

Universidade Federal do Maranhão
Centro de Ciências Exatas e Tecnologia
Programa de Pós Graduação em Engenharia de Eletricidade

SOLUÇÃO GERAL DA EQUAÇÃO ALGÉBRICA DE RICCATI
DISCRETA UTILIZANDO ESTIMADOR NÃO QUADRÁTICO E
DECOMPOSIÇÃO MATRICIAL APLICADO AO CONTROLE ÓTIMO
DE UM GERADOR EÓLICO

Jonathan Araujo Queiroz

São Luis
2016

Universidade Federal do Maranhão
Centro de Ciências Exatas e Tecnologia
Programa de Pós Graduação em Engenharia de Eletricidade

Jonathan Araujo Queiroz

SOLUÇÃO GERAL DA EQUAÇÃO ALGÉBRICA DE RICCATI
DISCRETA UTILIZANDO ESTIMADOR NÃO QUADRÁTICO E
DECOMPOSIÇÃO MATRICIAL APLICADO AO CONTROLE ÓTIMO
DE UM GERADOR EÓLICO

Dissertação apresentada ao Programa de Pós Graduação em Engenharia de Eletricidade da UFMA como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia de Eletricidade. Área de concentração: Automação e Controle - Linhas de Pesquisa: Processamento Analógico e Digital de Sinais

Orientador: Ph.D. Allan Kardec Duailibe Barros Filho

Coorientador: D.r João Viana da Fonseca Neto

São Luis
2016

Insira nesta página a sua ficha catalográfica (somente versão final). Obs. É conveniente converter o documento fornecido pela BAE (normalmente .doc) em um arquivo .ps. Para a versão preliminar da tese (antes da defesa), simplesmente remova essa página.

SOLUÇÃO GERAL DA EQUAÇÃO ALGÉBRICA DE RICCATI
DISCRETA UTILIZANDO ESTIMADOR NÃO QUADRÁTICO E
DECOMPOSIÇÃO MATRICIAL APLICADO AO CONTROLE ÓTIMO
DE UM GERADOR EÓLICO

Jonathan Araujo Queiroz

Dissertação aprovada em 08 de março de 2016

Prof. Allan Kardec Duailibe Barros Filho, Ph.D.
(Orientador)

Prof. João Viana da Fonseca Neto, Dr.
(Coorientador)

Profa. Laurinda Lúcia Nogueira dos Reis, Dra.
(Membro da banca Examinadora)

Profa. Ginalber Luis de Oliveira Serra, Dr.
(Membro da banca Examinadora)

Dedicatória

Dedico,

Em primeiro lugar dedico à Deus minha fortaleza, meu refúgio e proteção. A Ele que em seu imenso amor nos concedeu o dom de desvendar os segredos da vida.
Muito obrigado!!!

A Thaise de Cassia Muniz Queiroz minha auxiliadora, minha amada, minha esposa. A minha filha Maria Luísa a quem me motiva e inspira. A minha mãe Maria Antônia Araujo, exemplo de força e perseverança, ao meu Pai Francisco de Assis Queiroz, exemplo de senso crítico e busca do conhecimento, a minha irmã Tatiane por suas palavras de incentivo, a minha irmã Sarah por sempre estar disposta a me ajudar, a minha irmã caçula Raissa por toda a sua preocupação e amor a mim dedicado, e por fim minha sobrinha Geovana por sua doçura a toda minha família por fazerem parte desta conquista!.

Agradecimentos

Agradeço,

Ao professor Ph.D. Allan Kardec Duailibe Barros Filho, por tudo que pacientemente me ensinou, pelas riquíssimas contribuições a minha formação acadêmica. Gostaria de lhe agradecer também por dividir comigo e com meus colegas de laboratório um pouquinho do seu amplo conhecimento. Mais do que um agradecimento deixo minha admiração.

Aos professores João Viana da Fonseca Neto e Ewaldo Eder Carvalho Santana pela amizade, incentivo, motivação e dedicação, fundamentais para o desenvolvimento deste trabalho de pesquisa.

À PPGEE/UFMA e ao CNPq pela estrutura que oferece aos estudantes e pesquisadores.

E aqueles que, direta ou indiretamente, contribuíram para a elaboração deste trabalho.

A todos meu sincero, Obrigado!

Quanto + estudo, - acredito em acasos.

Jonathan Araujo Queiroz

Resumo

A equação algébrica Riccati discreta (*Discrete Algebraic Riccati Equation (DARE)*) tem desempenhado um papel cada vez mais importante na teoria de controle ótimo e de filtragem adaptativa. Por esse motivo, varias técnicas tem sido desenvolvidas para solucionar a DARE, por exemplo a abordagem baseada em auto vetores ou ainda abordagens relacionadas a subespaços invariantes, as quais requerem rigor e precisão matemáticas. No entanto, estas abordagens apresentam uma série de problemas, dentre eles, o fato de não poderem ser implementadas em tempo real devido ao seu custo computacional para estimar a solução da DARE em diversos sistemas, sobretudo sistemas com ordem superior a três. Com o intuito de resolver este problema, propomos solucionar a DARE utilizando um estimador baseado na soma das potencias pares do erro. O estimador é similar ao *Recursive Least Squares (RLS)*, mas com um desempenho melhor em termos de velocidade de convergência e precisão de estimativa, sem aumento significativo da complexidade computacional. O estimador é denominado (*Recursive Least Non-Squares (RLNS)*). Um outro aspecto para que possamos solucionar a DARE de forma geral, é garantir que a DARE seja numericamente bem condicionada. Para efetuar o condicionamento numérico da DARE, será utilizada uma técnica de decomposição matricial conhecida como inversa de Moore-Penrose ou inversa generalizada. A metodologia proposta é avaliada em um sistema multivariável de 6 ordem correspondente ao gerador eólico. O método é validado sob o ponto de vista de estabilidade numérica e velocidade de convergência.

Palavras-chave: Equação Algébrica de Riccati Discreta, Estimador não Quadrático, Decomposição Matricial, Estabilidade Numérica, Velocidade de Convergência Gerador Eólico.

Abstract

The discrete Riccati algebraic equation has played an increasingly important role in optimal control theory and adaptive filtering. For this reason, various techniques have been developed to solve the DARE, for example the approach based on self vectors or approaches related to invariant subspaces [1], which require mathematical rigor and precision. However, these approaches present a number of problems, among them the fact that they can not be implemented in real-time due to its high computational cost to estimate the solution of DARE in many systems, especially systems with higher order three. In order to overcome this problem, we propose to solve the DARE using as an estimator based on the sum of potential error pairs. The estimator is similar to the Recursive Least Squares (RLS), but with a better performance in terms of convergence speed and estimation accuracy without a significant increase in computational complexity. The estimator is called Recursive Least Non-Squares (RLNS). One other aspect in unraveling the general DARE is to ensure that DARE is numerically well conditioned. To perform the numerical conditioning of DARE, a matrix decomposition technique known as Moore-Penrose inverse or generalized inverse is used. The proposed method is evaluated in a multivariate system 6^{th} order corresponding to the wind generator. The method is evaluated under the numerical stability point of view and speed of convergence.

Discrete Algebraic Riccati Equation, Non-squares Approximators, Matrix decomposition, Numerical Stability, Convergence Speed, Wind Generator.

Lista de Figuras

3.1	Combinador linear adaptativo com a resposta desejado e o erro.	11
4.1	Estimação dos elementos da matriz solução P utilizando o RLNS e o RLS. A estrutura da estimação é da pela combinação das entradas p_n com os pesos w_n produzindo o valor estimado \hat{P} . Posteriormente, o erro é calculado pela diferença entre o valor estimado \hat{P} e o valor de referencia P_{Shur} , objetivando um erro mínimo. Por isso, o processo é adaptado pelos estimadores RLNS e RLS até a enésima iteração.	24
4.2	Os estimadores RLNS e RLS são utilizados para aproximar os parâmetros da diagonal principal da matriz P , para metodologia proposta e para metodologia clássica. Parâmetros são apresentados em função do erro, o valor de estado estacionário aqui assumido e inferior a 1%.	29
4.3	Os estimadores RLNS e RLS são utilizados para aproximar os parâmetros fora da diagonal principal da matriz P , para metodologia proposta e para metodologia clássica. Parâmetros são apresentados em função do erro, o valor de estado estacionário aqui assumido e inferior a 5%.	31

Lista de Tabelas

3.1	Decomposições matriciais.	18
4.1	Parâmetros do gerador eólico	22
4.2	Ajuste do fator de esquecimento (λ)	26
4.3	Número de iterações para o RLNS e RLS atingir o valor de estado estacionário com um erro inferior a 1%, para metodologia clássica e para metodologia proposta.	28
4.4	Número de iterações para o RLNS e RLS atingir o valor de estado estacionário com um erro inferior a 5%, para metodologia clássica e para metodologia proposta.	30

Sumário

1	Introdução	1
1.1	Motivações	2
1.2	Objetivo	3
1.2.1	Objetivo Geral	3
1.2.2	Objetivo Especifico	3
1.3	Organização do Texto	3
2	Equação Algébrica de Riccati Discreta	4
2.1	Introdução	4
2.2	DARE e o Controle Ótimo	5
2.2.1	Modelo em Espaço de Estado de um Gerador Eólico	5
2.3	Considerações Finais	8
3	Solução da Equação Algébrica de Riccati Discreta	9
3.1	Introdução	9
3.2	Primeira Etapa: Estimacão da matriz soluçao P para a DARE	10
3.2.1	Soluções Clássicas	10
3.2.2	Soluções Estimadas	11
3.3	Segunda Etapa: Condicionamento numérico para DARE	18
3.4	Considerações Finais	21
4	Solução da Equação Algébrica de Riccati Discreta Utilizando o Modelo em Espaço de Estado do Gerador Eólico	22
4.1	Simulações	22
4.2	Resultados	28
4.3	Considerações Finais	33
5	Conclusão e Perspectivas	34
A	Publicações	36

B	Decomposições	38
B.1	Decomposição em valores e vetores singulares (SVD)	38
B.2	Decomposição de Schur	39

Introdução

Durante muitos séculos, o grande desafio para a indústria foi automatizar suas linhas de produção. A diferença entre as linhas de produção manufaturadas e as linhas automatizadas, aconteceu em 1760 com a revolução industrial. Nos séculos seguintes foram crescentes os estudos sobre como deveria ser efetuado o controle desse maquinário. Esses estudos foram os primeiros passos da teoria de controle. Com o desenvolvimento da teoria de controle varias vertentes foram se consolidando, pois em muitos casos se deseja controlar de modo ótimo o que justifica a teoria de controle ótimo [2].

Um dos principais resultados na teoria controle ótimo é a solução provida pelo regulador linear quadrático (*Linear Quadratic Regulator (LQR)*), ou ainda, sua versão em tempo discreto o DLQR, que tem por objetivo garantir a estabilidade do sistema frente a pequenas perturbações, melhorando o tempo de resposta, ao minimizar a energia dissipada no processo da realimentação de estados para analisar os sistemas dinâmicos. A solução do LQR esta diretamente relacionada com a equação algébrica de Riccati (*Algebraic Riccati Equation (ARE)*), ou sua versão em tempo discreto conhecida como Equação Algébrica de Riccati Discreta (*Discrete Algebraic Riccati Equation (DARE)*) [3].

A DARE pode ser solucionada de diversas formas, por exemplo, a abordagem baseada nos auto vetores, a abordagens relacionadas a subespaços invariante [1, 4, 5], as quais podem ser consideradas como abordagens *propriamente matemática*, pois requerem rigor e precisão matemáticas. No entanto, estas abordagens apresentam uma série de problemas, dentre eles, o fato de não poderem ser implementadas em tempo real devido ao custo computacional para obter a solução da DARE em diversos sistemas, sobretudo sistemas com ordem superior a três [6].

Para contornar o problema do custo computacional, diversas técnicas são utilizadas para estimar a solução da DARE, tais como, logica Fuzzy [7, 8], algoritmo genético [9], redes neurais [10, 11, 12] entre outras. Tais métodos são largamente utilizados para solucionar a DARE, mesmo que não mantenham o rigor ou precisão matemáticas, logo o objetivo principal destas técnicas é obter, mesmo que de forma aproximada, a solução da DARE que possibilite aplicações em tempo real.

No contexto de soluções em tempo real para a DARE, uma outra técnica de estimação paramétrica tem sido utilizada, a filtragem adaptativa. Será utilizada nesta dissertação um filtro adaptativo como estimador não quadrático de parâmetros para a DARE. O estimador é

similar ao *Recursive least squares (RLS)* [13, 14], mas com um desempenho melhor em termos de velocidade de convergência e precisão de estimativa, sem aumento significativo da complexidade computacional [15]. Este estimador pode ser visto como uma generalização do RLS, o qual toma não só uma potencia par do erro, mas sim a soma das potencias pares do erro. O estimador é denominado *Recursive Least Non-Squares (RLNS)* [16, 17, 18, 19].

Um outro aspecto para que possamos solucionar a DARE de forma geral, é garantir que a DARE seja numericamente bem condicionada. Para isso, será utilizada uma técnica de decomposição matricial conhecida como inversa de Moore-Penrose ou inversa generalizada [20, 21].

Esta dissertação será baseada nestas duas técnicas, a filtragem adaptativa como técnica de estimação dos parâmetros para a DARE e a decomposição matricial, a qual garantirá o condicionamento numérico da solução para a DARE.

1.1 Motivações

Na atualidade, diversos problemas são encontrados nos mais distintos tipos de setores ligados a indústria, ciência, tecnologia, economia e muitos outros os quais necessitam de um controle que garanta o funcionamento do processo. A teoria de controle ótimo visa apresentar uma abordagem eficiente e robusta para esses problemas.

Uma grande variedade de técnicas tem sido aplicadas com o objetivo de controlar sistemas de forma ótima e em tempo real. No entanto, o controle ótimo em tempo real não é um problema simples do ponto de vista computacional. O controle de sistemas multivariáveis se dá pela solução da DARE que é uma equação matricial, aumentando assim o custo computacional. Por exemplo, no caso de uma matriz quadrada $n \times n$ as operações são da ordem de n^3 [6]. Portanto, sistemas com ordem superior a três são complexos para efetuar o controle em tempo real de forma ótima, pois requerem uma quantidade significativa de operações em um curto período de tempo. O objetivo então é desenvolver metodologias e/ou técnicas que possam se adaptar mais rapidamente as variações do sistema e com isso possibilitar o controle em tempo real.

Neste trabalho propomos uma metodologia para solucionar de forma geral a DARE, e consequentemente controlar de forma ótima sistemas nos quais esta equação esta associada. A fim de obter estimativas mais precisas e uma velocidade de convergência mais acentuada quando comparado ao RLS padrão. Propomos então, uma função de custo baseada na soma ponderada exponencialmente das potencias pares do erro [22], tornando assim o RLS, um caso particular dessa função. Assim, o problema é caracterizado e formulado como um problema de estimativa de parâmetros via RLNS.

Portanto, a solução geral e em tempo real da DARE associada ao modelo em espaço de estado de um gerador eólico é a principal motivação desse trabalho.

1.2 Objetivo

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é fornecer uma metodologia para solucionar de forma geral a DARE, o que possibilitará o controle ótimo de diversos sistemas multivariáveis (será utilizado o modelo em espaço de estado de um gerador eólico).

1.2.2 Objetivo Especifico

- Investigar as estratégias de convergência do estimadores RLNS e RLS.
- Avaliar o condicionamento numérico da metodologia proposta e da metodologia convencional.

1.3 Organização do Texto

Esta dissertação é organizada em cinco capítulos e dois apêndices. O primeiro capítulo apresenta uma introdução a respeito da DARE e sua relevância para a teoria de controle ótimo, bem como, a motivação e os objetivos gerais e específicos desta dissertação. O segundo capítulo trata exclusivamente da DARE, apresentando seu contexto histórico e sua aplicabilidade na teoria de controle ótimo, apresenta ainda o modelo espaço de estado de um gerador eólico, o qual será utilizado para avaliação da metodologia proposta. O terceiro capítulo consta as contribuições técnicas-científicas desta dissertação apresentando a metodologia proposta para solução da DARE. As simulações e resultados são abordados no quarto capítulo, que utiliza o modelo espaço de estado de um gerador eólico para avaliar a solução DARE. As conclusões e perspectivas de trabalhos futuros são descritos no quinto capítulo. No apêndice *A* é apresentado os artigos publicados e submetidos bem como os convites a futuras publicações. No apêndice *B* é abordado a decomposição em valores e vetores singulares (SVD) e a decomposição de Schur.

Equação Algébrica de Riccati Discreta

2.1 Introdução

O estudo sistemático do equação algébrica de Riccati discreta (*Discrete Algebraic Riccati Equation (DARE)*) começou no início da década de 1960, motivado principalmente por problemas de filtragem ótima lineares abordadas por Kalman [23] e por um controle otimizado de sistemas no contexto do Regulador Linear Quadrático [24]. Um esquema iterativo para solucionar a DARE foi proposto por Vaughan (1970), e em Hewer (1971). A abordagem sobre a teoria geométrica e subespaço invariante foi desenvolvido por Kucera (1972) [4], Pappas et al. (1980) [5] e Lancaster et al. (1986) [1].

Segundo Lancaster et al. [1], para solucionar problemas como do Regulador Linear Quadrático em sua versão discretizada (DLQR), se faz necessário a DARE definida por:

$$(A^T P A) - P + Q - A^T P B (R + B^T P B)^{-1} B^T P A = 0, \quad (2.1)$$

sendo A e B matrizes quadradas, $R^{m \times m} > 0$ e $Q^{n \times n} \geq 0$ são matriz de ponderação e $P^{n \times n} > 0$ é a matriz solução.

Este trabalho trata essencialmente da solução e aplicação da DARE, propondo uma forma geral de solução para a equação. Portanto, a fim de promover linearidade e fluidez de raciocínio os estudos a cerca da equação algébrica de Riccati em sua forma continua [25, 26, 27, 28, 10], ou ainda, a teoria geométrica para a DARE, bem como os de condições iniciais mais relaxadas não serão apresentadas neste trabalho, no entanto podem ser encontradas em Lancaster et al. [1].

2.2 DARE e o Controle Ótimo

Grandes esforços são feitos hoje para o desenvolvimento de sistemas de energia alternativa, como solar e eólica [29]. Nestes sistemas encontramos efeitos indesejáveis, provocados pelas não-estacionariedade naturais devido às mudanças ambientais de temperatura e pressão. Para minimizar e, por vezes, evitar esses efeitos indesejáveis é que, em muitos casos, deseja-se controlar esses sistemas de forma ótima, o que justifica a aplicação da teoria de controle ótimo neste contexto. Um dos principais resultados na teoria são as soluções fornecidas pelo regulador linear quadrático, e sua versão discreta o DLQR [3].

Feng *et al* [30] apresentou um método de projeto LQR para controle não-linear de um gerador de indução duplamente alimentados (*Doubly Fed Induction Generator (DFIG)*). O sistema foi concebido para melhorar a estabilidade transitória dos sistemas de energia e para melhorar o sistema de amortecimento. Mota *et al* [31] apresentou estratégias para problemas de estabilidade transitória, ou seja, uma estratégia de controle baseada no LQR para DFIG. As matrizes de ponderação do projeto LQR para o DFIG foram ajustadas utilizando filtros adaptativos como o *Least Mean Square (LMS)* e o *Recursive Least Square (RLS)* [31, 32].

O LQR tem por objetivo, atingir um determinado desempenho usando menor custo possível. A fim de alcançar esse desempenho, a equação algébrica Riccati (*Algebraic Riccati Equation (ARE)*) ou a versão em tempo discreto DARE, são utilizadas em certos casos.

A DARE pode ser solucionada de diversas formas, como por exemplo a abordagem baseada nos auto vetores, a abordagens relacionadas a subespaços invariantes [4], Pappas et al. [5] e Lancaster et al. [1], as quais requerem rigor e precisão matemáticas. No entanto, estas abordagens apresentam uma série de problemas, dentre eles, o fato de não poderem solucionar em tempo real a DARE devido ao seu custo computacional, sobretudo em sistemas com ordem superior a três [6]

No contexto de soluções em tempo real para a DARE, varias técnicas são utilizadas, como por exemplo, a logica Fuzzy [7, 8], algoritmo genético [9], redes neurais [10, 11, 12] entre outras. Neste trabalho propomos estimar a solução da DARE utilizando um combinador linear adaptativo baseado na soma das potencias pares do erro, o RLNS [15, 16, 17, 18, 19]. Um outro aspecto para que possamos solucionar a DARE de forma geral, é garantir o seu condicionamento numérico. Para isso, será utilizado uma técnica de decomposição matricial conhecida como inversa de Moore-Penrose ou inversa generalizada.

A seguir é apresentado o modelo em espaço de estado de um gerador eólico, no contexto do DLQR.

2.2.1 Modelo em Espaço de Estado de um Gerador Eólico

Assumindo que uma política de controle é dada pelo mapeamento de $h : X \rightarrow U$, sendo X o espaço de estado e U é o espaço de ação de controle que produz uma ação u_k a ser tomada no tempo k . Uma grande classe de importantes sistemas de tempo discreto pode ser descrito na forma de espaço de estado, dado por [3]

$$f(x_k, u_k) = Ax_k + Bu_k \quad (2.2)$$

$$u_k = h(x_k) = -Kx_k, \quad (2.3)$$

sendo $x_k \in \mathbb{R}^n$ o vetor associado a matriz de estado A , $u_k \in \mathbb{R}^m$ é o vetor associado a matriz de entrada B e K é a matriz de ganho da realimentação de estados.

Reescrevendo a Eq.(2.2) em função dos parâmetros do gerador de energia eólica, tem-se [33]:

$$f(x_k, u_k) = \frac{d}{dt} \begin{bmatrix} i_{dr}^a \\ i_{qr}^a \\ \omega_m \\ i_{gd}^e \\ i_{gq}^e \\ O_c \end{bmatrix} = \begin{bmatrix} -\frac{R_r}{\partial L_r} & \omega_{sl} & 0 & 0 & 0 & 0 \\ -\omega_{sl} & -\frac{R_r}{\partial L_r} & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{D_t}{2H} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{R}{L} & \omega_e & 0 \\ 0 & 0 & 0 & -\omega_e & -\frac{R}{L} & 0 \\ 0 & 0 & 0 & \frac{3O_{gd}^e}{2C_c O_c} & 0 & \frac{3O_{gd}^e i_{gd}^e}{2C_c O_c^2} \end{bmatrix} \begin{bmatrix} i_{dr}^a \\ i_{qr}^a \\ \omega_m \\ i_{gd}^e \\ i_{gq}^e \\ O_c \end{bmatrix} + \begin{bmatrix} \frac{1}{\partial L_r} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\partial L_r} & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2H} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{L} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{L} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{C_c} \end{bmatrix} \begin{bmatrix} O_{dr}^a \\ O_{qr}^a \\ T_e \\ O_{gd}^e \\ O_{gq}^e \\ i_c \end{bmatrix} \quad (2.4)$$

sendo C_c e capacitancia no barramento de corrente continua, D_t e constante de amortecimento do sistema concentrado no eixo do gerador, H constante de inercia, i_{dr}^a Corrente do eixo direto do rotor no referencial do fluxo estatorico, i_{qr}^a Corrente do eixo de quadratura do rotor no referencial do fluxo estatorico, i_c corrente no barramento, L indutancia, i_{gd}^e referencia da malha interna da corrente, L_r indutancia propria do rotor, R matriz de ponderação do controle, R_r resistencias do estator, T_e conjugado eletromagnetico desenvolvido pela maquina, O_c tensão no barramento CC, O_{dr}^a tensão d do rotor no referencial do fluxo estatorico, O_{qr}^a tensão do eixo de quadratura do rotor no referencial do fluxo estatorico, O_{gd}^e e O_{gq}^e sao as tensões de referencias, ω_m velocidade mecanica do rotor, ω_{sl} e a frequencia angular de escorregamento.

A função valor c associado ao sistema da Eqs.(2.2)-(2.3) tem uma forma quadrática que é dada por

$$c(x_k, u_k) = x_k^T Q x_k + u_k^T R u_k, \quad (2.5)$$

sendo as matrizes de ponderação $Q \in \mathfrak{R}^{6 \times 6} \geq 0$ e $R \in \mathfrak{R}^{6 \times 6} > 0$ simétricas e constituem respectivamente as ponderações nos vetores de estado e de controle.

O objetivo principal do controle DLQR, é selecionar uma política de controle K , que minimiza uma função valor dada por

$$\begin{aligned} V^K(x_k) &= \sum_{i=k}^{\infty} \gamma^{i-k} (x_i^T Q x_i + u_i^T R u_i) \\ &= \sum_{i=k}^{\infty} \gamma^{i-k} x_i^T (Q + K^T R K) x_i, \quad \forall x_k \in X. \end{aligned} \quad (2.6)$$

onde γ e o fator de desconto.

A solução ótima do DLQR, admite o seguinte forma quadrática [3]

$$V^K(x_k) = x_k^T P x_k \quad (2.7)$$

para algumas matriz simétrica $P \in \mathfrak{R}^{6 \times 6} > 0$. As Eqs. (2.6) and (2.7) tem as mesmas soluções.

Formulação DLQR

Após algumas manipulações algébricas com Eqs.(2.6) e (2.7), a equação de Bellman para o DLQR é dada por

$$x_k^T P x_k = x_k^T Q x_k + u_k^T R u_k + \gamma(x_{k+1}^T P x_{k+1}). \quad (2.8)$$

Reescrevendo a Eq.(2.8) em termos de ganho de realimentação da Eq.(2.3) e da dinâmica do sistema circuito fechado $x_{k+1} = (A - BK)x_k$ é assintoticamente estável, é expresso por

$$x_k^T P x_k = x_k^T [Q + K^T R K + \gamma(A - BK)^T P (A - BK)] x_k. \quad (2.9)$$

Uma vez que a Eq.(2.9) deve ser satisfeita para todos os estados x_k , tem-se uma equação linear em P , que é dada por

$$\gamma(A - BK)^T P (A - BK) - P + Q + K^T R K = 0. \quad (2.10)$$

Reescrevendo a Eq.(2.8) em função da Eq.(2.6), tem-se

$$x_k^T P x_k = x_k^T Q x_k + u_k^T R u_k + \gamma(Ax_k + Bu_k)^T P (Ax_k + Bu_k). \quad (2.11)$$

A política de controle deve satisfazer

$$R u_k + \gamma B^T P (A x_k + B u_k) = 0, \quad (2.12)$$

e o ganho de realimentação ótimo é dada por

$$K = -\gamma(R + \gamma B^T P B)^{-1} B^T P A. \quad (2.13)$$

Substituindo a Eq.(2.13) na Eq.(2.11), é obtido a equação da otimalidade de Bellman para a parametrização DLQR, dada por

$$\gamma(A^T P A) - P + Q - \gamma[A^T P B (R/\gamma + B^T P B)^{-1} B^T P A] = 0. \quad (2.14)$$

Esta equação também é conhecida como equação algébrica Riccati discreto (DARE).

2.3 Considerações Finais

Neste capítulo foi definido a DARE e apresentado o seu contexto histórico, bem como, a relação entre a teoria de controle ótimo e a DARE, para isso, o modelo em espaço de estado do gerador eólico foi utilizado.

O próximo capítulo é abordado a solução da DARE. Primeiramente, estimando a matriz P e posteriormente, apresenta a estratégia para o condicionamento numérico da DARE.

Solução da Equação Algébrica de Riccati Discreta

3.1 Introdução

Por definição [1], para solucionar a DARE, deve-se garantir que $(R + B^T P B)$ seja inversível. No entanto, isso só será possível se P existir. Portanto, para solucionar a DARE efetuam-se duas etapas:

1. A primeira etapa é estimar a matriz P . Para isso, serão utilizados dois estimadores; são eles, o RLS e o RLNS. Entretanto, qualquer técnica para a estimação de paramétrica poderá ser utilizada.
2. A segunda é garantir o condicionamento numérico do termo $(R + B^T P B)$. O condicionamento numérico deste termo se da por uma técnica de decomposição matricial conhecida como inversa de Moore-Penrose [20, 21].

Os passos descritos nas etapas anteriores são condições necessárias para solucionar de forma geral a equação algébrica de Riccati em sua forma discreta.

Assim, serão estimados os valores da matriz

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{pmatrix}, \quad (3.1)$$

utilizando os estimadores RLS e RLNS.

Depois, será realizado o condicionamento numérico do termo $(R + B^T P B)$.

3.2 Primeira Etapa: Estimação da matriz solução P para a DARE

A estimação dos parâmetros da matriz P de modelos lineares é dada por

$$P = \sum_{i=1}^n w_i p_{nn} = w_1 p_{11} + w_2 p_{12} + \cdots + w_n p_{nn},$$

ou ainda,

$$P = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{pmatrix} = w^T P_0, \quad (3.2)$$

sendo P_0 é a matriz de inicialização do sistema e W é a matriz de coeficientes.

As soluções para a Eq.(3.2) podem ser divididas em dois grupos. O primeiro grupo utiliza os métodos clássicos, os quais, requerem rigor e precisão matemática, aumentando o custo computacional e por consequência, dificulta aplicações em tempo real. Para possibilita aplicações em tem real, um segundo grupo é proposto para solucionar a DARE de forma estimada, estas soluções são providas por métodos adaptativos. A seguir são apresentados as soluções clássicas e as soluções estimadas para a DARE.

3.2.1 Soluções Clássicas

As soluções clássicas, as quais requerem rigor e precisão matemáticas, são propostas por autores como Lancaster et al. [1], que desenvolveu estudos sobre a Equação Algebraica de Riccati em suas formas continua e discreta. Lancaster et al. inicia com conceitos preliminares sobre a teoria geométrica da ARE e da DARE, apresentando também soluções para a ARE e para a DARE, utilizando matrizes hermitianas e lagrangeana em subespaço invariantes. Tópicos como condições de controlabilidade mais relaxada, estabilidade da solução, teoria de perturbação da ARE e da DARE, também são abordadas. Por fim, Lancaster et al. apresenta a equação de Riccati no contexto regulador linear quadrático e do filtro de Kalma discreto. Outros autores como Kucera et al. [4] e Pappas et al. [5], também propuseram técnicas para solucionar de forma clássica a DARE.

A pesar destes autores terem desenvolvido estudos acerca da DARE, pelo menos um aspecto não foi levado em consideração, que foi a solução da DARE em tempo real. Para possibilitar soluções em tempo real para a DARE, outros autores propuseram estimar a solução da DARE, mesmo que não mantenha o rigor e precisão matemáticas. Portanto, o objetivo é estima em tempo real a solução da DARE, cujo os valores estimados sejam os mais próximo quanto possíveis do valor real. Na seção seguinte é abordado as solução estimada para DARE.

3.2.2 Soluções Estimadas

Diversas técnicas são utilizadas para estimar a solução da DARE, tais como, lógica Fuzzy [7, 8], algoritmo genético [9], redes neurais [10, 11, 12] entre outras. No entanto, devido ao custo computacional, as estimativas em tempo real da matriz P fornecidas por estes métodos, ainda não estão próximas do valor real, acarretando em uma solução numericamente mal condicionada para a DARE. Portanto, deve-se utilizar um método computacionalmente mais simples e que forneça estimativas mais próximas aos valores reais. Deste modo, a técnica utilizada será baseada no Combinador Linear Adaptativo (CLA). Por convenção de notação, a matriz P estimada será denotada por \hat{P} , e a matriz de coeficientes w será chamada de vetor peso.

Combinador Linear Adaptativo

O combinador é chamado de linear porque para um conjunto fixo de pesos a sua saída será a combinação linear dos elementos de entrada. A idéia básica do CLA é obter um erro estimado da combinação das entradas (matriz P_0) vezes os pesos (vetor w), com uma resposta desejada (matriz P).

A Fig.3.1 ilustra um combinador linear com a resposta desejada e o erro. O erro, e_n é obtido subtraindo a saída \hat{P} do valor desejado d_n .

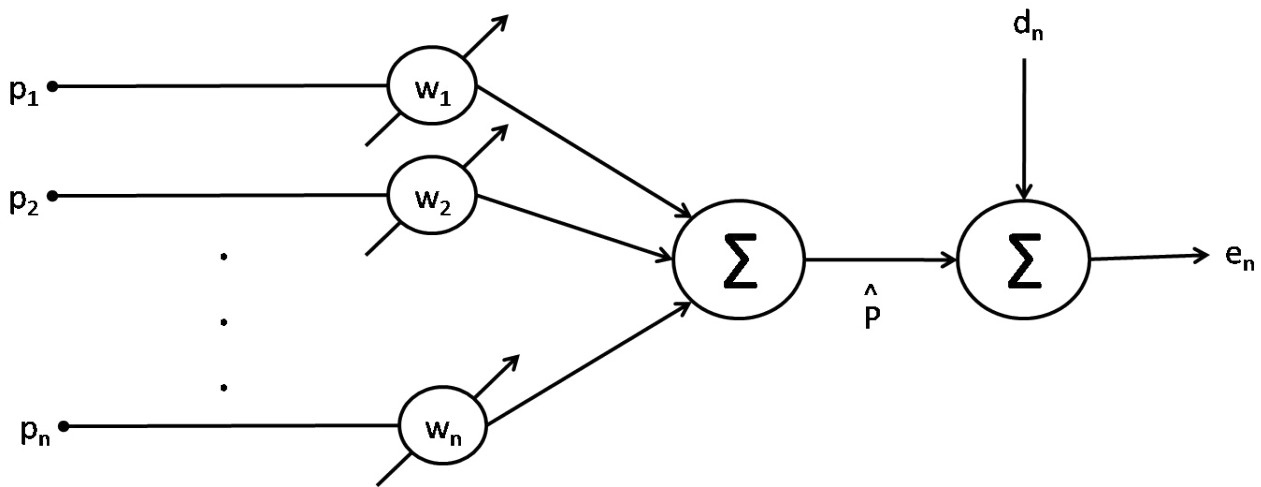


Figura 3.1: Combinador linear adaptativo com a resposta desejada e o erro.

No processo de adaptação o vetor peso do combinador linear é ajustado para produzir uma saída, \hat{P} , o mais próximo possível do valor desejado d_n . Isto é feito comparando a saída com a resposta desejada, para se obter o erro que é dado por

$$e_n = d_n - \hat{P} \quad (3.3)$$

Supondo-se que o valor dos coeficientes seja constante ao longo do tempo e substituindo Eq.(3.2) em Eq.(3.3) tem-se

$$e_n = d_n - w^T P_0. \quad (3.4)$$

De uma forma geral, a minimização do erro médio quadrático é a figura de mérito mais utilizada para verificar o desempenho da estimação. Portanto, para se obter o erro quadrático instantâneo eleva-se Eq.(3.4) ao quadrado

$$e_n^2 = d_n^2 + w^T P_0 P_0^T w - 2d_n P_0^T w. \quad (3.5)$$

Para se obter o erro médio quadrático avalia-se o valor esperado de Eq.(3.5)

$$E[e_n^2] = E[d_n^2] + w^T E[P_0 P_0^T] w - 2E[d_n P_0^T] w. \quad (3.6)$$

Pode-se representar a Eq.(3.6) definindo uma matriz ψ de autocorrelação da entrada como

$$\begin{aligned} \psi &= E \{ P_0 P_0^T \} = \\ &= E \begin{bmatrix} P_0^2 & P_0 x(n-1) & \cdots & P_0 P_0(n-N+1) \\ P_0(n-1)P_0 & P_0(n-1)^2 & \cdots & P_0(n-1)P_0(n-N+1) \\ \vdots & \vdots & \cdots & \vdots \\ P_0(n-N+1)P_0 & P_0(n-N+1)P_0(n-1) & \cdots & P_0^2(n-N+1) \end{bmatrix}. \end{aligned} \quad (3.7)$$

Os termos da diagonal principal de ψ são o quadrado dos componentes da entrada e os termos fora da diagonal principal indicam a correlação entre os elementos da entrada. A matriz de correlação traz informações sobre a dependência entre os diversos termos do vetor P_0 .

Pode-se também definir um vetor χ de correlação cruzada como

$$\chi = E[d_n P_0 d_n x(n-1) \cdots d_n x(n-N+1)]^T. \quad (3.8)$$

Este vetor quantifica a dependência entre o valor desejado e cada elemento do vetor de sinais de entrada.

Pode-se expressar Eq.(3.5) em termos de Eq.(3.6) e Eq.(3.7)

$$\xi = E\{e^2(n)\} = E\{d^2(n)\} + w^T \psi w - 2\chi^T w. \quad (3.9)$$

O gradiente da superfície de erro médio quadrático, $\nabla(\xi)$, pode ser encontrado diferenciando-se Eq.(3.9) em relação a cada peso

$$\nabla(\xi) = \frac{\partial \xi}{\partial w} = \left[\frac{\partial \xi}{\partial w_0} \quad \frac{\partial \xi}{\partial w_1} \quad \cdots \quad \frac{\partial \xi}{\partial w_{N-1}} \right]^T = 2\psi w - 2\chi. \quad (3.10)$$

Para se obter o erro médio quadrático mínimo, o vetor peso w deverá ter seu valor ótimo, w^* , isso acontece quando o gradiente for igual a zero

$$\nabla(\xi) = 0 = 2\psi w^* - 2\chi, \quad (3.11)$$

de Eq.(3.11) obtém-se o vetor peso ótimo w^*

$$w^* = \psi^{-1} \chi. \quad (3.12)$$

a Eq.(3.12) é conhecida como peso ótimo de Wiener.

Portanto, o CLA utiliza um algoritmo que, através de sucessivas iterações modifique seus coeficientes de modo que estes se aproximem o máximo possível do resultado apresentado na Eq.(3.12). Como resultado, o erro médio quadrático será mínimo, acarretando que \hat{P} será uma estimativa bastante aproximada de d_n .

Para a compreensão do erro médio quadrático mínimo, é sabido que o método dos mínimos quadrados, para o instante $n > 0$, os pesos são calculados tal que a quantidade

$$\xi_n = \sum_{i=1}^n \rho_i |e_i|^2 \quad (3.13)$$

é minimizada, motivo pelo qual recebe o nome mínimos quadrados.

Um algoritmo muito popular da família dos mínimos quadrados é o algoritmo RLS (*Recursive Least Squares*). Tal algoritmo tem como vantagem a baixa sensibilidade à natureza do sinal de entrada e uma maior velocidade de convergência quando comparado com os algoritmos de gradiente estocástico por exemplo. Um outro algoritmo não muito usual da família dos mínimos quadrados é o RLNS (*Recursive Least Non-Squares*) o aspecto chave do uso dessa nova abordagem e para incluir o erro instantâneo na determinação do vetor ganho.

O Algoritmo Mínimos Quadrados Recursivo

O algoritmo RLS (*Recursive Least Squares*) é um algoritmo de estimação paramétrica utilizado em diversas áreas, como por exemplo, filtragem adaptativa, para encontrar os coeficientes do filtro que relacionam à produção recursiva de mínimos quadrados do sinal de erro (diferença entre o sinal desejado e o verdadeiro).

No algoritmo RLS o fator ponderação ρ_n é escolhido como sendo

$$\rho_n = \lambda^{n-1} \quad (3.14)$$

sendo $0 \ll \lambda < 1$ uma constante positiva a ser escolhida.

O método dos mínimos quadrados padrão visto anteriormente corresponde ao caso em que $\lambda = 1$. O parâmetro λ é denominado fator de esquecimento. Claramente quando $\lambda < 1$, os fatores de ponderação definidos 3.14 dá um peso maior às amostras recentes das estimativas do erro comparado com as amostras mais antigas. Em outras palavras, a escolha de $\lambda < 1$ resulta em um esquema que dá mais ênfase às amostras recentes do dado observado e tende a esquecer as amostras antigas.

Substituindo 3.14 em 3.13 obtém-se a função de custo a ser minimizada na dedução do algoritmo RLS, que é dada por

$$\sum_{i=1}^n \lambda^{n-1} |e_i|^2. \quad (3.15)$$

Supondo que φ seja uma matriz não singular, o valor ótimo do vetor peso, para que a função de custo atinja este valor mínimo é definido pela equação normal escrita em forma matricial

$$w^* = \varphi_n^{-1} z_n, \quad (3.16)$$

sendo

$$\varphi_n = \sum_{i=1}^n \lambda^{n-i} p_n p_n^T, \quad (3.17)$$

a matriz de auto-correlação do sinal de entrada $p_n \in P_0$ e

$$z_n = \sum_{i=1}^n \lambda^{n-i} p_n d_n, \quad (3.18)$$

a matriz de correlação cruzada do vetor de entrada p_n com o sinal desejado d_n .

Expandindo a Eq.(3.17) e isolando o termo $i = n$, tem-se

$$\begin{aligned} \varphi_n &= \lambda \sum_{i=1}^{n-1} [\lambda^{n-1-i} p_i p_i^T] + p_n p_n^T, \\ \varphi_n &= \lambda [\varphi_{n-1}] + p_n p_n^T. \end{aligned} \quad (3.19)$$

Analogamente pode-se escrever a equação Eq.(3.18) da seguinte forma

$$\begin{aligned} z_n &= \lambda \sum_{i=1}^{n-1} [\lambda^{n-1-i} p_i p_i^T] + p_n d_n, \\ z_n &= \lambda [z_{n-1}] + p_n d_n. \end{aligned} \quad (3.20)$$

Como a matriz de auto-correlação é positiva definida e não singular, pode-se aplicar o lema de inversão de matrizes para equação recursiva 3.17, para obter a relação de recursividade da matriz φ_n , dada por

$$\varphi_n = \lambda^{-1} \varphi_{n-1} - \frac{\lambda^2 \varphi_{n-1}^{-1} p_n p_n^T \varphi_{n-1}^{-1}}{1 + \lambda^{-1} p_n^T \varphi_{n-1}^{-1} p_n}. \quad (3.21)$$

Por conveniência computacional escreve-se as seguintes igualdades

$$\eta = \varphi_n^{-1}, \quad (3.22)$$

$$\vartheta = \frac{\lambda^{-1} \eta_{n-1} p_n}{1 + \lambda^{-1} p_n^T \eta_{n-1} p_n}. \quad (3.23)$$

Reescrevendo a Eq.(3.21) como segue

$$\eta = \lambda^{-1} \eta_{n-1} - \lambda^{-1} \vartheta p_n^T \eta_{n-1} \quad (3.24)$$

sendo η a inversa da matriz de auto-correlação de dimensão $L \times L$ e ϑ o vetor ganho de dimensão $L \times 1$.

Reorganizando a Eq.(3.23), tem-se

$$\vartheta + \vartheta \lambda^{-1} p_n^T \eta_{n-1} p_n = \lambda^{-1} \eta_{n-1} p_n,$$

$$\vartheta = [\lambda^{-1} \eta_{n-1} - \vartheta \lambda^{-1} p_n^T \eta_{n-1}] p_n. \quad (3.25)$$

Substituindo a Eq.(3.24) em Eq.(3.25) tem-se que

$$\vartheta = \eta p_n,$$

$$\vartheta = \varphi_{n-1}^{-1} p_n. \quad (3.26)$$

Atualização do Vetor Peso RLS

No desenvolvimento de uma equação recursiva, são utilizadas as equações 3.16, 3.20 e 3.22 para expressar a estimativa do mínimo quadrado do vetor peso, w_n^* , no instante n como segue

$$w_n^* = \eta[\lambda z_{n-1} p_n]. \quad (3.27)$$

Substituindo a Eq.(3.24) na Eq.(3.27), tem-se

$$w_n^* = \eta[\lambda z_{n-1} p_n] w_n^* = P_{n-1} z_{n-1} - \vartheta p_n P_{n-1} z_{n-1} + \eta p_n d_n. \quad (3.28)$$

Das equações 3.26 e 3.28, segue-se

$$w_n^* = w_{n-1}^* - \vartheta \varepsilon_n \quad (3.29)$$

sendo ε_n a estimativa do erro a priori definida por

$$\varepsilon_n = d_n - u_T^n w_{n-1}^*. \quad (3.30)$$

O Algoritmo Mínimos Não-Quadrados Recursivo

Seja $J_{m,n}$ uma função par, contínua e simétrica baseada na soma das potência pares do erro, definida por

$$J_{m,n} = \sum_{j=1}^m k^{m-j} \sum_{i=1}^n \{\lambda^{n-i} [e_i]^{2j}\}, \quad (3.31)$$

sendo m , j e k inteiros positivos. Nota-se que para $j = 1$, $k = 1$ e $m = 1$ é obtido a função de custo do MSE (*Mean Square Error*). O fator peso exponencial, λ , é o fator de esquecimento convencional que normalmente é selecionado maior que zero e próximo de um, isto é, $0 << \lambda < 1$. A ponderação k^{m-j} é apenas um termo multiplicativo que desempenha o papel de acelerar a convergência de adaptação do algoritmo. Os termos mais interessantes são os que definem o primeiro somatório, j e m , ou seja, observa-se que $j > 1$ define a potência par do erro e que m adiciona diferentes potências do erro, determinando a função de custo como uma soma das potências pares do erro, definindo o erro como sendo

$$e_i = d_n - w^T p_n, \quad (3.32)$$

d_n é o valor desejado, w^T é o vetor de peso e $p_n \in P_0$ é o vetor de entrada.

A Eq.(3.31) é usada para derivar uma nova família de algoritmos do tipo RLS ainda mais rápido em relação a velocidade de convergência.

A fim de obter o vetor peso ótimo w^* , calculamos o gradiente instantâneo de $J_{m,n}$ que é dada por

$$\nabla J_{m,n} = - \sum_{j=1}^m -2 \cdot j \cdot k^{m-j} \sum_{i=1}^n \{\lambda^{n-i} p_i 2j [e_i]^{2j-1}\},$$

$$= \sum_{j=1}^m -2.a_j. \sum_{i=1}^n \{ \lambda^{n-i}. p_i. e_i^{\alpha_j}. e_i \}, \quad (3.33)$$

sendo $a_j = j.k^{m-j}$ e $\alpha_j = 2j - 2$

Assim, igualando $\nabla J_{m,n}$ a zero para que a função da Eq.(3.33) atinja o valor mínimo, definimos o valor ótimo do vetor peso, w^* , pela equação escrita em forma matricial,

$$w^* = \Phi_{m,n}^{-1} z_{m,n}, \quad (3.34)$$

sendo $\Phi_{m,n}$ a matriz de autocorrelação do vetor de entrada é definida por

$$\Phi_{m,n} = \sum_{j=1}^m \left\{ a_j \sum_{i=1}^n [\lambda^{n-1} e_i^{\alpha_j} p_i p_i^T] \right\}, \quad (3.35)$$

e o vetor de correlação cruzada, $z_{m,n}$, entre as entradas e a resposta desejada é definido por,

$$z_{m,n} = \sum_{j=1}^m \left\{ a_j \sum_{i=1}^n [\lambda^{n-1} e_i^{\alpha_j} d_n p_i^T] \right\}. \quad (3.36)$$

As deduções das Eqs.(3.37) e (3.38) serão feitas com o intuito de facilitar o cálculo da inversa da matriz de autocorrelação $\Phi_{m,n}$, que é necessária para determinar o valor estimado, w^* , do vetor ótimo de acordo com a Eq.(3.34).

Isolando o termo correspondente a $i = n$ da Eq.(3.35), tem-se

$$\Phi_{m,n} = \lambda \Phi_{m,n-1} + \left[\sum_{j=1}^m a_j. e_n^{\alpha_j} \right] p_n p_n^T. \quad (3.37)$$

Similarmente, isolando o termo correspondente a $i = n$ da Eq.(3.36), pode-se escrever

$$z_{m,n} = \lambda z_{m,n-1} + \left[\sum_{j=1}^m a_j. e_n^{\alpha_j} \right] p_n. d_n. \quad (3.38)$$

$\Phi_{m,n-1}$ é o valor anterior da matriz de autocorrelação e $z_{m,n-1}$ é o valor anterior do vetor de correlação cruzada.

Como a matriz de auto-correlação é positiva definida e não singular, pode-se aplicar o lema de inversão de matrizes para equação recursiva 3.17, para obter a relação de recursividade da matriz $\Phi_{m,n-1}$, dada por

$$\Phi_{m,n-1} = \lambda^{-1} \Phi_{m,n-1}^{-1} - \frac{\lambda^{-1} \Phi_{m,n-1}^{-1} p_n p_n^T \Phi_{m,n-1}^{-1}}{\lambda \left[\sum_{j=1}^m [a_j e_i^{\alpha_j}] \right]^{-1} p_n^T \Phi_{m,n-1}^{-1} p_n}. \quad (3.39)$$

Por conveniência de notação, tem-se

$$\eta = \Phi_{m,n}^{-1} \quad (3.40)$$

e

$$g_{m,n} = \frac{\eta_{m,n-1} p_n}{\lambda \left[\sum_{j=1}^m [a_j e_i^{\alpha j}] \right]^{-1} p_n^T \eta_{m,n-1} p_n}, \quad (3.41)$$

sendo g_n o vetor ganho.

Para facilitar futuras manipulações matemáticas, a Eq.(3.41) é reescrita como

$$g_{m,n} = \eta p_n \left[\sum_{j=1}^m [a_j e_i^{\alpha j}] \right]. \quad (3.42)$$

Assim

$$\eta = \lambda^{-1} (P_{n-1} - g_n p_n^T P_{n-1}). \quad (3.43)$$

Atualização do Vetor Peso RLNS

No desenvolvimento da estrutura de adaptação do algoritmo RLNS, serão utilizados as Eqs.(3.34), (3.38) e (3.40) para expressar a estimativa w_n^* do vetor peso no instante n como segue

$$w^* = \Phi_{m,n}^{-1} z_n,$$

$$w^* = \lambda \eta z_{m,n-1} + \eta \left[\sum_{j=1}^m \{a_j e_n^{\alpha j}\} \right] d_n p_n. \quad (3.44)$$

Substituindo a Eq.(3.43) na Eq.(3.44), tem-se

$$\begin{aligned} w^* &= \eta z_{m,n-1} - g_n p_n^T \eta z_{m,n-1} + \eta d_n p_n \left[\sum_{j=1}^m \{a_j e_n^{\alpha j}\} \right], \\ &= w_{m,n-1}^* - g_n p_n^T w_{m,n-1}^* + \eta d_n p_n \left[\sum_{j=1}^m \{a_j e_n^{\alpha j}\} \right], \end{aligned} \quad (3.45)$$

das Eqs.(3.42) e (3.45), é obtido a estrutura de atualização do algoritmo RLNS dada pela seguinte equação,

$$w^* = w_{m,n-1}^* + g_{m,n} e_n, \quad (3.46)$$

sendo e_n uma estimação a priori do erro, dada por

$$e_n = d_n - p_n^T w^*. \quad (3.47)$$

3.3 Segunda Etapa: Condicionamento numérico para DARE

O passo seguinte a estimação da matriz P , é garantir o condicionamento numérico do termo $(R + B^T P B)$. Para isso, será utilizado uma técnica de decomposição matricial.

Decompor uma matriz é na verdade fatorar a matriz em sua forma canônica. Geralmente estas decomposições são utilizadas para simplificar a análise de sistemas ou para implementar de maneira mais eficiente os algoritmos numéricos que envolvem operações com matrizes e vetores.

A tabela 3.1 lista alguns métodos de decomposição com os respectivos tipos de matrizes em que podem ser aplicados.

Tabela 3.1: Decomposições matriciais.

Decomposição	Tipo de Matriz	Fatoração	Notação
LU	$m \times n$	L.U	L: triangular inferior ($m \times m$) U: triangular superior ($m \times n$)
LUP ou LU com pivotamento parcial	$m \times n$	$P^{-1}.L.U$	L: triangular inferior ($m \times m$) U: triangular superior ($m \times n$) P: matriz de permutação ($m \times m$)
LDU	quadrada	$L.U.D$	L: triangular inferior U: triangular superior D: matriz diagonal
Cholesky	simétrica e positiva definida	$L.L^H$	L: triangular inferior
LDL	simétrica	$L.D.L^H$	L: triangular inferior D: matriz diagonal
Espectral ou em valores característicos	quadrada e valores característicos distintos	$X.D.X^{-1}$	X: matriz regular dos vetores característicos nas colunas D: matriz diagonal dos valores característicos
SVD	$m \times n$	$U.\Sigma.V^T$	U: matriz unitária ($m \times m$) V: matriz unitária ($n \times n$) Σ : diagonal ($m \times n$) com os p valores singulares ≥ 0 , $p = \min(m,n)$
QR	$m \times n$	$Q.R$	Q: matriz ortogonal ($m \times m$) R: triangular superior ($m \times n$)
Jordan	quadrada	$T.J.T^{-1}$	T: transformação similar J: forma canônica de Jordan
Schur	quadrada	$U.S.U^T$	U: matriz unitária S: forma canônica de Schur, triangular superior com os valores característicos na diagonal
Hessenberg	quadrada	$Q.H.Q^T$	Q: matriz ortogonal H: forma canônica de Hessenberg, elementos abaixo da sub-diagonal nulos. Se A for simétrica ou hermitiana então H é tridiagonal.

Portanto, varias são as formas para decompor uma matriz, dentre elas, será utilizado uma técnica chamada decomposição em valores singulares (*Singular Value Decomposition (SVD)*), mais especificamente uma aplicação da SVD conhecida como **inversa de Moore-Penrose ou inversa generalizada**.

Inversa de Moore-Penrose

O cálculo da matriz inversa é comum em problemas de otimização, linear bem como em outras áreas. Nesta seção, generaliza-se o conceito de inversa de modo que toda matriz possua uma inversa generalizada. Essa generalização é conhecida como pseudo-inversa de Moore-Penrose.

Por conveniência de notação, tem-se $(R + B^T P B) = Y$ e Y^+ é a inversa generalizada ou inversa de Moore-Penrose.

Moore, em 1920, definiu uma inversa generalizada e provou sua unicidade [20]. Alguns anos depois, em 1955, Penrose mostrou que para toda matriz $Y \in \mathbb{R}^{m \times n}$ existe uma única matriz $Y^+ \in \mathbb{R}^{n \times m}$ satisfazendo as seguintes condições [21]:

1. $Y Y^+ Y = Y$;
2. $Y^+ Y Y^+ = Y^+$;
3. $(Y Y^+)^T = Y Y^+$;
4. $(Y^+ Y)^T = Y^+ Y$.

Essas condições são chamadas de *Condições de Penrose*. O teorema a seguir, mostra que a matriz Y^+ satisfazendo as condições Penrose existe e é única.

Teorema 1 (inversa de Moore-Penrose). *Seja Y uma matriz $m \times n$. Então existe uma única matriz Y^+ de ordem $n \times m$ que satisfaz as condições Penrose.*

Demonstração. Seja $U \Sigma V^T$ a fatorização em valores singulares de Y , tal que $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ são matrizes ortonormais e $\Sigma \in \mathbb{R}^{m \times n}$ é uma matriz diagonal cujos elementos são autovalores não nulos. Então

$$Y^+ = V \Sigma^+ U^T, \quad (3.48)$$

sendo Σ^+ é a matriz diagonal $n \times m$, do tipo

$$\Sigma^+ = \begin{pmatrix} \Sigma_1^{-1} & \dots & 0 \\ 0 & \Sigma_2^{-1} & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \Sigma_n^{-1} & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix},$$

a matriz Σ^+ possui as últimas $n - r$ linhas nulas e as últimas $m - r$ colunas nulas. Portanto $\Sigma \Sigma^+ = I_m$.

Então,

1. $Y Y^+ Y = U \Sigma V^T (V \Sigma^+ U^T) U \Sigma V^T = U \Sigma \Sigma^+ \Sigma V^T = V \Sigma U^T = Y$;
2. $Y^+ Y Y^+ = V \Sigma^+ U^T (U \Sigma V^T) V \Sigma^+ U^T = V \Sigma^+ U^T = Y^+$;
3. $(Y Y^+)^T = (U \Sigma V^T (V \Sigma^+ U^T))^T = (U \Sigma \Sigma^+ U^T)^T = (U U^T)^T = (I)^T = I$;
por outro lado,
 $Y Y^+ = U \Sigma V^T (V \Sigma^+ U^T) = U \Sigma \Sigma^+ U^T = U U^T = (I)^T = I$;
4. $(Y^+ Y)^T = (V \Sigma^+ U^T (U \Sigma V^T))^T = (V \Sigma^+ \Sigma V^T)^T = (V V^T)^T = (I)^T = I$;
por outro lado,
 $Y^+ Y = V \Sigma^+ U^T (U \Sigma V^T) = V \Sigma^+ \Sigma V^T = V V^T = I$

Suponha que exista um matriz C que também satisfaça as condições de Penrose. Então,

$$\begin{aligned} C &\stackrel{2}{=} CY^+C \stackrel{3}{=} C(Y^+C)^T = CC^TY^T \stackrel{1}{=} CC^T(YCY)^T = \\ &= C(C^TY^T)((Y^+)^TY^T) = C(YC)^T(YY^+)^T \stackrel{3}{=} \\ &\stackrel{3}{=} CY, CYY^+ \stackrel{1}{=} CYY^+ \end{aligned}$$

por outro lado

$$\begin{aligned} Y^+ &\stackrel{2}{=} Y^+YY^+ \stackrel{4}{=} (Y^+Y)^TY^+ = Y^T(Y^+)^TY^+ \stackrel{1}{=} (YCY)^T \\ (Y^+)^TY^+ &= (Y^TC^T)(Y^T(Y^+)^T)Y^+ = (CY)^T(Y^+Y)^TY^+ \\ &\stackrel{4}{=} CYY^+Y^+Y^+ = CYY^+ \end{aligned}$$

Logo $Y^+ = C$, assim Y^+ é a única matriz que satisfaz as condições de Penrose. \square

3.4 Considerações Finais

Este capítulo apresentou a metodologia proposta para solução da DARE, a qual, baseia-se em duas etapas. A primeira é obter a matriz P , para isso, duas classes foram apresentadas, as soluções clássicas e as soluções estimadas para a DARE. E a segunda etapa apresentou a estratégia de condicionamento numérico, baseado na inversa generalizada, para a DARE.

No próximo capítulo será utilizado o modelo em espaço de estado do gerador eólico descrito na Eq.(2.4) para obtermos os parâmetros da DARE referentes a Eq.(2.1), com o objetivo de avaliar a metodologia proposta.

Solução da Equação Algébrica de Riccati Discreta Utilizando o Modelo em Espaço de Estado do Gerador Eólico

4.1 Simulações

As simulações foram realizadas no modelo em espaço de estado do gerador eólico descrito na Eq.(2.4) para obtermos os parâmetros da DARE referentes a Eq.(2.1). O objetivo das simulações é avaliar as estimativas dos elementos da matriz P e a velocidade de convergência, utilizando a metodologia proposta (matriz inversa generalizada) e comparar com a da metodologia convencional (matriz inversa). Os parâmetros do gerador eólico utilizados nas simulações são apresentados na Tabela 4.1 [33]:

Tabela 4.1: Parâmetros do gerador eólico

Parâmetros	Valor	Unidade
Resistência no Estator (R_s)	3	Ohms
Resistência no Rotor (R_r)	2.9876	Ohms
Indutância no Estator (L_s)	0.0149	H
Indutância no Rotor (L_r)	0.015	H
Indutância Mútua (L_m)	0.5992	H
Constantes de Inércia (H_T)	0.01	-
Coefficiente de Inércia (D_t)	0.015	Nms/(rad)
Tensão Nominal (V_{nom})	220	V
Corrente Nominal (I_{nom})	3.66	A
Conjugado (T_{nom})	6.36	Nm
Velocidade nominal (ω_{nom})	377	rad/s
Frequência (f_s)	60	Hz
Número de pares de pólos (P)	1	-

Os valores da Tabela 4.1 foram aplicados na Eq.(2.4) para obter as matrizes

$$A = \begin{bmatrix} -101.48 & 193.74 & 0.00 & 0.00 & 0.00 & 0.00 \\ -193.74 & -101.48 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & -0.15 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & -250.00 & 377.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & -377.00 & -250.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & -767.72 & 0.00 & 22.72 \end{bmatrix} \quad (4.1)$$

e

$$B = \begin{bmatrix} 33.97 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 33.97 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 100.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & -83.33 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & -83.33 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 454.54 \end{bmatrix}. \quad (4.2)$$

As matrizes de ponderação Q e R utilizadas foram

$$Q = \begin{bmatrix} 100.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 100.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 100000.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1000.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.10 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1000.00 \end{bmatrix} \quad (4.3)$$

e

$$R = \begin{bmatrix} 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \end{bmatrix}. \quad (4.4)$$

O valor de referência da matriz P , foi obtido por meio do método de Schur [34], amplamente utilizado na literatura, e é dada por

$$P_{Schur} = \begin{bmatrix} 103.0361 & 1.5927 & 0 & 0 & 0 & 0 \\ 1.5927 & 103.0361 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0002 & 0.2146 & -0.0793 \\ 0 & 0 & 0 & 0.2146 & 0.3337 & -0.1021 \\ 0 & 0 & 0 & -0.0793 & -0.1021 & 1.0001 \end{bmatrix}. \quad (4.5)$$

A matriz solução P é estimada pela Eq.(3.2) e avaliada na Eq.(2.1). A estrutura da estimação da matriz P é dada pela Fig.4.1

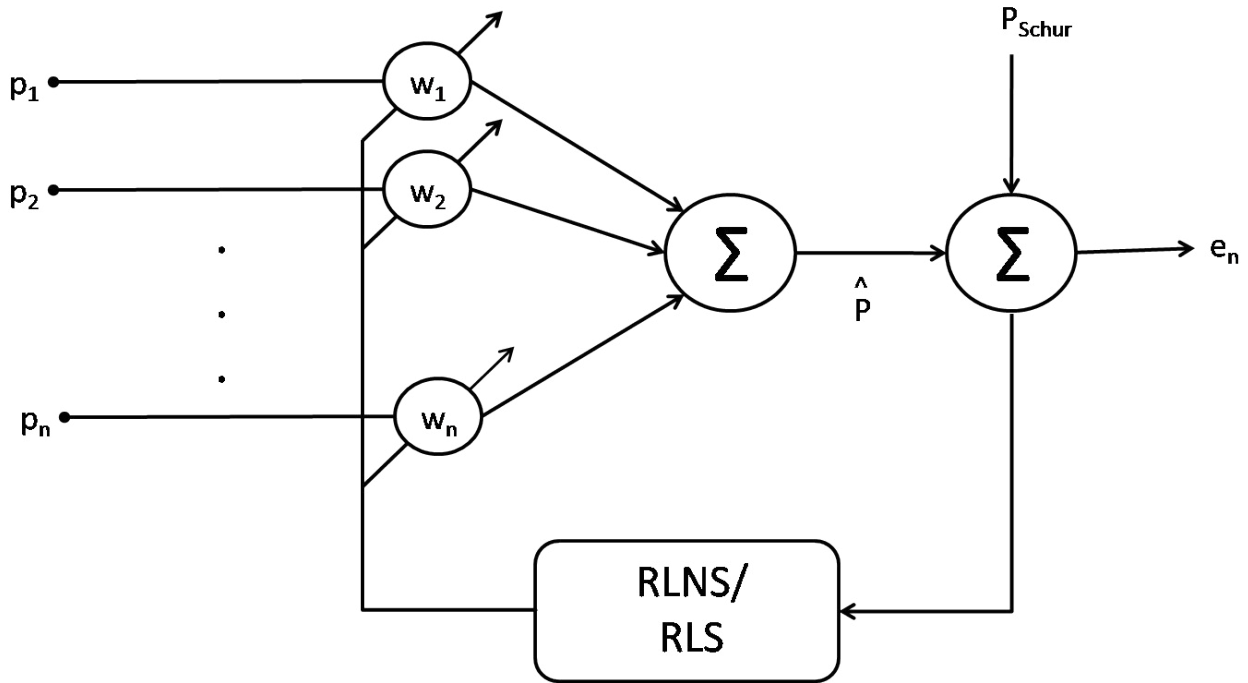


Figura 4.1: Estimação dos elementos da matriz solução P utilizando o RLNS e o RLS. A estrutura da estimação é dada pela combinação das entradas p_n com os pesos w_n produzindo o valor estimado \hat{P} . Posteriormente, o erro é calculado pela diferença entre o valor estimado \hat{P} e o valor de referência P_{Schur} , objetivando um erro mínimo. Por isso, o processo é adaptado pelos estimadores RLNS e RLS até a n -ésima iteração.

Os p_n pertencem a matriz de inicialização do sistema P_0 , que é dada por

$$P_0 = \begin{bmatrix} 100000.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 100000.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 100000.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 100000.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 100000.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 100000.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 100000.00 \end{bmatrix}. \quad (4.6)$$

Condicionamento Numérico

Para avaliar o condicionamento numérico da solução para a DARE, os estimadores RLS e RLNS utilizaram os seguintes parâmetros

Parâmetros da simulação utilizando a matriz inversa

- Os parâmetros utilizados pelo RLS foram:
 1. As matrizes A, B, Q, R, P_{Schur} e P_0 são os mesmo mencionados anteriormente;
 2. Ciclo de 1500 iterações.
- Os parâmetros utilizados pelo RLNS foram os mesmos parâmetros que o RLS, além de mais dois parâmetros livres:
 1. A mais alta ordem da função de custo ($M = 6$), que também controla o número de termos na soma;
 2. E o parâmetro k_i acelerador do ganho ($k_1 = 28, k_2 = 15$ e $k_3 = 6$).

Parâmetros da simulação utilizando a inversa generalizada

- Os parâmetros utilizados pelo RLS foram iguais aos do item anterior.
- Para o RLNS utilizamos os mesmos parâmetros que o RLS, além de mais dois parâmetros livres:
 1. $M = 4$;
 2. $k_1 = 27$ e $k_2 = 0.51$.

O fator de esquecimento (λ) para os estimadores RLS e RLNS foi escolhido de acordo com a Tabela 4.2

Tabela 4.2: Ajuste do fator de esquecimento (λ)

Parâmetros	p_{11}		p_{22}		p_{33}		p_{44}		p_{55}		p_{66}	
	Referencia	103.0361	RLNS	RLS	RLNS	RLS	RLNS	RLS	RLNS	RLS	RLNS	RLS
Estimadores		103.0361	RLNS	RLS	RLNS	RLS	RLNS	RLS	RLNS	RLS	RLNS	RLS
$\lambda = 0.90$	122.6079	103.0782	103.3313	-1.7536	1.0883	1.6639	984.3217	1.2900	25.8030	2.5940	1.0571	
$\lambda = 0.91$	102.3734	103.0651	103.2958	1.1704	9.8988	2.0435	997.2209	-2.2298	-5.6988	3.5948	1.1872	
$\lambda = 0.92$	111.0750	103.1231	103.0926	1.8575	4.7631	4.5643	2.6893	33.7813	76.7604	17.3460	6.9865	
$\lambda = 0.93$	102.3067	103.0451	103.2929	9.8892	1.0659	996.9328	1.1303	-9.6617	3.4678	-127.2150	1.8824	
$\lambda = 0.94$	102.3208	111.0426	106.9395	9.9818	-5.6809	885.3782	7.5102	-6.8767	5.8140	1.1006	1.1544	
$\lambda = 0.95$	102.5850	103.0186	103.0121	9.5564	1.0363	3.9419	964.4110	89.7459	0.2080	3.3932	1.0636	
$\lambda = 0.96$	1.2436	103.0392	103.1740	-2.7576	9.3540	5.2802	1.3553	4.7261	8.4339	-3.8163	1.5908	
$\lambda = 0.97$	101.7521	103.0751	103.0520	9.6089	9.8961	2.1870	1.0444	-3.8087	5.7992	-7.5497	973.2835	
$\lambda = 0.98$	103.0395	103.0344	103.3711	1.0013	1.0000	982.9840	1.0011	-0.8314	0.4016	1.0157	1.0005	
$\lambda = 0.99$	106.7929	103.3195	107.2263	9.9492	9.9855	814.0200	1.0011	24.6867	0.0405	1.5620	994.5690	

Na tabela 4.2 o fator de esquecimento variou entre 0.90 a 0.99 para os filtros RLS e RLNS. O fator de esquecimento que melhor estimou os parâmetros da matriz P foi o $\lambda = 0.98$ para ambos estimadores. No entanto, os elementos de p_{11} a p_{66} mostram que o RLNS obteve estimativas mais precisas quando comparadas com as do RLS, como por exemplo, o elemento p_{33} que tem como valor de referencia 1.0000, estimou para o RLS o valor de 9.9981, enquanto que para o RLNS o valor foi de 1.0002.

A variação de um centésimo no fator de esquecimento na tabela 4.2 foi significativa para o RLS, como mostra o elemento p_{11} que para $\lambda = 0.90$ o valor estimado foi de 122.6079 enquanto que para $\lambda = 0.91$ o valor estimado foi 102.3734. No entanto, as estimativas de p_{11} feitas pelo RLNS para os mesmos fatores de esquecimento foram de 103.0782 e 103.0651 respectivamente. Comportamentos similares podem ser observados nos demais parâmetros, caracterizando então uma maior robustez a variação do fator de esquecimento para o filtro RLNS do que para o RLS.

4.2 Resultados

Os resultados são expressos em função do erro percentual dos algoritmos RLS e RLNS para a metodologia proposta (matriz inversa generalizada) e comparada com a da metodologia convencional (matriz inversa). Ajustando o tempo de adaptação como o número de iterações necessárias para o erro permanece dentro da gama de 1% do valor de estado estacionário para diagonal principal e 5% do valor de estado estacionário para fora diagonal principal, o (-) significa que os algoritmos RLNS e/ou RLS não atingiram o valor do estado estacionário [35].

A Tabela 4.3 mostra a estimação do erro para os elementos da diagonal principal da matrix P . As estimações são feitas utilizando os filtros RLNS e RLS para metodologia clássica e para metodologia proposta.

Tabela 4.3: Número de iterações para o RLNS e RLS atingir o valor de estado estacionário com um erro inferior a 1%, para metodologia clássica e para metodologia proposta.

Erro ≤ 1	Matriz inversa		Matriz inversa generalizada	
	Número de iterações			
	RLS	RLNS	RLS	RLNS
p_{11}	1706	362	634	299
p_{22}	-	445	572	279
p_{33}	1664	350	676	256
p_{44}	-	1848	1237	256
p_{55}	1730	823	656	312
p_{66}	1748	1034	589	361

A Tabela 4.3 mostra que para todo os elementos da diagonal principal da matriz P o número de iterações que o RLNS leva para atingem o valor de estado estacionário e inferior ao RLS para ambas as metodologia, caracterizando uma maior velocidade de convergência do RLNS quando comparado ao RLS. Mostra também que o a metodologia proposta é mais robusta a instabilidades numéricas, enquanto que a metodologia convencional não foi capaz de garantir estabilidade numérica para o estimador RLS para essa gama de erro, como visto nos elemento p_{22} e p_{44} .

A Fig4.2 ilustra o comportamento da estimação do erro para os elementos da diagonal principal da matrix P . As estimações são feitas utilizando os algoritmos RLNS e RLS na metodologia proposta e na metodologia clássica.

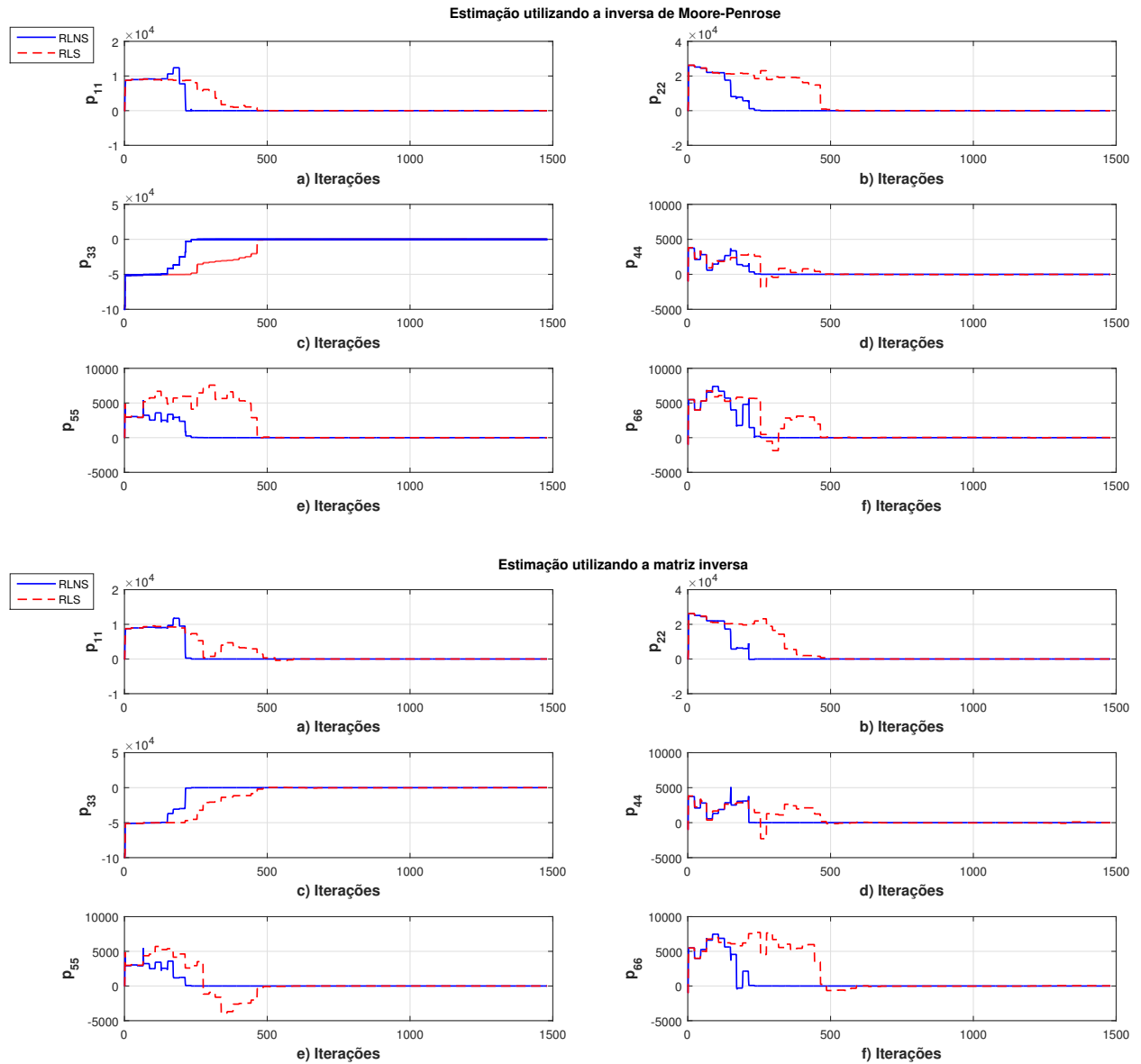


Figura 4.2: Os estimadores RLNS e RLS são utilizados para aproximar os parâmetros da diagonal principal da matriz P , para metodologia proposta e para metodologia clássica. Parâmetros são apresentados em função do erro, o valor de estado estacionário aqui assumido e inferior a 1%.

A Tabela 4.4 mostra a estimação do erro para os elementos fora da diagonal principal da matrix P . As estimações são feitas utilizando os filtros RLNS e RLS para metodologia clássica e para metodologia proposta.

Tabela 4.4: Número de iterações para o RLNS e RLS atingir o valor de estado estacionário com um erro inferior a 5%, para metodologia clássica e para metodologia proposta.

Erro ≤ 5	Matriz inversa		Matriz inversa generalizada	
	Número de iterações			
	RLS	RLNS	RLS	RLNS
p_{12}	1286	470	952	256
p_{13}	1862	494	903	238
p_{14}	1736	380	1227	275
p_{15}	1638	457	1076	258
p_{16}	1756	427	986	294
p_{23}	1719	574	752	304
p_{24}	-	430	-	267
p_{25}	1776	376	987	352
p_{26}	1694	420	1054	239
p_{34}	1869	434	1045	279
p_{35}	-	594	-	286
p_{36}	1809	464	1154	266
p_{45}	1892	494	953	359
p_{46}	1784	367	1065	261
p_{56}	1802	472	954	358

A Tabela 4.4 mostra que para os elementos fora da diagonal principal da matriz P o número de iterações que o RLNS leva para atingem o valor de estado estacionário é inferior ao RLS para ambas as metodologias, caracterizando uma maior velocidade de convergência do RLNS quando comparado ao RLS, com exceção dos elementos p_{24} e p_{35} que não atingiram o valor de estado estacionário para ambas as metodologias.

As Tabelas 4.3, e 4.4 mostram que a velocidade de convergência da metodologia proposta (matriz inversa generalizada) foi superior em ate 30% quando comparada com a da metodologia convencional (matriz inversa). Mostraram também, que o RLNS foi mais eficiente na estimação dos parâmetros da matriz P quando comparados com o RLS. Para alguns casos a velocidade de convergência foi até 70% mais rápida para o RLNS do que para o RLS, proporcionando erros menores que 1% para os elementos da diagonal principal e menor que 5% para os elementos fora da diagonal principal.

A Fig.4.3 ilustra o comportamento da estimação do erro para os elementos da fora da diagonal principal da matrix P . As estimações são feitas utilizando os algoritmos RLNS e RLS na metodologia proposta e na metodologia clássica.

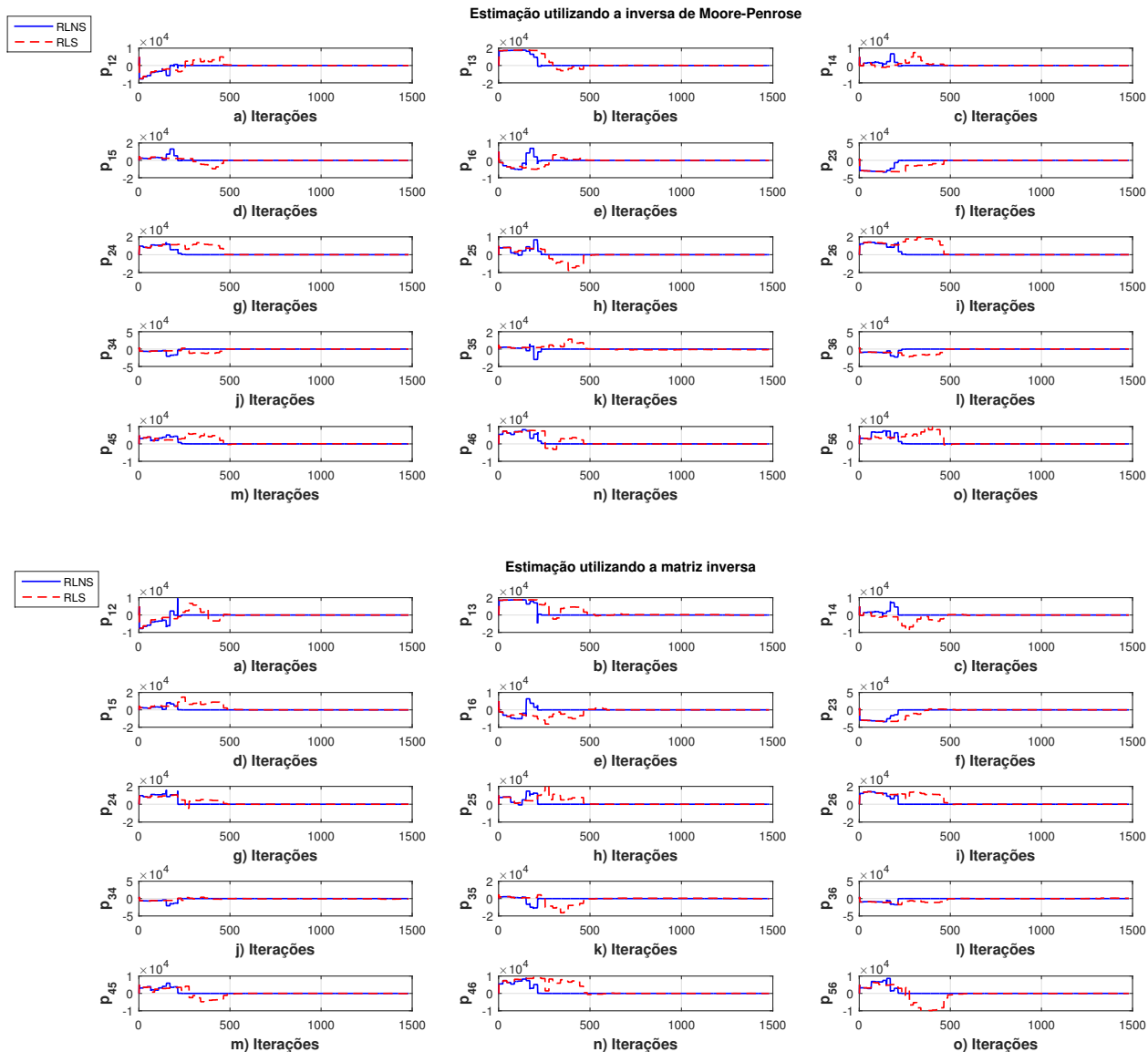


Figura 4.3: Os estimadores RLNS e RLS são utilizados para aproximar os parâmetros fora da diagonal principal da matrix P , para metodologia proposta e para metodologia clássica. Parâmetros são apresentados em função do erro, o valor de estado estacionário aqui assumido e inferior a 5%.

Além da velocidade de convergência apresentada nas Tabelas 4.3 e 4.4 as Figs.4.2 e 4.3 apresentam à metodologia proposta mais robusta a instabilidades numéricas. Por exemplo, o elemento p_{44} em que a estimação feita pelo RLNS para matriz inversa apresenta instabilidade numérica em todo o processo, por outro lado, devido as instabilidades numéricas o RLS não atinge o regime permanente. Já para a estimação feita pela inversa generalizada, o RLNS além de convergir 19% mais rápido, apresenta também uma menor sensibilidade as instabilidades numéricas, já o RLS, apresenta robustez as variações numéricas além de atingir a convergência.

As matriz de ganho da realimentação de estado utilizada como referência e dada por

$$K_{Schur} = \begin{bmatrix} 0.6407 & 1.3945 & 0 & 0 & 0 & 0 \\ -1.3945 & 0.6407 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.9992 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1497 & 0.0907 & 0.0169 \\ 0 & 0 & 0 & 0.2915 & 0.2214 & -0.0082 \\ 0 & 0 & 0 & -0.3225 & -0.4150 & 0.2441 \end{bmatrix}. \quad (4.7)$$

As matriz de ganho da realimentação de estado utilizando a matriz inversa para os estimadores RLS e RLNS são dadas por

$$K_{RLS} = \begin{bmatrix} -0.6399 & -1.3938 & 0.0000 & 0.0021 & -0.0035 & -0.0001 \\ 1.3942 & -0.6410 & -0.0000 & -0.0011 & 0.0017 & 0.0001 \\ 0.0000 & 0.0000 & -0.9992 & 0.0000 & -0.0001 & -0.0000 \\ 0.0135 & 0.0107 & 0.0001 & -0.1636 & -0.1034 & -0.0147 \\ -0.0105 & -0.0086 & -0.0001 & -0.2952 & -0.1981 & 0.0077 \\ -0.0008 & -0.0005 & -0.0000 & 0.3259 & 0.4133 & -0.2444 \end{bmatrix}. \quad (4.8)$$

$$K_{RLNS} = \begin{bmatrix} -0.6407 & -1.3946 & -0.0000 & -0.0004 & 0.0005 & 0.0000 \\ 1.3944 & -0.6407 & -0.0000 & 0.0001 & -0.0001 & -0.0000 \\ 0.0000 & -0.0000 & -0.9992 & 0.0000 & -0.0000 & -0.0000 \\ -0.0018 & -0.0022 & 0.0000 & -0.1651 & -0.0724 & -0.0158 \\ 0.0010 & 0.0014 & -0.0000 & -0.2819 & -0.2328 & 0.0076 \\ 0.0002 & 0.0002 & -0.0000 & 0.3238 & 0.4134 & -0.2442 \end{bmatrix}. \quad (4.9)$$

As matriz de ganho da realimentação de estado utilizando a matriz inversa generalizada para os estimadores RLS e RLNS são dadas por

$$K_{RLS} = \begin{bmatrix} -0.6412 & -1.3943 & 0.0000 & -0.0013 & 0.0017 & 0.0001 \\ 1.3956 & -0.6407 & -0.0000 & 0.0011 & -0.0012 & -0.0001 \\ 0.0000 & 0.0000 & -0.9992 & 0.0002 & -0.0002 & -0.0000 \\ -0.0096 & -0.0051 & 0.0005 & -0.1060 & -0.1180 & -0.0207 \\ 0.0061 & 0.0037 & -0.0003 & -0.3074 & -0.2154 & 0.0098 \\ 0.0008 & 0.0003 & -0.0000 & 0.3168 & 0.4192 & -0.2436 \end{bmatrix}. \quad (4.10)$$

$$K_{RLNS} = \begin{bmatrix} -0.6407 & -1.3945 & 0.0000 & 0.0000 & -0.0000 & -0.0000 \\ 1.3945 & -0.6407 & 0.0000 & 0.0001 & -0.0001 & -0.0000 \\ -0.0000 & 0.0000 & -0.9992 & -0.0000 & 0.0000 & 0.0000 \\ -0.0004 & 0.0002 & -0.0000 & -0.1501 & -0.0900 & -0.0168 \\ 0.0003 & -0.0001 & 0.0000 & -0.2912 & -0.2219 & 0.0082 \\ 0.0000 & -0.0000 & 0.0000 & 0.3225 & 0.4149 & -0.2441 \end{bmatrix}. \quad (4.11)$$

4.3 Considerações Finais

Neste capítulo foi apresentado a solução da DARE, no contexto do modelo espaço de estado do gerador eólico. Para isso, utilizamos os passos da metodologia proposta (matriz inversa generalizada) e comparamos com a metodologia convencional (matriz inversa).

Os resultados apresentados neste capítulo mostram que a velocidade de convergência da metodologia proposta (matriz inversa generalizada) foi superior em até 30% quando comparada com a da metodologia convencional (matriz inversa), mostram também que o algoritmo proposto RLNS tem velocidade de convergência de até 70% mais rápido que o RLS. Proporcionando que a matriz de ganho da realimentação de estado atinja valores próximos aos valores de referência, garantindo assim o controle ótimo do gerador eólico.

Conclusão e Perspectivas

Neste dissertação, foi apresentado uma metodologia para solucionar de forma geral a equação algébrica de Riccati em sua forma discreta (DARE). A metodologia foi avaliada em um sistema multivariável de 6 ordem, o gerador eólico.

Para melhor compreensão, os resultados podem ser comparados da seguinte maneira

- Comparação entre os estimadores RLS e RLNS

Pode-se comparar os estimadores RLS e RLNS na metodologias proposta (matriz inversa generalizada) e na metodologia convencional (matriz inversa). Para ambas metodologias o estimado RLNS se mostrou mais eficiente na estimação dos parâmetros das matrizes P e K .

- Comparação entre as metodologias

A metodologia é avaliada, comparando o estimador RLS na metodologia convencional com o RLS na metodologia proposta, comparação similar pode ser realizada com o estimador RLNS. Para ambos os estimadores a metodologia proposta mostrou-se mais eficiente na estimação dos parâmetros das matrizes P e K .

Os resultados mostram estimativas mais precisas e menor instabilidade numérica para a metodologia proposta quando comparada com a da metodologia convencional, como consequência, a velocidade de convergência da metodologia proposta é superior em até 30% quando comparada com a da metodologia convencional, e o algoritmo RLNS tem velocidade de convergência de até 70% mais rápido que o RLS. Os resultados ainda apresentam, tanto para metodologia proposta quanto para metodologia convencional, que a matriz de ganho da realimentação de estado estimada pelo RLNS mais próximo ao valor de referência quando comparados ao RLS.

Portanto, com base nos resultados aqui apresentados, a forma mais eficiente de garantir a otimalidade dos controladores no sistema do gerador eólico, é utilizar o RLNS como estimador de parâmetros e a matriz inversa generalizada como forma de condicionar numericamente o sistema, o que por consequência garantirá o controle ótimo do gerador eólico.

No entanto, o cálculo da matriz inversa generalizada, e o cálculo do estimador RLNS utilizam um número maior de operações do que as técnicas clássicas, aumentando assim o esforço computacional. Portanto, nossos próximos trabalhos é mensurar e avaliar o esforço computacional da metodologia proposta.

Outros trabalhos poderão ser desenvolvidos baseado na metodologia proposta. Por exemplo,

- Estimção da matriz P

Ao invés de utilizar o RLS e/ou RLNS para estimar a matriz P . Poderiam ser utilizadas técnicas de lógica fuzzy, algoritmos genético, redes neuronais, ou ainda, poderia ser desenvolvido uma função de custo convexa que não possua mínimos locais somente mínimo global.

- Estabilidade numérica para a DARE

Garantir a estabilidade numérica para DARE não por uma inversa generalizada, mas sim, por outro tipo de decomposição, como a QR , LU , LU^{td} entre outras.

Publicações

Os artigos relacionados a seguir são referentes aos estudos descritos nesse trabalho:

Congresso:

- QUEIROZ, JONATHAN ARAUJO; NETO, JOAO VIANA DA F. ; BARROS, ALLAN KARDEC . Solution of Algebraic Riccati Equation for optimal control using Non-Square estimator. In: 2015 IEEE 24th International Symposium on Industrial Electronics (ISIE), 2015, Buzios. 2015 IEEE 24th International Symposium on Industrial Electronics (ISIE). p. 125.
- QUEIROZ, JONATHAN ARAUJO; BARROS, ALLAN KARDEC ; NETO, JOAO VIANA DA F. ; SANTANA, EWALDO . Optimal control of a wind generator system using non-squares estimators. In: 2015 IEEE 24th International Symposium on Industrial Electronics (ISIE), 2015, Buzios. 2015 IEEE 24th International Symposium on Industrial Electronics (ISIE). p. 1452.
- J. A. Queiroz, P. H. M. Rêgo, J. a. V. F. Neto, C. d. Silva, E. Santana, and A. K. Barros, **Convergence analysis using non-squares estimators to approximate the solution of hjb-riccati equation for the design dlqr via hdp**, in Proceedings of the 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, ser. UKSIM 14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 63-68. [Online]. Available: <http://dx.doi.org/10.1109/UKSim.2014.107>
- J. A. Queiroz, P. H. Rego, J. V. Neto, C. d. Silva, E. Santana, and A. K. Barros, **Estimators based on nonsquares loss functions to approximate hjb-riccati equation solution for dlqr design via hdp**, in Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on, Oct 2013, pp. 3238-3243. Available: 10.1109/SMC.2013.552
- W. R. M. Santos, J. A. Queiroz, J. a. V. d. F. Neto, P. H. M. Rêgo, E. Santana, and G. Andrade, **Rls algorithms and convergence analysis method for online dlqr control design via heuristic dynamic programming**, in Proceedings of the 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, ser.

UKSIM 14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 77-83. [Online]. Available: <http://dx.doi.org/10.1109/UKSim.2014.109>

Recebemos o convite para publicar a versão estendida do artigo intitulado **Solution of Algebraic Riccati Equation for optimal control using Non-Square** nos seguintes jornais:

- Journal of Electrical Engineering (ISSN: 2328-2223)
- Journal of Control Science and Engineering (ISSN: 2328-2231)

Decomposições

B.1 Decomposição em valores e vetores singulares (SVD)

Para toda matriz $A \in \mathbb{R}^{m \times n}$ com $\text{posto}(A)$ igual a r , existem então números $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$, os valores singulares de A , uma matriz ortogonal $U^{m \times m}$, $U = [u_1, \dots, u_m]$ é uma matriz ortogonal $V^{n \times n}$, $V = [v_1, \dots, v_n]$, onde u e v são vetores colunas, tais que são válidas as seguintes relações:

$$A = U\Sigma V^T$$

$$\Sigma = U^T A V$$

onde Σ é uma matriz $n \times n$ dada por

$$\Sigma = \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} \quad (\text{B.1})$$

onde $D^{r \times r}$ é uma matriz diagonal com o i -ésimo elemento da diagonal $d_{ii} = \sigma > 0$ para $1 \leq i \leq r$. Além disso, $U \in \mathbb{R}^{m \times m}$ é uma matriz ortogonal cujas colunas são os vetores característicos de $A.A^T$, $V \in \mathbb{R}^{n \times n}$ é uma matriz ortogonal cujas colunas são os vetores característicos de $A^T.A$, $\Sigma \in \mathbb{R}^{m \times n}$ é uma matriz diagonal contendo a raiz quadrada dos valores característicos de $A.A^T$ (que são equivalentes aos valores característicos de $A^T.A$), arranjados em ordem decrescente. Os vetores característicos de $A.A^T$ e $A^T.A$ estão arranjados nas colunas de U e V , respectivamente, na ordem de seus valores característicos na matriz Σ .

Os elementos, σ_i , da diagonal de Σ são denominados de valores singulares de A , sendo todos não-negativos. Além disso, o número de valores singulares positivos é igual ao $\text{rank}(A)$. Os vetores colunas de U são denominados de vetores singulares à esquerda de A e os vetores colunas de V são denominados de vetores singulares à direita de A , e as relações entre estes vetores são:

$$A.v_i = \sigma_i u_i$$

e

$$A^T \cdot u_i = \sigma_i v_i$$

A matriz:

$$A^+ = V\Sigma^+U^T = (A^T \cdot A)^{-1} \cdot A^T$$

é chamada de pseudo-inversa de A , onde os elementos da diagonal de Σ^+ consistem os valores singulares positivos de Σ , na mesma ordem.

B.2 Decomposição de Schur

O teorema que apresentamos abaixo, devido a Schur, é semelhante, mas não idêntico, ao Teorema de Jordan: toda matriz de $\text{Mat}(\mathbf{C}, n)$ pode ser levada por uma transformação de similaridade induzida por uma matriz unitária a uma matriz triangular superior. Esse teorema é alternativamente denominado Teorema da Triangularização de Schur ou Teorema da Decomposição de Schur.

Teorema da Decomposição de Schur: Seja $A \in \mathbf{C}^{n \times n}$. Então, existe $U \in \mathbf{C}^{n \times n}$, unitária, e $S \in \mathbf{C}^{n \times n}$, triangular superior, tais que $A = U^* S U$.

Demonstração (1):

Provemos o teorema usando a indução matemática. O teorema é trivialmente verdadeiro para $n = 1$, usando $U = I$.

Suponhamos agora que o resultado é válido para qualquer matriz de ordem menor do que n e mostremos que também é válido para matrizes de ordem n .

Seja u um vetor próprio de A associado ao vetor próprio λ , $Au = \lambda u$, $u \neq 0$.

Supondo que u é um vetor unitário, caso contrário normaliza-se o vetor dividindo pela sua norma. É sempre possível selecionar uma matriz V de dimensão $n \times (n - 1)$ tal que $U_1 = (uV) = (u, v_1, v_2, \dots, v_{n-1})$ seja matriz unitária. Neste sentido escolhemos $n - 1$ vetores da base canônica de \mathbf{C}^n que, conjuntamente com u , formem um conjunto de vetores linearmente independentes.

Em seguida utiliza-se o processo de Gram-Schmidt para ortonormalizar as colunas de V que são constituídas por estes últimos $n - 1$ vetores. Sabendo que $AU_1 = A(uV) = (AuAV) = (\lambda uAV)$ então

$$U_1^* A U_1 = \begin{bmatrix} u^* \\ v^* \end{bmatrix} \begin{bmatrix} \lambda u & AV \end{bmatrix} = \begin{bmatrix} u^* \lambda u & u^* \lambda V \\ V^* \lambda u & V^* \lambda V \end{bmatrix},$$

como U_1 é formado e formado por colunas de vetores ortonormais tem-se que

$$u^* \lambda u = \lambda(u^* u) = \lambda$$

e

$$V^*\lambda u = \lambda(V^*u) = \lambda \begin{bmatrix} v_1^*u \\ v_2^*u \\ \vdots \\ v_{n-1}^*u \end{bmatrix} \quad u = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix},$$

Logo

$$U_1^*AU_1 = \begin{bmatrix} \lambda & \times & \cdots & \times \\ 0 & & & \\ \vdots & & B & \\ 0 & & & \end{bmatrix}$$

onde $B = V^*AV$.

Assim, por hipótese existe uma matriz unitária V_1 tal que $V_1^*BV_1 = T$ é uma matriz triangular superior.

seja $U = U_1 \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix}$ uma matriz unitária pois V_1 também é unitaria e como o produto de duas matrizes unitarias também é unitaria, tem-se

$$U^*AU = \left(U_1 \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix} \right)^* AU_1 \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix}^* U_1^*AU_1 \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix}$$

Como

$$U_1^*AU_1 = \begin{bmatrix} \lambda & u^*AV \\ 0 & B \end{bmatrix},$$

então

$$\begin{aligned} U^*AU &= \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix}^* U_1^*AU_1 \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix}^* \begin{bmatrix} \lambda & u^*AV \\ 0 & B \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix} = \\ &= \begin{bmatrix} 1 & 0 \\ 0 & V_1^* \end{bmatrix} \begin{bmatrix} \lambda & u^*AVV_1 \\ 0 & BV_1 \end{bmatrix} = \begin{bmatrix} \lambda & u^*AVV_1 \\ 0 & V_1^*BV_1 \end{bmatrix} \end{aligned}$$

Mas $V_1^*BV_1$ é a matriz triangular superior T_1 , assim $U^*AU = \begin{bmatrix} \lambda & u^*AVV_1 \\ 0 & T_1 \end{bmatrix} = T$, é uma matriz triangular superior. ■

Demonstração (2):

Começemos observando que se $A = U^*SU$ com U unitário e S triangular superior.

Seja $n \geq 2$ e v_i um autovetor de A com autovalores λ_i e $\|v_1\| = 1$. Seja $U^{(1)}$ uma matriz unitária da forma $U^{(1)} = [u_1^{(1)}, u_2^{(1)}, \dots, u_n^{(1)}]$ com $u_1^{(1)}$, ou seja, a primeira coluna é o vetor de v_1 . Então,

$$U^{(1)*}AU^{(1)} = \begin{pmatrix} \lambda_1 & b^{(1)T} \\ 0_{n-1} & A^{(1)} \end{pmatrix}$$

onde

$$b^{(1)} = \begin{pmatrix} b_1^{(1)} \\ \vdots \\ b_{n-1}^{(1)} \end{pmatrix}, 0_{n-1} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}, A^{(1)} = \begin{pmatrix} a_{11}^{(1)} & \cdots & a_{1(n-1)}^{(1)} \\ \vdots & \ddots & \vdots \\ a_{(n-1)1}^{(1)} & \cdots & a_{(n-1)(n-1)}^{(1)} \end{pmatrix}$$

Supondo que a afirmação é válida para matrizes $(n-1) \times (n-1)$, então existe uma matriz unitária $V \in \mathbf{C}^{n-1}$ tal que $V^* A^{(1)} V = S^{(1)}$ triangular superior. Assim, definindo a matriz $U^{(2)} \in \mathbf{C}^{n \times n}$ por

$$U^{(2)} = \begin{pmatrix} 1 & 0_{n-1}^T \\ 0_{n-1} & V \end{pmatrix}$$

teremos

$$\begin{aligned} (U^{(1)} U^{(2)})^* A U^{(1)} U^{(2)} &= U^{(2)*} U^{(1)*} A U^{(1)} U^{(2)} \\ &= \begin{pmatrix} 1 & 0_{n-1}^T \\ 0_{n-1} & V \end{pmatrix} \begin{pmatrix} \lambda_1 & b^{(1)T} \\ 0_{n-1} & A^{(1)} \end{pmatrix} \begin{pmatrix} 1 & 0_{n-1}^T \\ 0_{n-1} & V \end{pmatrix} \\ &= \begin{pmatrix} \lambda_1 & (V^T b^{(1)T})^T \\ 0_{n-1} & V^* A^{(1)} V \end{pmatrix} \\ &= \begin{pmatrix} \lambda_1 & (V^T b^{(1)T})^T \\ 0_{n-1} & S^{(1)} \end{pmatrix}, \end{aligned}$$

que é triangular superior, pois $S^{(1)}$ o é. Como $U^{(1)}$ e $U^{(2)}$ são matrizes unitárias, então $U^{(1)} U^{(2)}$ também é unitária, portanto o teorema está provado. ■

Referências Bibliográficas

- [1] P. Lancaster and L. Rodman, *Algebraic Riccati equations*, ser. Oxford science publications. Oxford: Clarendon Press, 1995. [Online]. Available: <http://opac.inria.fr/record=b1078964>
- [2] F. Lewis, D. Vrabie, and V. Syrmos, *Optimal Control*. Wiley, 2012. [Online]. Available: <http://books.google.com.br/books?id=FlFetwAACAAJ>
- [3] F. L. Lewis and D. Vrabie, “Reinforcement learning and adaptive dynamic programming for feedback control,” *Cir. and Sys. Mag.*, vol. 09, no. 3, pp. 32–50, Sep. 2009. [Online]. Available: <http://dx.doi.org/10.1109/MCAS.2009.933854>
- [4] V. Kucera, “The discrete riccati equation of optimal control,” *Kybernetika*, vol. 08, no. 5, pp. (430)–447, 1972. [Online]. Available: <http://eudml.org/doc/28850>
- [5] T. Pappas, A. Laub, and N. R. Sandell, “On the numerical solution of the discrete-time algebraic riccati equation,” *Automatic Control, IEEE Transactions on*, vol. 25, no. 4, pp. 631–641, Aug 1980.
- [6] G. H. Golub and C. F. Van Loan, *Matrix Computations (4rd Ed.)*. Baltimore, MD, USA: Johns Hopkins University Press, 2013.
- [7] Y. Zhu, “Fuzzy optimal control for multistage fuzzy systems,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 41, no. 4, pp. 964–975, Aug 2011.
- [8] B. M. Al-Hadithi, A. Jiménez, and R. G. Lopez, “Fuzzy optimal control using generalized takagi-sugeno model for multivariable nonlinear systems,” *Applied Soft Computing*, vol. 30, no. 0, pp. 205 – 213, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1568494615000848>
- [9] “Modelos e convergência de um algoritmo genético para alocação de auto-estrutura via rlq,” *IEEE LATIN AMERICA TRANSACTIONS*, vol. 6, no. 1, pp. 1 – 9, 2008.
- [10] F. N. da Silva, *Métodos Neurais para a Solução da Equação Algébrica de Riccati e o LQR*.

- [11] D. Liu, D. Wang, D. Zhao, Q. Wei, and N. Jin, “Neural-network-based optimal control for a class of unknown discrete-time nonlinear systems using globalized dual heuristic programming,” *Automation Science and Engineering, IEEE Transactions on*, vol. 9, no. 3, pp. 628–634, July 2012.
- [12] C.-M. Hong, F.-S. Cheng, and C.-H. Chen, “Optimal control for variable-speed wind generation systems using general regression neural network,” *International Journal of Electrical Power and Energy Systems*, vol. 60, no. 0, pp. 14 – 23, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0142061514000787>
- [13] E. Walach and B. Widrow, “The least mean fourth (lmf) adaptive algorithm and its family,” *IEEE Trans. Inf. Theor.*, vol. 30, no. 2, pp. 275–283, Sep. 2006.
- [14] C. Da Silva, E. Santana, E. Aguiar, M. A. F. De Araújo, and A. K. Barros, “An adaptive recursive algorithm based on non-quadratic function of the error,” *Signal Process.*, vol. 92, no. 4, pp. 853–856, Apr. 2012.
- [15] C. da Silva, A. K. Barros, E. Santana, M. A. de Araújo, M. V. de S. Lopes, J. V. Fonseca, and J. C. Principe, “An algorithm based on non-squared sum of the errors,” *Signal Processing*, no. 0, pp. –, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0165168415001139>
- [16] J. A. Queiroz, P. H. Rego, J. V. Neto, C. d. Silva, E. Santana, and A. K. Barros, “Estimators based on non-squares loss functions to approximate hjb-riccati equation solution for dlqr design via hdp,” in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, Oct 2013, pp. 3238–3243.
- [17] J. A. Queiroz, P. H. M. Rêgo, J. a. V. F. Neto, C. d. Silva, E. Santana, and A. K. Barros, “Convergence analysis using non-squares estimators to approximate the solution of hjb-riccati equation for the design dlqr via hdp,” in *Proceedings of the 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, ser. UKSIM '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 63–68. [Online]. Available: <http://dx.doi.org/10.1109/UKSim.2014.107>
- [18] J. Araujo Queiroz, A. Kardec Barros, J. Viana da F. Neto, and E. Santana, “Optimal control of a wind generator system using non-squares estimators,” in *Industrial Electronics (ISIE), 2015 IEEE 24th International Symposium on*, June 2015, pp. 1452–1457.
- [19] J. Araujo Queiroz, J. Viana da F. Neto, and A. Kardec Barros, “Solution of algebraic riccati equation for optimal control using non-square estimator,” in *Industrial Electronics (ISIE), 2015 IEEE 24th International Symposium on*, June 2015, pp. 125–130.
- [20] E. H. Moore, “On the reciprocal of the general algebraic matrix,” *Bulletin of the American Mathematical Society*, vol. 26, pp. 394–395, 1920.

- [21] R. Penrose, “A generalized inverse for matrices,” *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 51, no. 03, pp. 406–413, 1955. [Online]. Available: <http://dx.doi.org/10.1017/s0305004100030401>
- [22] C. da Silva, A. K. Barros, E. Santana, M. A. de Araújo, M. V. de S. Lopes, J. V. Fonseca, and J. C. Principe, “An algorithm based on non-squared sum of the errors,” *Signal Processing*, vol. 117, no. 0, pp. 188 – 191, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0165168415001139>
- [23] R. E. Kalman. (1960) Contributions to the Theory of Optimal Control. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.26.4070>
- [24] P. Dorato and A. H. Levis, “Optimal linear regulators: The discrete-time case,” *Automatic Control, IEEE Transactions on*, vol. 16, no. 6, pp. 613–620, Dec 1971.
- [25] A. Ferrante and L. Ntogramatzidis, “The generalized continuous algebraic riccati equation and impulse-free continuous-time {LQ} optimal control,” *Automatica*, vol. 50, no. 4, pp. 1176 – 1180, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0005109814000521>
- [26] A. S. A. Dilip and H. K. Pillai, “Yet another characterization of solutions of the algebraic riccati equation,” *Linear Algebra and its Applications*, vol. 481, no. 0, pp. 1 – 35, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0024379515002773>
- [27] C. Coumarbatch and Z. Gajic, “Exact decomposition of the algebraic riccati equation of deterministic multimodeling optimal control problems,” *Automatic Control, IEEE Transactions on*, vol. 45, no. 4, pp. 790–794, Apr 2000.
- [28] C. Pearce, “On the solution of a class of algebraic matrix riccati equation,” *Automatic Control, IEEE Transactions on*, vol. 31, no. 3, pp. 252–255, Mar 1986.
- [29] D. Connolly, H. Lund, B. Mathiesen, and M. Leahy, “A review of computer tools for analysing the integration of renewable energy into various energy systems,” *Applied Energy*, vol. 87, no. 4, pp. 1059 – 1082, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0306261909004188>
- [30] F. Wu, X.-P. Zhang, P. Ju, and M. Sterling, “Decentralized nonlinear control of wind turbine with doubly fed induction generator,” *Power Systems, IEEE Transactions on*, vol. 23, no. 2, pp. 613 –621, may 2008.
- [31] L. Barros, W. Mota, J. da Silva, and C. Barros, “An optimal control strategy for dfig,” in *Industrial Technology (ICIT), 2010 IEEE International Conference on*, march 2010, pp. 1727 –1732.
- [32] J. Fang, G. Li, X. Liang, and M. Zhou, “An optimal control strategy for reactive power in wind farms consisting of vsfc dfig wind turbine generator systems,” in *Electric Utility Deregulation and Restructuring and Power Technologies (DRPT), 2011 4th International Conference on*, july 2011, pp. 1709 –1715.

-
- [33] V. P. Pinto, J. C. T. Campos, L. L. N. Dos Reis, C. B. Jacobina, and N. Rocha, “Robustness and performance analysis for the linear quadratic gaussian/loop transfer recovery with integral action controller applied to doubly fed induction generators in wind energy conversion systems,” *Electric Power Components and Systems*, vol. 40, no. 2, pp. 131–146, 2011. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/15325008.2011.629331>
- [34] C. hua Guo, Alan, and J. Laub, “A schur method for solving algebraic riccati equations,” *IEEE Trans. Auto. Control*, pp. 913–921, 1979.
- [35] K. Ogata, *Modern Control Engineering*, 4th ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.