resource-efficient scheduling. *arXiv:2006.05902v1 [eess.SY]*, 2020. Citado na página 17.