

**UNIVERSIDADE FEDERAL DO MARANHÃO**

**CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA**

**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE**

**REDES NEURONAIS ARTIFICIAIS ESTRUTURADAS  
PARA ALOCAÇÃO ONLINE DE AUTOESTRUTURAS**

**BRUNO FRANÇA COELHO**

**ORIENTADOR: PROF. DR. JOÃO VIANA DA FONSECA NETO**

**ORIENTADORA: PROF<sup>A</sup>. DR<sup>A</sup>. PATRÍCIA HELENA M. RÊGO**

São Luís  
2019

**UNIVERSIDADE FEDERAL DO MARANHÃO**

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE

**REDES NEURONAIS ARTIFICIAIS ESTRUTURADAS  
PARA ALOCAÇÃO ONLINE DE AUTOESTRUTURAS**

**BRUNO FRANÇA COELHO**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão, como parte dos requisitos para a obtenção do título de Mestre em Engenharia de Eletricidade, área de concentração: Automação e Controle de Processos Industriais.

Orientador: Dr. João Viana da Fonseca Neto

Orientadora: Dr<sup>a</sup>. Patrícia Helena M. Rêgo

São Luís  
2019

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

Coelho, Bruno França.

Redes Neurais Artificiais Estruturadas para Alocação  
Online de Autoestruturas / Bruno França Coelho. - 2019.  
103 p.

Coorientador(a): Patrícia Helena Moraes Rêgo.

Orientador(a): João Viana da Fonseca Neto.

Dissertação (Mestrado) - Programa de Pós-graduação em  
Engenharia de Eletricidade/ccet, Universidade Federal do  
Maranhão, São Luís, 2019.

1. Alocação de Autoestrutura. 2. Online. 3. Redes  
Neurais Artificiais. I. Fonseca Neto, João Viana da.  
II. Rêgo, Patrícia Helena Moraes. III. Título.

# **UNIVERSIDADE FEDERAL DO MARANHÃO**

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE

## **REDES NEURONAIS ARTIFICIAIS ESTRUTURADAS PARA ALOCAÇÃO ONLINE DE AUTOESTRUTURAS**

**BRUNO FRANÇA COELHO**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão, como parte dos requisitos para a obtenção do título de Mestre em Engenharia de Eletricidade, área de concentração: Automação e Controle de Processos Industriais.

Aprovado em \_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_.

Membros da Banca:

---

Prof. Dr. João Viana da Fonseca Neto  
(Orientador – UFMA)

---

Prof. Dr. Francisco das Chagas de Sousa  
(UFMA)

---

Prof. Dr. Roberto Célio Limão de Oliveira  
(UFPA)

São Luís  
2019

*A meus pais, Antônio José e Maria das Neves, e minha irmã Sâmila Bruna, exemplos de vida e caráter, pelo apoio incondicional em todas as decisões de minha vida.*

# Agradecimentos

Agradeço primeiramente a Deus pelo dom da vida, da sabedoria e da saúde, sem Ele nada seria possível.

Ao meu professor e orientador João Viana pela orientação e dedicação aplicados neste trabalho. Meu agradecimento sincero pelo acompanhamento, pela compreensão, auxílio e tempo despendido durante todo curso, sempre me motivando em todos os momentos em que precisei e que me mostrou os caminhos a serem seguidos.

À Prof. Patrícia Helena, pelas trocas de experiências e valorosas contribuições para o desenvolvimento deste trabalho.

Aos meus Pais, Antônio José e Maria das Neves, e minha irmã Sâmila Bruna pelo incentivo, apoio e compreensão ao longo destes dois anos.

À Nívea Raquel, pela sua compreensão nos momentos que me ausentei, pelo companheirismo e pelo apoio.

Aos meus companheiros do LABSECI, Ana Caroline, Daniel Trovão, Ernesto Franklin, Evandro Martins, Geiza Mendonça, José Pinheiro e Wenllson Carlos, pela ajuda, por todos os ensinamentos que compartilhamos e momentos de descontração.

Aos meus amigos Daniel Meireles, Naylson Silva e Frederico Ivo pelo companheirismo e amizade.

Aos membros da banca examinadora pelos comentários, sugestões e contribuições, que ajudaram a melhorar a qualidade e a redação final do manuscrito.

A todos os professores e colaboradores do Programa de Pós-Graduação em Engenharia de Eletricidade e da Universidade Federal do Maranhão, que de alguma forma contribuíram com o desenvolvimento deste trabalho e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo suporte financeiro.

*Ora, se algum de vós tem falta de sabedoria, peça-a a Deus, que a todos dá liberalmente e não censura, e ser-lhe-á dada. Peça-a, porém, com fé, não duvidando; pois aquele que duvida é semelhantemente à onda do mar, que é sublevada e agitada pelo vento. Não pense tal homem que receberá do Senhor alguma coisa, homem vacilante que é, e inconsistente em todos os seus caminhos.*

*(Tiago 1:5-8)*

# Resumo

Para melhorar o desempenho da estabilidade e a forma de resposta dos sistemas MIMO, o controle dos modos é relevante para impor as especificações do projeto. Um método bioinspirado para alocação online da autoestrutura para o projeto de sistemas de controle multivariáveis é apresentado neste documento. O método é baseado em redes neurais artificiais para a execução da lei de controle e para o treinamento dos ganhos do controlador por realimentação de estados. Com o objetivo de sintonizar online os controladores de espaço de estados, a matriz de ganho que satisfaz todas as especificações de projeto, é apresentada uma proposta para calcular as matrizes de ganho que melhor atendem a uma determinada faixa operacional de sistemas dinâmicos MIMO. A proposta é avaliada em dois sistemas multivariáveis, o primeiro é um modelo matemático que representa um circuito RLC de quarta ordem com duas tensões de entrada e dois níveis de tensão controláveis, e o segundo modelo representa o controle direcional-lateral de uma aeronave F-16.

**Palavras-chave:** Alocação de autoestrutura, online, redes neurais artificiais.



# Abstract

To improve the stability performance and the response form of MIMO systems, control of the modes is relevant to impose the design specifications. A bioinspired method for online eigenstructure assignment for the design of multivariable control systems is presented in this paper. The method is based on artificial neural networks for the execution of the control law and for training the controller gains by state feedback. Aiming online tuning of state space controllers, the gain matrix that satisfies all the design specifications, a proposal is presented to compute the gain matrices that best meet a given operating range of MIMO dynamic systems. The proposal is evaluated in two multivariate systems, the first is a mathematical model that represents a fourth-order RLC circuit with two input voltages and two controllable voltage levels, and second model represents the lateral-directional control of an F-16 aircraft.

**Keywords:** Eigenstructure assignment, online, artificial neural networks.

# Lista de ilustrações

Figura 1 –	Projeção do autovetor desejado. . . . .	28
Figura 2 –	Resposta em Frequência desejável para um sistema multivariável (PINTO, 2012). . . . .	38
Figura 3 –	Estrutura generalizada de um neurônio. (a) Neurônio Natural (ARBIB, 2002). (b) Neurônio artificial (HAYKIN et al., 2009). . . . .	41
Figura 4 –	Diagrama de blocos para realizar o cálculo dos mínimos quadrados usando uma abordagem de neurocomputação para resolver o sistema de equações lineares (HAM; KOSTANIC, 2001). . . . .	47
Figura 5 –	Diagrama de blocos do sistema em espaço de estados, em malha fechada com a rede neuronal. . . . .	49
Figura 6 –	Arquitetura do Dispositivo de Controle da RNA-E para Alocação de Autoestruturas. . . . .	50
Figura 7 –	Diagrama de blocos da rede neuronal para o cálculo do matriz de ganhos $W$ . . . . .	52
Figura 8 –	Diagrama de blocos da rede neuronal para o cálculo do matriz de ganhos $K$ . . . . .	53
Figura 9 –	Metodologia aplicada aos experimentos computacionais. . . . .	57
Figura 10 –	Diagrama esquemático do circuito elétrico de 4ª. . . . .	57
Figura 11 –	Evolução do processo iterativo das colunas da matriz $W_{rna}$ com taxa de aprendizado $\mu = 0,6$ . Algoritmo $EA-RNA-Core-RNA-E_W$ . . . . .	61
Figura 12 –	Evolução do processo iterativo das colunas da matriz $K_{rna}$ com taxa de aprendizado $\mu = 0,6$ . Algoritmo $EA-RNA-Core-RNA-E_K$ . . . . .	61
Figura 13 –	Evolução do processo iterativo das colunas da matriz $W_{rna}$ com taxa de aprendizado $\mu = 0,7$ . Algoritmo $EA-RNA-Core-RNA-E_W$ . . . . .	62
Figura 14 –	Evolução do processo iterativo das colunas da matriz $K_{rna}$ com taxa de aprendizado $\mu = 0,7$ . Algoritmo $EA-RNA-Core-RNA-E_K$ . . . . .	63
Figura 15 –	Evolução do processo iterativo das colunas da matriz $W_{rna}$ com taxa de aprendizado $\mu = 0,8$ . Algoritmo $EA-RNA-Core-RNA-E_W$ . . . . .	64
Figura 16 –	Evolução do processo iterativo das colunas da matriz $K_{rna}$ com taxa de aprendizado $\mu = 0,8$ . Algoritmo $EA-RNA-Core-RNA-E_K$ . . . . .	64
Figura 17 –	Evolução do processo iterativo das colunas da matriz $W_{rna}$ com taxa de aprendizado $\mu = 0,9$ . Algoritmo $EA-RNA-Core-RNA-E_W$ . . . . .	65
Figura 18 –	Evolução do processo iterativo das colunas da matriz $K_{rna}$ com taxa de aprendizado $\mu = 0,9$ . Algoritmo $EA-RNA-Core-RNA-E_K$ . . . . .	66
Figura 19 –	Evolução do processo iterativo das colunas da matriz $K_{rna}$ com taxa de aprendizado de $\beta$ . . . . .	68

Figura 20 – Gráfico dos valores singulares do sistema em malha aberta e em malha fechada com a matriz de ganhos $K$ . . . . .	69
Figura 21 – SVD . . . . .	70
Figura 22 – Gráfico Valores Singulares sistemas . . . . .	70
Figura 23 – Evolução do processo iterativo das colunas da matriz $W_rna$ com taxa de aprendizado $\mu = 0,001$ . Algoritmo $EA-RNA-Core-RNA-E_W$ . . . . .	77
Figura 24 – Evolução do processo iterativo das colunas da matriz $K_rna$ com taxa de aprendizado $\mu = 0,4$ . Algoritmo $EA-RNA-Core-RNA-E_K$ . . . . .	78
Figura 25 – Evolução do processo iterativo das colunas da matriz $W_rna$ com taxa de aprendizado $\mu = 0,002$ . Algoritmo $EA-RNA-Core-RNA-E_W$ . . . . .	79
Figura 26 – Evolução do processo iterativo das colunas da matriz $K_rna$ com taxa de aprendizado $\mu = 0,5$ . Algoritmo $EA-RNA-Core-RNA-E_K$ . . . . .	79
Figura 27 – Evolução do processo iterativo das colunas da matriz $W_rna$ com taxa de aprendizado $\mu = 0,003$ . Algoritmo $EA-RNA-Core-RNA-E_W$ . . . . .	80
Figura 28 – Evolução do processo iterativo das colunas da matriz $K_rna$ com taxa de aprendizado $\mu = 0,6$ . Algoritmo $EA-RNA-Core-RNA-E_K$ . . . . .	81
Figura 29 – Evolução do processo iterativo das colunas da matriz $W_rna$ com taxa de aprendizado $\mu = 0,005$ . Algoritmo $EA-RNA-Core-RNA-E_W$ . . . . .	82
Figura 30 – Evolução do processo iterativo das colunas da matriz $K_rna$ com taxa de aprendizado $\mu = 0,7$ . Algoritmo $EA-RNA-Core-RNA-E_K$ . . . . .	82
Figura 31 – Evolução do processo iterativo das colunas da matriz $K_rna$ com taxa de aprendizado $\beta$ . Algoritmo $EA-RNA-Core-RNA-E_K$ . . . . .	84
Figura 32 – Gráfico dos Valores Singulares dos Sistemas . . . . .	85
Figura 33 – Gráfico dos valores Singulares dos Sistemas . . . . .	86

# Lista de tabelas

Tabela 1 – Matrizes $W_{rna}$ obtidas através da mudança da taxa de aprendizado. . .	66
Tabela 2 – Matrizes $K_{rna}$ de ganhos obtidos através da mudança da taxa de aprendizado. . . . .	67
Tabela 3 – Valores singulares máximos e mínimos dos sistemas em malha fechada.	69
Tabela 4 – Autovalores do circuito elétrico. . . . .	70
Tabela 5 – Autovetores do Circuito Elétrico associados aos Autovalores 1 e 2. . . .	71
Tabela 6 – Acoplamento CV do circuito elétrico - Sistema 0. . . . .	72
Tabela 7 – Acoplamento CV do circuito elétrico - Sistema 1. . . . .	72
Tabela 8 – Acoplamento CV do circuito elétrico - Sistema 2. . . . .	72
Tabela 9 – Acoplamento CV do circuito elétrico - Sistema 3. . . . .	72
Tabela 10 – Acoplamento CV do circuito elétrico - Sistema 4. . . . .	72
Tabela 11 – Acoplamento WB do circuito elétrico - Sistema 0. . . . .	73
Tabela 12 – Acoplamento WB do circuito elétrico - Sistema 1. . . . .	73
Tabela 13 – Acoplamento WB do circuito elétrico - Sistema 2. . . . .	73
Tabela 14 – Acoplamento WB do circuito elétrico - Sistema 3. . . . .	74
Tabela 15 – Acoplamento WB do circuito elétrico - Sistema 4. . . . .	74
Tabela 16 – Sensibilidades individuais dos autovalores. . . . .	74
Tabela 17 – Matrizes $W_{rna}$ obtidas através da mudança da taxa de aprendizado. . .	83
Tabela 18 – Matrizes $K_{rna}$ de ganhos obtidos através da mudança da taxa de aprendizado. . . . .	83
Tabela 19 – Valores singulares máximos e mínimos dos sistemas em malha fechada.	85
Tabela 20 – Autovalores do F-16. . . . .	87
Tabela 21 – Autovetores do F-16 associados aos Autovalores. . . . .	87
Tabela 22 – Acoplamento CV do F-16 - Sistema 0. . . . .	87
Tabela 23 – Acoplamento CV do F-16 - Sistema 1. . . . .	88
Tabela 24 – Acoplamento CV do F-16 - Sistema 2. . . . .	88
Tabela 25 – Acoplamento CV do F-16 - Sistema 3. . . . .	88
Tabela 26 – Acoplamento CV do F-16 - Sistema 4. . . . .	89
Tabela 27 – Acoplamento WB do F-16 - Sistema 0. . . . .	89
Tabela 28 – Acoplamento WB do F-16 - Sistema 1. . . . .	89
Tabela 29 – Acoplamento WB do F-16 - Sistema 2. . . . .	90
Tabela 30 – Acoplamento WB do F-16 - Sistema 3. . . . .	90
Tabela 31 – Acoplamento WB do F-16 - Sistema 4. . . . .	90
Tabela 32 – Sensibilidades individuais dos autovalores. . . . .	91

# Lista de abreviaturas e siglas

AE	Autoestrutura
AAE	Alocação de Autoestruturas
AG	Algoritmo Genético
IC	Inteligência Computacional
LU	Decomposição LU
LI	Linearmente Independente
MIMO	Múltiplas Entradas e múltiplas saídas
MFD	do inglês (Matrix Fractional Description)
MP	Método da Projeção
RNA	Rede Neural Artificial
RNA-E	Rede Neural Artificial Estruturada
SLP	do inglês (Single Layer Perceptron)
SSD	do inglês (State-Space Description)

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	Justificativas	16
1.2	Motivação	16
1.3	Estado da Arte	17
1.4	Objetivos	19
1.4.1	Objetivos Gerais	19
1.4.2	Objetivos Específicos	19
1.5	Organização da Dissertação	19
<b>2</b>	<b>ALOCAÇÃO DE AUTOESTRUTURAS E MÉTODOS PARA ANÁLISE DE SISTEMAS MULTIVARIÁVEIS</b>	<b>21</b>
2.1	Sistemas Multivariáveis	21
2.1.1	Descrição do Problema de Alocação de Autoestruturas	23
2.1.2	Teorema de Moore	25
2.2	Alocação de Autovetores Desejados	27
2.3	Métodos de Análise de Sistemas Multivariáveis	31
2.3.1	A AE e a Resposta do Sistema Dinâmico	31
2.3.2	Sensibilidade Individual dos Autovalores	36
2.3.3	Decomposição em Valores Singulares	36
<b>3</b>	<b>REDES NEURONAIS ARTIFICIAIS SLP E ESTRUTURADAS PARA ALOCAÇÃO ONLINE DE AUTOESTRUTURAS</b>	<b>39</b>
3.1	Redes Neurais Artificiais	39
3.1.1	Neurônio Artificial	40
3.1.2	Treinamento de uma Rede Neuronal	41
3.1.3	Algoritmos de Aprendizado	42
3.2	Redes Neurais Estruturadas	43
3.2.1	Sistemas de Equações Algébricas Lineares Simultâneas	44
3.2.2	Solução de Equações Lineares pelos Mínimos Quadrados	45
3.2.3	Resolução de Problemas Lineares usando a Abordagem Neural baseada em Mínimos Quadrados	46
3.3	Proposta de RNA-E para AAE	49
3.3.1	Algoritmo RNA-E	53
<b>4</b>	<b>ESTUDOS DE CASO E EXPERIMENTOS COMPUTACIONAIS</b>	<b>56</b>
4.1	Circuito Elétrico de 4ª ordem	57

4.1.1	Descrição em Espaço de Estados . . . . .	58
4.1.2	Alocação da Autoestrutura através do Método da Projeção . . . . .	59
4.1.3	Alocação Completa da Autoestrutura Desejada do Circuito Elétrico RLC . .	60
4.1.3.1	Simulação com $\mu = 0,6$ para $RNA-E_W$ e $RNA-E_K$ . . . . .	60
4.1.3.2	Simulação com $\mu = 0,7$ para $RNA-E_W$ e $RNA-E_K$ . . . . .	62
4.1.3.3	Simulação com $\mu = 0,8$ para $RNA-E_W$ e $RNA-E_K$ . . . . .	63
4.1.3.4	Simulação com $\mu = 0,9$ para $RNA-E_W$ e $RNA-E_K$ . . . . .	65
4.1.3.5	Método Iterativo de Richardson . . . . .	67
4.1.4	Análise dos Valores Singulares . . . . .	68
4.1.5	Análise Modal . . . . .	70
4.1.6	Análise da Sensibilidade Individual dos Autovalores . . . . .	74
<b>4.2</b>	<b>Aeronave F-16 . . . . .</b>	<b>75</b>
4.2.1	Alocação da Autoestrutura através do Método da Projeção . . . . .	75
4.2.2	Alocação completa da Autoestrutura Desejada do Sistema F-16 . . . . .	76
4.2.2.1	Simulação com $\mu = 0,001$ para $RNA-E_W$ e $\mu = 0,4$ para $RNA-E_K$ . . . . .	77
4.2.2.2	Simulação com $\mu = 0,002$ para $RNA-E_W$ e $\mu = 0,5$ para $RNA-E_K$ . . . . .	78
4.2.2.3	Simulação com $\mu = 0,003$ para $RNA-E_W$ e $\mu = 0,6$ para $RNA-E_K$ . . . . .	80
4.2.2.4	Simulação com $\mu = 0,005$ para $RNA-E_W$ e $\mu = 0,7$ para $RNA-E_K$ . . . . .	81
4.2.2.5	Método Iterativo de Richardson . . . . .	83
4.2.3	Análise dos Valores Singulares . . . . .	84
4.2.4	Análise Modal . . . . .	86
4.2.5	Análise da Sensibilidade Individual dos Autovalores . . . . .	90
	<b>5 CONCLUSÃO . . . . .</b>	<b>92</b>
<b>5.1</b>	<b>Conclusões Gerais . . . . .</b>	<b>92</b>
<b>5.2</b>	<b>Contribuições . . . . .</b>	<b>93</b>
<b>5.3</b>	<b>Trabalhos Futuros . . . . .</b>	<b>93</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>94</b>
	<b>A MODELAGEM MATEMÁTICA DA AERONAVE F-16 . . . . .</b>	<b>99</b>
<b>A.1</b>	<b>Descrição do Modelo . . . . .</b>	<b>99</b>

# Capítulo 1

## Introdução

As abordagens bioinspiradas, tais como redes neuronais associadas a estruturas de otimização, são os pilares para a formulação e o desenvolvimento de métodos para solução de problemas de alocação completa e parcial de autoestruturas em sistemas dinâmicos com múltiplas entradas e múltiplas saídas (MIMO). A alocação completa leva em consideração a especificação de todos os autovalores e os componentes dos autovetores associados, enquanto a atribuição parcial considera apenas a especificação de alguns pares da autoestrutura ( $AE$ ).

Após a formulação dos métodos para alocação de autoestruturas ( $AAE$ ) que são propostos nesta dissertação, apresenta-se os métodos para solução do problema de  $AAE$  nas formas explicitadas no parágrafo anterior, a saber completa e parcial. Os métodos de solução baseiam-se em metodologias bioinspiradas do tipo redes neuronais para resolver o problema de  $AAE$ . Estes métodos de solução são comparados com os métodos clássicos que são apresentados em (LIU; PATTON, 1998).

O problema de alocação de autoestrutura é de grande importância na teoria e aplicações de controle, pois a estabilidade e o comportamento dinâmico de um sistema multivariável linear são governados pela autoestrutura do sistema (LU; CHIANG; THORP, 1993). Em geral, a velocidade de resposta é determinada pelos autovalores, enquanto a forma da resposta é fornecida pelos autovetores. A alocação da autoestrutura é um método excelente para incorporar especificações clássicas sobre amortecimento, tempo de acomodação e desacoplamento de modo em uma estrutura MIMO (SOBEL; LALLMAN, 1988) e tem se mostrado uma ferramenta útil para o projeto de controle de voo (SOBEL; SHAPIRO; JR, 1994). A técnica de alocação de autoestrutura é usada para projetar leis de controle de voo para aeronaves com muitos efeitos de controle, e a técnica juntamente com o projeto *feedforward* adequado podem alcançar o desacoplamento estático com estabilidade interna, que é um requisito importante em muitos sistemas de controle de voo.

O objetivo deste trabalho é desenvolver um método para alocação de autoestrutura



em sistemas MIMO por realimentação de estados e também desenvolver um algoritmo para alocação de AE que se baseia em redes neuronais para execução da lei de controle por realimentação de estado e uma outra RNA-E para o treinamento da primeira. O uso da rede neuronal justifica-se porque os sistemas estudados podem ser sobredeterminados ou subdeterminados em larga escala, mal condicionados, e sistemas que possuem incertezas associadas. Além disso, tal estrutura de RNA evita o cálculo da matriz inversa (HAM; KOSTANIC, 2001).

## 1.1 Justificativas

O problema de Alocação de Autoestrutura (AAE) tem sua origem no problema de alocação de autovalores de sistemas multivariáveis, em que observou-se que, para um determinado autovalor alocado, existe um conjunto de autovetores que correspondem ao autovalor especificado, o que não acontece nos sistemas monovariáveis. Devido à influência que os autovalores exercem na resposta transitória de sistemas dinâmicos, o problema de alocação de autoestruturas tornou-se relevante no projeto e na análise de sistemas de controle.

Os índices de desempenho (estabilidade, velocidade de resposta e formas de onda) de um sistema dinâmico multivariável podem ser modificados de acordo com as especificações do projetista através da Alocação de Autoestruturas. Portanto, o desenvolvimento de métodos e técnicas para alocação de autovalores e autovetores em sistemas MIMO justifica-se através da sua importância no desempenho do sistema.

Os trabalhos de pesquisas desenvolvidos por (NETO, 2000), (FERREIRA; NETO, 2003) e (FERREIRA; NETO; TORRICO, 2003) apresentam uma alternativa para resolver o problema de AAE no contexto de controle ótimo, os mesmos apresentam modelos de busca baseados em técnicas de computação evolutiva, algoritmo genético (AG). Nesta pesquisa, apresenta-se uma proposta para solucionar o problema de AAE com a utilização de duas redes neuronais estruturadas, a primeira rede fornece os parâmetros para a segunda rede e, assim, obtendo uma matriz de ganho que aloque a autoestrutura desejada.

## 1.2 Motivação

A motivação para o desenvolvimento desta pesquisa é guiada por meio de técnicas de inteligência computacional (IC), trata-se de sintetizar o conhecimento necessário para resolver um problema de controle multivariável no conjunto das teorias de controle moderno e clássico. No intuito de realizar a alocação de autoestruturas (AAE) de forma online, esta autoestrutura é composta pelos pares de autovalores e autovetores, os quais influenciam nas respostas de sistemas de malha fechada.

### 1.3 Estado da Arte

As redes neurais tem se mostrado uma eficiente ferramenta para solução online de sistemas de equações lineares com coeficientes reais constantes e/ou variáveis no tempo (CICHOCKI; UNBEHAUEN, 1992b), (CICHOCKI; UNBEHAUEN, 1992a), (ZHANG, 1993), (HANG et al., 1997) . Estas redes, semelhantes a neurônios, podem ser utilizadas, após pequenas modificações, em problemas relacionados, a saber, inversão e pseudo-inversa de matrizes e para resolução de problemas de programação linear e quadrática. Especial ênfase é dada aos problemas mal condicionados.

Redes neurais recorrentes também são utilizadas para resolver equações lineares simultâneas. Em muitos problemas de grande escala, o número de incógnitas envolvidas é muito grande. Esses problemas em larga escala geralmente precisam ser resolvidos em tempo real. (HANG et al., 1997) utiliza um arranjo sistólico que fornece aceleração linear sobre a execução sequencial em uma máquina com um único processador. O arranjo sistólico é baseado em uma topologia de anel e execução síncrona, permitindo o uso de um único controlador para todos os elementos de processamento.

A ideia de uma rede neuronal estruturada para solução de uma variedade de problemas de álgebra matricial é apresentada por (WANG; MENDEL, 1990). Estes problemas são representados por uma arquitetura de uma RNA-E, esta rede deve ser treinada, para corresponder a alguns padrões desejados e obter a solução para o problema a partir dos pesos da rede estruturada resultante. Os algoritmos de treinamento para os problemas são *back-propagation* ou a modificação do erro do *back-propagation*. Em (WANG; MENDEL, 1990) apresenta três redes estruturadas e os correspondentes algoritmos de treinamento que são a decomposição da matriz LU, resolução de equações lineares e decomposição de valores singulares.

Podemos citar como aplicação relevante da AAE o trabalho desenvolvido por (GAN; DANAI, 1999), o mesmo explora a utilidade de uma rede neuronal recorrente baseada em modelos (MBRNN) no diagnóstico de falhas. A característica marcante dessa rede é a incorporação do conhecimento analítico do sistema em sua estrutura, o que reduz a demanda por treinamento. A fim de fornecer uma base para o desempenho deste método e aproveitar as outras soluções baseadas em modelos em sua formulação.

Em (YAICI; HARICHE, 2014) apresenta um método para alocação total de uma autoestrutura desejada usando bloco de polos. Os sistemas descritos nas equações do espaço de estados (SSD) são transformados em sistemas na descrição das frações matriciais (MFD), os autovalores são chamados de autovalores latentes e os autovetores são chamados de vetores latentes. O método proposto permite alocar, exatamente, todo o conjunto (e ainda mais) de valores latentes e vetores obtidos de uma autoestrutura desejada usando compensação dinâmica. A condição necessária é a controlabilidade do bloco ou a

observabilidade do bloco do sistema.

A referência (LIU; LU, 1997) desenvolve uma técnica de projeto para memórias associativas com capacidades de aprendizagem e esquecimento por meio de redes neuronais de realimentação artificial. A técnica de síntese proposta utiliza o método da AAE. As redes geradas são capazes de aprender novos padrões, bem como de esquecer os padrões aprendidos, sem a necessidade de recalcular todos os pesos de interconexão e entradas externas. Em muitos aspectos, os resultados constituem melhorias significativas em relação ao método de produto externo, a regra de aprendizagem de projeção e o método pseudo-inversa com restrições de estabilidade.

Um estudo teórico é apresentado em (WEI; MOTTERSHEAD, 2016) sobre a viabilidade da aplicação do controle ativo para o isolamento de vibrações em estruturas leves por meio da diagonalização de blocos das matrizes do sistema e ao mesmo tempo alocar autovalores às subestruturas escolhidas separadamente. O mesmo, faz a alocação da autoestrutura usando o método de receptações, o método proposto funciona com sucesso quando os autovalores do sistema de malha aberta são controláveis e os autovalores de malha-fechada são distintos. O sistema de malha fechada é desacoplado para formar subestruturas independentes e é demonstrado que autovalores podem ser alocados às subestruturas separadamente. No caso de matrizes de sistemas em bandas, o número de atuadores requeridos pode ser reduzido para duas vezes a largura de semi-banda.

Uma outra aplicação da AAE é proposto por (KONARA; ANNAKAGE, 2016). Neste, é apresentado um controlador de amortecimento em um sistema de potência usando AAE à parcial direita. A técnica de AAE seleciona um conjunto de autovalores de malha fechada junto com seus autovetores à direita ou esquerda. A seleção de autovetores oferecem uma flexibilidade extra que é explorada ao projetar um controlador de amortecimento robusto que fornece o amortecimento requerido sob várias condições de operação. Diferentemente dos algoritmos propostos anteriormente na literatura, este incorpora diferentes condições de operação no próprio estágio de projeto para obter um controlador de amortecimento robusto. Mais importante, quando os sinais medidos remotamente são usados no controlador, a robustez pode ser melhorada.

A referência (BELOTTI; RICHIEDEI, 2018) propõe uma abordagem híbrida na qual o sistema mecânico e o controlador são projetados simultaneamente para melhorar a capacidade de obtenção da autoestrutura desejada. De fato, a modificação adequada dos parâmetros inerciais e elásticos do sistema modifica o conjunto de autovetores que podem ser obtidos através do controle ativo. Este, demonstra que tal objetivo pode ser efetivamente realizado minimizando a classificação de uma determinada matriz que depende das características do sistema original e dos pares da autoestrutura desejada.

Os modos físicos de um sistema podem ser desacoplados pela modelagem apropriada dos autovetores como apresentado por (ALBOSTAN; GÖKAŞAN, 2018). Este apresenta

o projeto de um controlador robusto com base na técnica de alocação de autoestrutura. Mantendo a estrutura do autovetor desejado o máximo possível, o algoritmo de otimização Nelder-Mead é implementado para a busca da solução robusta ideal. Diferentes funções de custo de otimização são investigadas em termos de sensibilidade de autovalores e robustez de estabilidade. A eficácia do sistema de controle é demonstrada com simulações não lineares no modelo matemático de aeronaves F-16.

## 1.4 Objetivos

Os objetivos desta pesquisa em alocação de autoestruturas via aprendizado por máquinas estão categorizados em gerais e específicos. Desta forma, distinguem-se pontos abrangentes a serem alcançados para que sejam logrados os fins essenciais.

### 1.4.1 Objetivos Gerais

Apresentar um método para resolver de forma online o problema de Alocação de Autoestrutura em sistemas MIMO utilizando redes neuronais artificiais. A solução proposta consiste em desenvolver RNA-E e RNA-SLP para projeto online de sistemas de controle.

### 1.4.2 Objetivos Específicos

- Estudar modelos e pesquisas relacionadas à Alocação de Autoestruturas.
- Pesquisar projetos de sistemas de controle.
- Formular o problema de AAE baseada em redes neuronais para aplicações online.
- Desenvolver algoritmos baseados em RNA-E e RNA-SLP como ferramenta para Alocação de Autoestruturas.
- Desenvolver um algoritmo de aprendizado para treinar a RNA para aplicações online em sistemas de controle inteligente.
- Avaliar o projeto dos algoritmos propostos em modelos matemáticos de sistemas e verificar o seu desempenho em face a convergência e os objetivos de controle.

## 1.5 Organização da Dissertação

Esta dissertação está organizada em capítulos, que estão dispostos da seguinte forma: No Capítulo 2 apresenta-se as preliminares, abordando-se a formulação clássica do problema de alocação de autoestruturas e métodos de solução, bem como três métodos

de análise de sistemas multivariáveis , tais como análise modal, análise da sensibilidade individual e por fim decomposição de valores singulares. No Capítulo 3 apresenta-se a formulação e as soluções propostas para resolver o problema de alocação de autoestrutura de forma bio-inspirada e online, apresenta-se os algoritmos que compõem o sistema de controle para alocação de autoestruturas baseadas em redes neurais do tipo *feedforward* e estruturadas. No Capítulo 4, apresentam-se os resultados provenientes de avaliações em modelos matemáticos de plantas de ordem reduzida de sistemas do mundo real, tais como sistema circuitos elétricos e uma aeronave F-16. Salientando-se as vantagens e desvantagens do método do ponto de vista algorítmico, i.e., custos computacionais, tais como: memória e quantidade de operações e perspectivas de implementação em sistemas do mundo real. No Capítulo 5, apresenta-se conclusões e comentários a respeito do desenvolvimento do trabalho.

## Capítulo 2

# Alocação de Autoestruturas e Métodos para Análise de Sistemas Multivariáveis

O problema de alocação de autoestrutura (AAE) consiste na especificação de autovalores e autovetores desejados associados aos sistemas multivariáveis (MIMO), isto permite grande flexibilidade na modelagem de respostas do sistema em malha fechada. Os autovalores são responsáveis pela estabilidade e taxa de crescimento e decaimento da resposta temporal dos sistemas, enquanto que os autovetores estão relacionados com sua forma, ou seja, o efeito da autoestrutura na resposta do sistema. Isto significa verificar os efeitos provocados pelas interações entre a autoestrutura, as condições iniciais e as matrizes de entrada e de saída do sistema (LIU; PATTON, 1998).

Este capítulo apresenta formulações matemáticas do problema de AAE, inclui a definição de alocação de autovalores e autovetores desejados, o compromisso entre autovalores e autovetores em relação ao comportamento dinâmico de sistemas multivariáveis, alocação da autoestrutura para o caso complexo, alocação completa da AE desejada e o cálculo das matrizes do ganho de retroação de estados. Ainda neste capítulo são apresentados dois métodos de análise de sistemas multivariáveis, a saber, análise dos modos do sistema dinâmico e análise dos valores singulares.

### 2.1 Sistemas Multivariáveis

Um sistema multivariável (MIMO) é caracterizado por múltiplas-entradas e múltiplas saídas. Considere, em particular, um sistema MIMO modelado por combinações lineares das variáveis de estado  $x_1, x_2, \dots, x_n$  e as entradas de controle  $u_1, u_2, \dots, u_q$ ,

ou seja,

$$\begin{aligned} \dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + b_{11}u_1 + \cdots + b_{1q}u_q \\ &\vdots \\ \dot{x}_n &= a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n + b_{n1}u_1 + \cdots + b_{nq}u_q. \end{aligned} \quad (2.1)$$

Similarmente, para as variáveis de saída  $y_1, y_2, \dots, y_p$ , tem-se

$$\begin{aligned} y_1 &= c_{11}x_1 + c_{12}x_2 + \cdots + c_{1n}x_n + d_{11}u_1 + \cdots + d_{1q}u_q \\ &\vdots \\ y_p &= c_{p1}x_1 + c_{p2}x_2 + \cdots + c_{pn}x_n + d_{p1}u_1 + \cdots + d_{pq}u_q. \end{aligned} \quad (2.2)$$

A lei de controle do sistema dinâmico é definida como

$$\begin{aligned} u_1 &= K_{11}x_1 + K_{12}x_2 + \cdots + K_{1n}x_n \\ &\vdots \\ u_q &= K_{q1}x_1 + K_{q2}x_2 + \cdots + K_{qn}x_n \end{aligned}, \quad (2.3)$$

A descrição do sistema de Eqs. (2.1) e (2.2) na forma vetorial é dada por

$$\dot{x} = Ax + Bu \quad (2.4)$$

$$y = Cx + Du, \quad (2.5)$$

sendo  $A \in \mathfrak{R}^{n \times n}$  a matriz que representa a dinâmica do sistema,  $B \in \mathfrak{R}^{n \times q}$  a matriz de controle,  $u \in \mathfrak{R}^{q \times 1}$  o vetor controle,  $C \in \mathfrak{R}^{p \times n}$  a matriz de saída,  $D \in \mathfrak{R}^{p \times q}$  a matriz de transmissão direta,  $x \in \mathfrak{R}^n$  o vetor de estados,  $y \in \mathfrak{R}^{p \times 1}$  o vetor de saídas. As Eqs. (2.4) e (2.5) são referidas como descrição em espaço de estado.

De forma similar, a descrição da lei de controle de Eq. (2.3) na forma compacta é dada por

$$u = -Kx, \quad (2.6)$$

sendo  $K \in \mathfrak{R}^{q \times n}$  é a matriz de ganho de realimentação de estados.

Substituindo a Eq. (2.3) na Eq. (2.1), tem-se a equação de estados para o regulador que é dada por

$$\dot{x} = A_{cl}x, \quad (2.7)$$

sendo  $A_{cl}$  a matriz de malha fechada que é definida por

$$A_{cl} = (A - BK). \quad (2.8)$$

### 2.1.1 Descrição do Problema de Alocação de Autoestruturas

O problema a ser resolvido é determinar a matriz de ganho de realimentação de estado  $K$  que aloca a autoestrutura à direita especificada pelo projetista, ou seja, um determinado conjunto de autovalores  $\lambda_{di}$  e autovetores direitos  $V_{di}$ .

A autoestrutura desejada é representada pelo conjunto  $\{\Lambda_d, V_d\}$ , onde  $\Lambda_d$  é a matriz diagonal de autovalores do sistema dinâmico em malha fechada definido como

$$\Lambda_d = \text{diag} \left[ \lambda_{d1} \quad \lambda_{d2} \quad \cdots \quad \lambda_{dn} \right], \quad (2.9)$$

e  $V_d$  é a matriz de autovetores associada é dada por

$$V_d = \left[ v_{d1} \quad v_{d2} \quad \cdots \quad v_{dn} \right], \quad (2.10)$$

sendo  $v_{di} \in \mathbb{C}^n$  e  $\lambda_{di} \in \mathbb{C}$ .

Para garantir a AE desejada, a matriz de ganhos  $K$  deve satisfazer

$$A_{cl}V_d = \left[ \lambda_{d1}v_{d1} \quad \lambda_{d2}v_{d2} \quad \cdots \quad \lambda_{dn}v_{dn} \right]. \quad (2.11)$$

A partir da Eq. (2.11), observa-se que o problema consiste em determinar a solução de um conjunto de  $n$  sistemas lineares. Desta forma, para cada par  $(\lambda_i, v_i)$ , tem-se um sistema linear para ser resolvido. No caso de realimentação completa dos estados, os  $n$  sistemas lineares devem ser resolvidos, o que não ocorre no caso da alocação parcial da autoestrutura. Consequentemente, a complexidade computacional do processo iterativo para determinar o vetor de ganho  $K$  depende do número de variáveis de controle e da ordem  $n$  do sistema, que estão relacionadas ao tamanho da matriz de ganho e ao número de sistemas lineares.

Substituindo a Eq. (2.8) na Eq. (2.11), considerando que  $n$  sistemas lineares em termos das componentes dos autovetores  $v_i$  e das componentes  $\lambda_i$  devem ser resolvidos para solucionar o problema de AE, tem-se

$$(A - BK)v_i = \lambda_i v_i. \quad (2.12)$$

Assim, o problema de AE em termos da  $i$ -ésima autoestrutura calculada é dada por

$$Av_i - BKv_i = \lambda_d v_i \quad (2.13)$$

Observe que a Eq. (2.13) é escrita explicitamente em termos dos ganhos  $K$ , ou seja, a matriz de ganho de realimentação de estados em malha fechada é desconhecida. A forma clássica que relaciona a AAE na sua forma geral é dada por

$$(A - \lambda_i I)v_i - BKv_i = 0. \quad (2.14)$$



A Eq. (2.14), que é não-linear para os desconhecidos  $K$  e  $v_i$ , em sua forma vetorial é expressa por

$$[(A - \lambda_i I) \quad B] \begin{bmatrix} v_i \\ -Kv_i \end{bmatrix} = 0, \quad i = 1, \dots, n. \quad (2.15)$$

Desta forma, o problema de  $AAE$  pode ser descrito como um problema de espaço nulo.

Para resolver o problema de  $AAE$ , utiliza-se o seguinte artifício matemático, levando em conta que a autoestrutura é especificada. Assim, a variável auxiliar  $w_i$  é inserida em (2.15) e é dada por

$$w_i = Kv_i \quad (2.16)$$

O valor de  $w_i$  para um dado  $v_i$  é calculado no contexto de busca de uma dada matriz de ganho  $K$  que deve satisfazer o conjunto de restrições da autoestrutura desejada, ou seja,  $\{\Lambda_d, V_d\}$ .

Substituindo a Eq. (2.16) na Eq. (2.15), obtém-se uma equação que relaciona o par  $(v_i, \lambda_i)$  com a variável auxiliar  $w_i$ . A nova relação é dada por

$$(A - \lambda_i I)v_i - Bw_i = 0 \quad (2.17)$$

A solução direta para determinação do autovetor  $v_i$  é dada pela inversa da matriz  $(A - \lambda_i I)$  por

$$v_i = (A - \lambda_i I)^{-1} Bw_i. \quad (2.18)$$

Determina-se o vetor de ganhos  $w_i$  de acordo com a seguinte equação

$$w_i = (B^T B)^{-1} B^T (A - \lambda_i I)v_i \quad (2.19)$$

Uma vez que  $w_i$  é calculado, para cada  $i$ , a partir da Eq. (2.19), o ganho de realimentação  $K$  torna-se

$$K = WV_d^{-1}, \quad (2.20)$$

sendo  $W = [w_1 \ w_2 \ \dots \ w_n]$

A Alocação de autoestrutura tem por base o teorema de (MOORE, 1976), o qual salienta a caracterização da classe de todos os autovetores que podem ser obtidos para um dado conjunto de autovalores de malha fechada. Tem, também, o objetivo de apresentar um algoritmo para calcular a matriz de ganhos de malha fechada que fornece a autoestrutura desejada e que é a base para o desenvolvimento de vários métodos de alocação. O principal objetivo é mostrar que, se certas condições necessárias e suficientes forem satisfeitas, existe uma matriz  $K$  de malha fechada que impõe a autoestrutura especificada. Esta proposição é apresentada na sequência.

### 2.1.2 Teorema de Moore

Moore foi o primeiro a identificar a liberdade oferecida pela realimentação de estados além da especificação dos autovalores de malha fechada para o caso em que os autovalores são distintos. Moore considera o sistema descrito por (KIMURA, 1978) e deriva condições necessárias e suficientes para a existência de  $K$ , que produz os autovalores e autovetores prescritos.

Antes de apresentar o teorema de Moore, define-se

$$S_\lambda \triangleq [ \lambda_i I - A \quad B ], \quad (2.21)$$

sendo  $\lambda_i$  o  $i$ -ésimo elemento do conjunto de autovalores  $\{\lambda_i\}_{i=1}^n$ . Associada à Eq.(2.21), tem-se a matriz particionada

$$R_\lambda = [ N_\lambda \quad M_\lambda ]^T, \quad (2.22)$$

sendo que as colunas de  $R_\lambda$  formam uma base do espaço nulo da matriz  $S_\lambda$ . Considera-se que  $B$  tenham colunas LI, desta forma as colunas de  $N_\lambda$  são LI.

**Teorema (MOORE, 1976).** Seja  $\{\lambda_i\}_{i=1}^n$  um conjunto de autovalores auto-conjugados distintos. Então, existe uma matriz  $K^{m \times n} \in \Re$  tal que (SOBEL; SHAPIRO; JR, 1994)

$$(A + BK)v_i = \lambda_i v_i \quad i = 1, \dots, n. \quad (2.23)$$

se, e somente se, para cada  $i$ , as três condições a seguir são satisfeitas:

- (i) os vetores  $\{v_i\}_{i=1}^n$  são linearmente independentes em  $\mathbb{C}^n$ ,
- (ii)  $v_i = v_j^*$  sempre que  $\lambda_i = \lambda_j^*$ ,
- (iii)  $v_i \in \text{sub}_{ger}(N_{\lambda_i})$ , sendo que o  $\text{sub}_{ger}(N_{\lambda_i})$  representa o subespaço de  $\mathbb{C}^n$  gerado pelas colunas de  $N_{\lambda_i}$ .

Se  $K$  existe e  $\text{posto}(B) = m$ , então  $K$  é única.

A seguir são apresentados as provas necessárias e suficientes do teorema de Moore.

**Prova:**

**Necessidade.** A necessidade de (i) e (ii) segue da teoria matricial. Agora, da Eq.(2.23), tem-se que

$$(\lambda_i I - A)v_i = BKv_i, \quad (2.24)$$

ou

$$[ \lambda_i I - A \quad B ] \begin{bmatrix} v_i \\ Kv_i \end{bmatrix} = 0, \quad (2.25)$$

uma vez que as colunas de  $R_{\lambda_i}$  formam uma base para o espaço nulo de  $S_{\lambda_i}$ , em consequência tem-se  $v_i \in \text{sub}_{ger}(N_{\lambda_i})$ .

**Suficiência.** Assumindo que o conjunto de autovetores  $\{v\}_{i=1}^n$  satisfaz as condições (i), (ii) e (iii), então existe um vetor  $z_i$  (real ou complexo) tal que

$$v_i = N_{\lambda_i} z_i. \quad (2.26)$$

A partir das definições de  $S_\lambda$  e  $R_\lambda$  tem-se

$$(\lambda_i I - A)N_{\lambda_i} z_i + BM_{\lambda_i} z_i = 0. \quad (2.27)$$

Se  $K$  pode ser escolhido de modo que

$$-M_{\lambda_i} z_i = K v_i. \quad (2.28)$$

então

$$[\lambda_i I - (A + BK)] v_i = 0. \quad (2.29)$$

Moore prossegue mostrando que isso pode ser feito da seguinte maneira, se existe um  $K$ , logo, a seguinte condição deve ser satisfeita

$$K \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} = \begin{bmatrix} -M_{\lambda_1} z_1 & -M_{\lambda_2} z_2 & \cdots & -M_{\lambda_n} z_n \end{bmatrix}. \quad (2.30)$$

Para cada  $\lambda_i$  com valores reais, então temos que  $v_i$  e  $z_i$  são reais e a matriz de autovetores é não singular. Daí,

$$K = \begin{bmatrix} -M_{\lambda_1} z_1 & -M_{\lambda_2} z_2 & \cdots & -M_{\lambda_n} z_n \end{bmatrix} \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}^{-1}. \quad (2.31)$$

Se os autovalores desejados são complexos, a equação acima passa por uma leve mudança. Sendo  $w_i = M_{\lambda_i} z_i$  e assumindo que  $\lambda_1 = \lambda_2^*$ , e  $v_1 = v_2^*$  é dado a partir da segunda proposição de Moore, isto implica que  $z_1 = z_2^*$ . E assumindo que todos os outros autovalores são reais, então  $K$  precisa satisfazer

$$K \begin{bmatrix} v_{1R} + jv_{1I} & v_{1R} - jv_{1I} & v_3 & \cdots & v_n \end{bmatrix} = \begin{bmatrix} w_{1R} + jw_{1I} & w_{1R} - jw_{1I} & w_3 & \cdots & w_n \end{bmatrix}. \quad (2.32)$$

Multiplicando ambos os lados da Eq. (2.32) por uma matriz de transformação não singular

$$\left[ \begin{array}{cccc|c} 1/2 & -j1/2 & & & 0 \\ 1/2 & +j1/2 & & & 0 \\ \hline & & 0 & & I \end{array} \right], \quad (2.33)$$

obtem-se

$$K \begin{bmatrix} v_{1R} & v_{1I} & v_3 & \cdots & v_n \end{bmatrix} = \begin{bmatrix} w_{1R} & w_{1I} & w_3 & \cdots & w_n \end{bmatrix}. \quad (2.34)$$

Desde que o conjunto de  $\{v\}_{i=1}^n$  seja LI, a matriz  $\begin{bmatrix} v_{1R} & v_{1I} & v_3 & \cdots & v_n \end{bmatrix}$  é não singular e, novamente, a matriz  $K$  pode ser calculada.

Finalmente, uma vez que uma autoestrutura fixada define unicamente  $A + BK$ ,  $K$  também é único desde que o  $\text{posto}[B] = m$ .

Na próxima Seção é apresentado o método de projeção para alocação de autoestrutura desejada.

## 2.2 Alocação de Autovetores Desejados

É conhecido que um autovetor à direita admissível em malha fechada pertence a um subespaço de dimensão  $m$ , onde  $m$  é o número de entradas (ou controles) e é geralmente menor que o número  $n$  de estados. Isto implica que os autovetores à direita em malha fechada não podem ser alocados arbitrariamente e somente aqueles que pertencem ao subespaço admissível correspondente podem ser alocados com exatidão. Normalmente, um autovetor desejado não é dado no subespaço do autovetor admissível. Assim, o autovetor desejado deve ser projetado no subespaço admissível, que é ótimo de acordo com algumas funções de desempenho. Segundo (ANDRY; SHAPIRO; CHUNG, 1983) a função de desempenho é dada por

$$J_p = (v_{d_i} - v_i)^T \tilde{W}_i (v_{d_i} - v_i), \quad (2.35)$$

sendo  $v_{d_i} \in \mathfrak{R}^{n \times 1}$  o autovetor desejado e  $\tilde{W}_i \in \mathfrak{R}^{n \times n}$  é uma matriz de ponderação diagonal. Existe um conjunto limitado de variações que são permitidos para as especificações de projeto, uma vez especificados os autovalores, tem-se um grau de liberdade matemático na escolha dos vetores para satisfazer os autovalores especificados. A projeção ótima é executada escolhendo o autovetor do subespaço admissível correspondente que minimiza a função de desempenho  $J_p$ , isto é

$$\min_{v_i} (v_{d_i} - v_i)^T \tilde{W}_i (v_{d_i} - v_i). \quad (2.36)$$

De forma geral, o autovetor desejado  $v_d$  nem sempre está contido no subespaço prescrito. Então é feita uma escolha do melhor possível autovetor alcançável. Este melhor autovetor é a projeção do autovetor  $v_d$  no subespaço gerado pelas colunas de  $(\lambda_i I - A)^{-1} B$ , como descrito na Fig. 1 (SOBEL; SHAPIRO; JR, 1994).

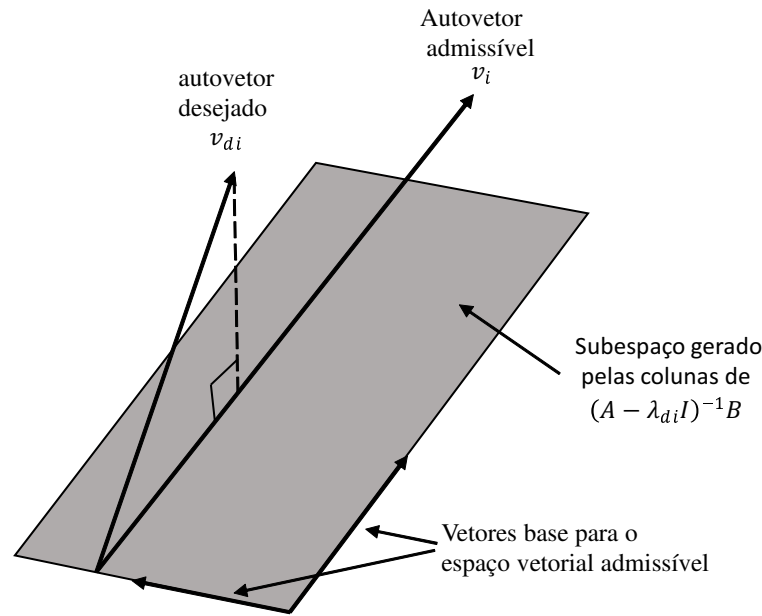


Figura 1 – Projeção do autovetor desejado.

A combinação gerada pelo espaço nulo fornece um vetor  $v_i$  que usado como autovetor produzirá um sistema em malha fechada com o autovalor desejado  $\lambda_i$ . O vetor desejado é projetado no espaço admissível para obtenção de um autovetor alcançável. Se o autovetor à direita for parametrizado como

$$v_i = (\lambda_i I - A)^{-1} B W_i, \quad (2.37)$$

a solução que minimiza a função de desempenho  $J_p$  da Eq. (2.35) é dada por

$$w_{o_i} = (P_{v_i}^T \tilde{W}_i P_{v_i})^{-1} P_{v_i}^T \tilde{W}_i v_{d_i}, \quad (2.38)$$

sendo

$$P_{v_i} = (\lambda_i I - A)^{-1} B, \quad (2.39)$$

e o melhor autovetor à direita é determinado por

$$v_{o_i} = P_{v_i} W_{o_i}. \quad (2.40)$$

Na maioria dos casos, apenas alguns dos elementos do autovetor desejado de malha fechada  $v_d$  são especificados. Em particular, os elementos especificados em um autovetor desejado são necessários ser zero ou unidade (SOBEL; SHAPIRO, 1987). Na prática, isso é importante porque a forma modal da resposta temporal do sistema de malha fechada desejado depende dos autovetores, bem como dos autovalores do sistema. Por exemplo, certos estados de um sistema não devem ser observados para terem um papel significativo na ação de modos específicos, enquanto outros devem ser vistos para exibirem um efeito forte, e vice-versa. De acordo com o papel dos autovetores do sistema de malha fechada,

isto pode ser assegurado atribuindo a cada um dos elementos do autovetor correspondentes aos estados que não devem ser observados na resposta de um determinado modo, o valor zero. Aqueles que devem ser observados são atribuídos um valor que é diferente de zero (nominalmente unidade)(LIU; PATTON, 1998). Para tal situação, uma abordagem para encontrar os melhores autovetores alcançáveis é agora descrita.

Considere agora a alocação de autovalores complexos. A alocação de autovalores reais pode ser realizada usando uma abordagem similar simplificada. O autovetor desejado complexo assumirá a seguinte forma

$$v_{d_i} = v_{d_i}^{re} + jv_{d_i}^{im} = \begin{pmatrix} v_{i1} \\ \bullet \\ v_{i3} \\ \bullet \end{pmatrix} + j \begin{pmatrix} \bullet \\ v_{i2} \\ v_{i4} \\ \bullet \end{pmatrix}, \quad (2.41)$$

sendo ( $\bullet$ ) um elemento não especificado (ou não restrito) e  $v_{ik}$  é um elemento especificado que está na  $k$ -ésima posição do autovetor desejado,

$$(A + BKC)(v_i^{re} + jv_i^{im}) = (v_i^{re} + jv_i^{im})(\lambda_i^{re} + j\lambda_i^{im}). \quad (2.42)$$

A representação aritmética real do subespaço do autovetor admissível à direita é dada de modo que as primeiras  $n$  linhas correspondam à parte real do autovetor admissível e o restante à parte imaginária. Assim,  $v_{d_i}$  é reformulado para

$$\bar{v}_{d_i} = \begin{pmatrix} v_{d_i}^{re} \\ v_{d_i}^{im} \end{pmatrix}. \quad (2.43)$$

Igualando as partes reais e imaginárias, obtém-se

$$\begin{bmatrix} v_{d_i}^{re} \\ v_{d_i}^{im} \end{bmatrix} = \begin{bmatrix} \lambda_i^{re} I - A & -\lambda_i^{im} I \\ \lambda_i^{im} I & \lambda_i^{re} I - A \end{bmatrix} \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix} \begin{bmatrix} W_i^{re} \\ W_i^{im} \end{bmatrix}. \quad (2.44)$$

O subespaço do autovetor admissível à direita é construído a partir das colunas de  $P_{v_i}^c$ . Uma vez que somente alguns elementos de  $\bar{v}_{d_i}$  são especificados, esses elementos e as linhas correspondentes de  $P_{v_i}^c$  são selecionados simultaneamente. É possível calcular o vetor  $\bar{W}_i$  tal que

$$\hat{v}_{d_i} = \hat{P}_{v_i}^c \bar{W}_i, \quad (2.45)$$

sendo

$$\bar{W}_i = \begin{bmatrix} W_i^{re} \\ W_i^{im} \end{bmatrix} \in \mathfrak{R}^{2m \times 1}, \quad (2.46)$$

sendo

$$P_{v_i}^c = \tilde{A}^{-1} \tilde{B}, \quad (2.47)$$

$$\tilde{A} = \begin{bmatrix} \lambda_i^{re} I - A & -\lambda_i^{im} I \\ -\lambda_i^{im} I & \lambda_i^{re} I - A \end{bmatrix}, \quad (2.48)$$

$$\tilde{B} = \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix}. \quad (2.49)$$

Se o número de elementos especificados em  $\hat{v}_{d_i}$  for menor que (ou igual a)  $2m$  ( $m$  no caso real), o sistema de Eq. (2.45) é indeterminado (ou exclusivamente determinado no caso de igualdade), de modo que  $v_d$ , será exatamente alcançado. No entanto, se  $\hat{v}_{d_i}$  tem mais de  $2m$  elementos (no caso real), o sistema de Eq. (2.45) é sobredeterminado (ou seja, sem solução). Neste caso, precisa-se calcular o vetor de projeto  $\bar{W}_i$  da Eq. (2.45) de modo que a função de desempenho  $J_p$  dada por

$$J_p = (\hat{v}_{d_i} - \hat{P}_{v_i}^c \bar{W}_i)^T \tilde{W}_i (\hat{v}_{d_i} - \hat{P}_{v_i}^c \bar{W}_i), \quad (2.50)$$

seja minimizada. Assim, uma solução de melhor ajuste para o problema dos mínimos quadrados ponderados é obtida por

$$\bar{W}_{o_i} = ((\hat{P}_{v_i}^c)^T \tilde{W}_i \hat{P}_{v_i}^c)^{-1} (\hat{P}_{v_i}^c)^T \tilde{W}_i \hat{v}_{d_i}, \quad (2.51)$$

e o melhor autovetor admissível à direita  $v_{o_i} = v_{o_i}^{re} + \tilde{j} v_{o_i}^{im}$  é, desta forma, calculado por

$$\bar{v}_{o_i} = \begin{bmatrix} v_{o_i}^{re} \\ v_{o_i}^{im} \end{bmatrix} = P_{R_i}^c \bar{W}_{o_i}. \quad (2.52)$$

Com base nas formulações descritas nas Eqs.(2.38), (2.39) e (2.40) pode-se desenvolver um algoritmo para alocação completa da autoestrutura especificada pelo método dos mínimos quadrados. Se a autoestrutura desejada não for admissível o algoritmo fornece o melhor autovetor possível admissível para as condições de projeto.

Algoritmo AAE-LS-MP1

- 
- 1 ▷ Configuração - Inicialização
  - 2 ▷ Selecionar as matrizes do sistema dinâmico -  $A$  e  $B$
  - 3 ▷ Selecionar a autoestrutura desejada -  $\lambda_{d_i}$ ,  $v_{d_i}$
- 
- 4 **para**  $n \leftarrow 1 : n_\lambda$
  - 5     **faça**
  - 6     ▷ Bases para os subespaços de autovetores admissíveis
  - 7      $P_{v_i} = (\lambda_i I - A)^{-1} B$
  - 8     ▷ Representação do autovetor admissível na base  $P_{v_i}$
  - 9      $W_{o_i} = (P_{v_i}^T \tilde{W}_i P_{v_i})^{-1} P_{v_i}^T \tilde{W}_i v_{d_i}$
  - 10    ▷ Autovetores admissíveis
  - 11     $v_{o_i} = P_{v_i} W_{o_i}$
  - 12    **fim**
  - 13 ▷ Matriz de Ganhos de Realimentação
  - 14  $K = W_o^{-1} v_o$
-

Algoritmo AAE-LS-MP2

---

```

1  ▷ Configuração - Inicialização
2  ▷ Selecionar as matrizes do sistema dinâmico -  $A$  e  $B$ 
3   $\tilde{A} = \begin{bmatrix} \lambda_i^{re} I - A & -\lambda_i^{im} I \\ -\lambda_i^{im} I & \lambda_i^{re} I - A \end{bmatrix}$  e  $\tilde{B} = \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix}$ 
4  ▷ Selecionar a autoestrutura desejada -  $\lambda_{di}, v_{di}$ 
5   $\bar{v}_{o_i} = \begin{bmatrix} v_{o_i}^{re} \\ v_{o_i}^{im} \end{bmatrix}$  e  $\lambda_{o_i} = \begin{bmatrix} \lambda_{o_i}^{re} \\ \lambda_{o_i}^{im} \end{bmatrix}$ 

```

---

```

6  para  $n \leftarrow 1 : n_\lambda$ 
7    faça
8    ▷ Bases para os subespaços de autovetores admissíveis
9     $P_{v_i}^c = \tilde{A}^{-1} \tilde{B}$ 
10   ▷ Representação do autovetor admissível na base  $P_{v_i}$ 
11    $\tilde{W}_{o_i} = ((\hat{P}_{v_i}^c)^T \tilde{W}_i \hat{P}_{v_i}^c)^{-1} (\hat{P}_{v_i}^c)^T \tilde{W}_i \hat{v}_{d_i}$ 
12   ▷ Autovetores admissíveis
13    $\hat{v}_{A_i} = \hat{P}_{v_i}^c \tilde{W}_i$ 
14   fim
15   ▷ Matriz de Ganhos de Realimentação
16    $K = W_o^{-1} V_A$ 

```

---

## 2.3 Métodos de Análise de Sistemas Multivariáveis

Nas subsecções a seguir são apresentados métodos para análise dos sistemas dinâmicos, primeiro apresenta-se a análise dos modos do sistema dinâmico, em seguida, é apresentado o método de análise individual dos autovalores e a análise da decomposição do valores singulares.

### 2.3.1 A AE e a Resposta do Sistema Dinâmico

Apresenta-se nesta subseção a influência exercida pela autoestrutura (AE) na resposta temporal do sistema dinâmico. A ideia básica é expressar a equação de estado em termos da autoestrutura e, logo após, fazer uma análise da influência dos autovalores e autovetores na resposta do sistema. Esta seção tem como referência o trabalho de (NETO, 2000). Seja o sistema linear homogêneo:

$$\dot{x} = Ax \quad (2.53)$$

com o conjunto de  $n$  autovalores distintos  $\{\lambda_i\}_{i=1}^n$ , aos quais correspondem  $n$  autovetores normalizados  $\{v_i\}_{i=1}^n$  linearmente independentes que constituem uma base do espaço  $n$  dimensional e são definidos por:

$$Av_i = \lambda_i v_i, \langle v_i, v_j \rangle = \delta_{i,j}, \quad i, j = 1, 2, \dots, n, \quad (2.54)$$

$$\delta_{i,j} = \begin{cases} 1, & \text{Se } i = j \\ 0, & \text{Se } i \neq j \end{cases}. \quad (2.55)$$



Portanto o vetor de estado  $x$  pode ser definido como:

$$x(t) = \sum_{i=1}^n a_i(t)v_i. \quad (2.56)$$

Aplicando-se uma transformação de equivalência

$$x = Vz, \quad (2.57)$$

obtém-se a forma normal ou modal

$$\dot{z} = \Lambda z = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} z, \quad (2.58)$$

sendo  $\Lambda = V^{-1}AV$  e as colunas de  $V$ , conhecida como matriz modal, são os autovetores  $v_i$ , associados aos autovalores  $\lambda_i$ , e as colunas de  $V^{-1}$  são os autovetores recíprocos ou autovetores à esquerda. Ainda,

$$\dot{z}_i = \lambda_i z_i; \quad i = 1, 2, \dots, n \quad (2.59)$$

cujas soluções são imediatas

$$z_i = \alpha_i e^{\lambda_i t}; \quad i = 1, 2, \dots, n; \quad \alpha_i = z_i(0). \quad (2.60)$$

Da Eq. (2.60), verifica-se de imediato que: 1. O sistema será assintoticamente estável se todos os autovalores tiverem parte real negativa. 2. Se certos autovalores tiverem parte real negativa de magnitude bem maior do que as outras, suas contribuições transitórias serão desprezíveis e poderemos tratar o sistema como tendo uma ordem inferior a  $n$ . Portanto,

$$x = Vz = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} \begin{bmatrix} \alpha_1 e^{\lambda_1 t} \\ \alpha_2 e^{\lambda_2 t} \\ \cdots \\ \alpha_n e^{\lambda_n t} \end{bmatrix} = \sum_{i=1}^n \alpha_i e^{\lambda_i t} v_i \quad (2.61)$$

$$x = \sum_{i=1}^n a_i(t)v_i = \sum_{i=1}^n \alpha_i e^{\lambda_i t} v_i, \quad (2.62)$$

mostra que para cada vetor característico  $v_i$ , corresponde um modo  $\alpha_i e^{\lambda_i t}$  e que a resposta livre é uma combinação linear dos modos do sistema.

As constantes podem ser facilmente determinadas pelo emprego da base recíproca. Se  $\begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$  constitui uma base de um espaço  $X$  de dimensão  $n$ , o conjunto de vetores  $\begin{bmatrix} w_1 & w_2 & \cdots & w_n \end{bmatrix}$  definidos tal que

$$\langle v_i, w_j \rangle = \delta_{i,j}, \quad i, j = 1, 2, \dots, n \quad (2.63)$$

é a base recíproca do espaço  $X$ . De (2.62), para autovalores reais:

$$x(0) = \sum_{i=1}^n \alpha_i v_i, \quad (2.64)$$

e escreve-se a resposta livre do sistema como:

$$x(t) = \sum_{i=1}^n \langle w_1, x(0) \rangle e^{\lambda_i t} v_i. \quad (2.65)$$

Para autovalores reais, se as condições iniciais forem tais que  $\alpha_i = 0$  para  $i \neq j$ , ou seja, se apenas o  $j$ -ésimo autovetor for excitado, então,

$$x(t) = \alpha_j e^{\lambda_j t} v_j. \quad (2.66)$$

Manipulando (2.53) e derivando (2.66)

$$\dot{x}(t) = Ax(t) = \alpha_j e^{\lambda_j t} Av_j = \alpha_j \lambda_j e^{\lambda_j t} v_j. \quad (2.67)$$

Portanto, neste caso, o vetor velocidade instantânea  $\dot{x}(t)$ , tangente à trajetória de estado, é orientado ao longo do autovetor, onde, qualquer trajetória que começa ou que intercepta um autovetor permanece sobre esta direção característica.

Analogamente, para o sistema linear estacionário forçado,

$$\dot{x}(t) = Ax(t) + Bu(t), \quad (2.68)$$

pode-se decompor a função forçante ao longo dos autovetores  $v_i$  fazendo

$$Bu(t) = H(t) = \sum_{i=1}^n h_i(t) v_i. \quad (2.69)$$

Desta forma, podemos escrever a equação de transição de estado como

$$x(t) = \sum_{i=1}^n \langle w_1, x(0) \rangle e^{\lambda_i t} v_i + \int_0^t \sum_{i=1}^n \langle w_1, Bu(\tau) \rangle e^{\lambda_i(t-\tau)} v_i d\tau. \quad (2.70)$$

A equação (2.70) mostra a influência da autoestrutura na resposta do sistema dinâmico, caracterizando esta resposta como uma combinação linear dos  $n$  modos  $e^{\lambda_i t} v_i$ ,  $i = 1, \dots, n$ , onde cada autovalor  $\lambda_i$  e o seu correspondente autovetor à direita  $v_i$  definem a característica no domínio do tempo e a forma do  $i$ -ésimo modo. A referida equação também ressalta o efeito da entrada sobre cada modo. Se a entrada for escolhida de tal maneira que apenas o  $j$ -ésimo modo for excitado e se o sistema for estável, teremos em regime

$$x_\tau(t) = \int_0^t \sum_{i=1}^n \langle w_1, Bu(\tau) \rangle e^{\lambda_i(t-\tau)} v_i d\tau. \quad (2.71)$$

Substituindo (2.70) em (2.5), obtém-se uma expressão que permite salientar simultaneamente a influência da autoestrutura tanto em nível de entrada quanto de saída do sistema dinâmico. Então,

$$y(t) = C \sum_{i=1}^n v_i e^{\lambda_i t} \langle w_i, x(0) \rangle + C \int_0^t \sum_{i=1}^n v_i e^{\lambda_i(t-\tau)} \langle w_i, Bu(\tau) \rangle d\tau. \quad (2.72)$$

Fazendo

$$\gamma_i = Cv_i, \quad i = 1, 2, \dots, n, \quad (2.73)$$

então  $\gamma_i$  são as colunas da matriz  $CV$

$$\begin{bmatrix} \gamma_1 & \gamma_2 & \cdots & \gamma_n \end{bmatrix} = C \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} \quad (2.74)$$

e

$$\beta'_j = w'_j B, \quad j = 1, 2, \dots, n, \quad (2.75)$$

sendo  $B'_j$  são linhas da matriz  $WB$

$$\begin{bmatrix} \beta'_1 & \beta'_2 & \cdots & \beta'_j \end{bmatrix}^T = \begin{bmatrix} w'_1 & w'_2 & \cdots & w'_j \end{bmatrix}^T B. \quad (2.76)$$

Como o conjunto dos  $n$  autovetores  $\{w_i\}_{i=1}^n$  é o recíproco do conjunto dos  $n$  autovetores  $\{v_i\}_{i=1}^n$ , tem-se que:

$$W = V^{-1}. \quad (2.77)$$

Substituindo (2.73) e (2.75) em (2.72) e sabendo que  $\langle w_i, Bu(\tau) \rangle = \langle w_i, B'u(\tau) \rangle$ , tem-se

$$y(t) = \sum_{i=1}^n \gamma_i e^{\lambda_i t} \langle w_i, x(0) \rangle + \sum_{i=1}^n \int_0^t \gamma_i e^{\lambda_i(t-\tau)} \langle \beta_i, u(\tau) \rangle d\tau. \quad (2.78)$$

O termo integral é uma convolução, então:

$$y(t) = \sum_{i=1}^n \gamma_i e^{\lambda_i t} \langle w_i, x(0) \rangle + \sum_{i=1}^n \gamma_i e^{\lambda_i t} * \langle \beta_i, u(\tau) \rangle. \quad (2.79)$$

Observando a equação (2.79), verifica-se que  $\gamma_i \langle w_i, x(0) \rangle$  e  $\langle \beta_i, u(\tau) \rangle$  relacionam as saídas, as condições iniciais e as entradas, respectivamente, com os autovetores, estabelecendo a magnitude modal  $e^{\lambda_i t}$ .

Os vetores linha  $\beta'_i$  são as linhas de  $WB = V^{-1}B$  e representam o acoplamento entre as entradas e os autovetores de  $A$ , que caracterizam o movimento no espaço de estado. Os vetores coluna  $\gamma_i$  são as colunas de  $CV$  e representam o acoplamento entre os autovetores no espaço de estado e o espaço de saída.

Todos os movimentos possíveis no espaço de estado podem ser expressos como uma soma dos movimentos característicos associados com os autovalores, os chamados modos como já visto. Uma entrada  $u(t)$  no sistema é acoplada ao  $i$ -ésimo modo pela realização do produto escalar  $(\beta_i, u(t))$ ; este componente é unido com o movimento característico exponencial para obter-se:

$$e^{\lambda_i}(\beta_i, u(t)). \quad (2.80)$$

Portanto a saída é composta da soma dos movimentos ao longo do conjunto de vetores  $\gamma_i$  do espaço de saída como

$$y(t) = \sum_{i=1}^n \gamma_i e^{\lambda_i} \langle \beta_i, u(t) \rangle. \quad (2.81)$$

Aplica-se a transformada de *Laplace* nas Eqs. (2.4) e (2.5) e considera-se condições iniciais nulas, e substituindo a solução da equação de estado  $x(s)$  na saída, tem-se a matriz de transferência  $G(s)$

$$G(s) = C(sI - A)^{-1}B. \quad (2.82)$$

Aplicando a transformada inversa de *Laplace* na Eq. (2.82), no domínio do tempo tem-se a matriz de resposta ao impulso dada por

$$G(t) = Ce^{At}B. \quad (2.83)$$

Das Eqs. (2.58) e (2.83) tem-se

$$G(s) = Ce^{V\Lambda W t}B \quad (2.84)$$

Expandindo em série de *Taylor* o termo  $e^{V\Lambda W t}$  e depois retornando ao termo exponencial, obtém-se este termo expresso em termos da autoestrutura da matriz  $A$

$$\begin{aligned} e^{V\Lambda W t} &= I + V\Lambda tW + \frac{(V\Lambda W)^2 t^2}{2!} + \frac{(V\Lambda W)^3 t^3}{3!} + \dots \\ &= V \left( I + \Lambda t + \frac{\Lambda^2 t^2}{2!} + \frac{\Lambda^3 t^3}{3!} + \dots \right) W \\ &= Ve^{\Lambda t}W. \end{aligned} \quad (2.85)$$

Substituindo a forma exponencial final da Eq. (2.86) na Eq. (2.84), tem-se

$$G(t) = CVe^{\Lambda t}WB. \quad (2.86)$$

Considerando que todos os autovalores  $\Lambda$  são distintos e sabendo-se de (2.75) que  $\beta'_i = w'_i B$  e de (2.73) que  $\gamma_i = Cv_i$ , (2.86) que pode ser expressa como:

$$G(t) = \sum_{i=1}^n \gamma_i e^{\Lambda t} \beta'_i. \quad (2.87)$$

Devido à clareza com que  $G(t)$  em (2.87) salienta o acoplamento da saída com os vetores coluna  $\gamma_i$ ; e da entrada com os vetores linha  $\beta'_i$ , esta equação é útil para mostrar o efeito dos modos na controlabilidade e na observabilidade do sistema dinâmico.

Quando vistos através da matriz de saída  $C$ , os vetores modais  $\{v_i\}_{i=1}^n$ , são representados pelos vetores coluna  $\{\gamma_i\}_{i=1}^n$ . Se qualquer vetor  $\gamma_i$  tende para zero, o  $i$ -ésimo modo é não observável, caso contrário é observável; se todos os modos são observáveis o sistema é totalmente observável. Desta maneira, verifica-se a observabilidade dos modos e do sistema através do acoplamento dos vetores modais com as linhas da matriz de saída.

De forma similar, quando vistos através da matriz de entrada  $B$  os vetores modais recíprocos  $\{w_i\}_{i=1}^n$  são representados pelos vetores linha  $\{\beta'_i\}_{i=1}^n$ . Se qualquer vetor  $\beta'_i$  tende para zero, o  $i$ -ésimo modo é não controlável, caso contrário é controlável; se todos os modos são controláveis o sistema é totalmente controlável. Desta maneira verifica-se a controlabilidade dos modos e do sistema através do acoplamento dos vetores modais recíprocos com as colunas da matriz de entrada.

### 2.3.2 Sensibilidade Individual dos Autovalores

Os sistemas multivariáveis também são avaliados com relação à sensibilidade dos autovalores. A sensibilidade individual dos autovalores é expressa em termos dos autovetores à direita e à esquerda. É possível alocar a autoestrutura de forma que obtenha-se autovalores de baixa sensibilidade (LIU; PATTON, 1998). A sensibilidade individual do  $i$ -ésimo autovalor de uma dada matriz é expressa por

$$\eta_i = \frac{\|L_i\|_2 \|v_i\|_2}{\langle L_i^T, v_i \rangle}, \quad (2.88)$$

sendo  $v_i$  e  $L_i$  os autovetores à direita e à esquerda, respectivamente,  $\|L_i\|_2 \|v_i\|_2$  a norma-2 dos  $i$ -ésimos autovetores,  $\langle L_i^T, v_i \rangle$  o produto interno dos autovetores.

Autovalor de baixa sensibilidade produz variação relativamente pequena em sua posição quando sujeito a perturbações. Um autovalor é considerado perfeitamente condicionado quando a sensibilidade for igual a unidade, uma vez que gera menor mudança na posição do autovalor.

### 2.3.3 Decomposição em Valores Singulares

A Decomposição em Valores Singulares (SVD) é muito útil para a análise de sistemas multivariáveis. Os valores singulares máximos e mínimos são muito usados no estudo da robustez e desempenho de sistemas MIMO no domínio da frequência (PINTO, 2012).

Considerando a frequência fixa  $\omega$  onde  $G(j\omega)$  é uma matriz complexa  $p \times m$ , e denota-se  $G(j\omega)$  simplesmente por  $G$ , qualquer matriz  $G$  pode ser decomposta em valores

singulares (SKOGESTAD; POSTLETHWAITE, 2005),

$$G = U\Sigma V^H, \quad (2.89)$$

sendo

- $\Sigma$  é uma matriz  $p \times m$  com  $k = \min\{p, m\}$  valores singulares não negativos,  $\sigma_i$ , arranjados em ordem decrescente ao longo da diagonal principal;
- $G^H$  é a transposta conjugada complexa de  $G$ , os valores singulares são as raízes quadradas dos autovalores de  $G^H G$ :

$$\sigma_i(G) = \sqrt{\lambda_i G G^H}. \quad (2.90)$$

- $U$  é uma matriz unitária  $p \times p$  de vetores singulares de saída,  $u_i$ ;
- $V$  é uma matriz unitária  $m \times m$  de vetores singulares de entrada,  $v_i$  (SKOGESTAD; POSTLETHWAITE, 2005) (ALBERTOS; ANTONIO, 2006).

Pode-se obter informações importantes sobre as propriedades dos sistemas MIMO com a interpretação do gráfico dos valores singulares. Em particular, quantifica a banda de ganho da planta em cada frequência. Isto é uma generalização natural da informação contida na magnitude do gráfico de Bode clássico para plantas SISO. A principal diferença aqui é que, no caso multivariável, a banda de ganho é descrita por duas curvas, não uma.

1. A planta tem um ganho alto em  $\omega$  se  $\sigma_{\min}G(j\omega) \gg 1$ ,
2. A planta tem um ganho baixo em  $\omega$  se  $\sigma_{\max}G(j\omega) \ll 1$ .

Observa-se que os limites para garantir a robustez de sistemas MIMO são dados em termos do mínimo valor singular do ganho de malha que dever ser grande nas baixas frequências para garantir a robustez de desempenho e o máximo valor singular de ganho de malha deve ser baixo nas altas frequências para garantir a robustez de estabilidade. Portanto, o controlador projetado satisfaz as condições de desempenho e estabilidade no domínio da frequência.

Se a sensibilidade  $S(j\omega)$  é pequena, como descrito em baixas frequências, então:

$$\sigma_{\max}(S) = \sigma_{\max}[(I + GK)^{-1}] \approx \frac{1}{\sigma_{\min}(GK)}. \quad (2.91)$$

Isto é, um grande valor  $\sigma_{\min}(GK)$  garante um pequeno valor para  $\sigma_{\max}(S)$ . Por outro lado, se a cosensibilidade  $T(j\omega)$  é pequena, como é desejado em altas frequências, então

$$\sigma_{\max}(T) = \sigma_{\max}[GK(I + GK)^{-1}] \approx \sigma_{\max}(GK). \quad (2.92)$$

Ou seja, um pequeno valor de  $\sigma_{max}(GK)$  garante um valor pequeno de  $\sigma_{max}(T)$ . Isto significa que especificações de  $S(j\omega)$  é pequena em baixas frequências e  $T(j\omega)$  é pequena em altas frequências, podendo igualmente ser formulado em termos de  $\sigma_{min}(GK)$ , sendo grande em baixas frequências e  $\sigma_{max}(GK)$  sendo pequena em altas frequências. Assim, todas as especificações de desempenho serão em termos do máximo e mínimo valores singulares de ganho de malha  $GK(j\omega)$ . O significado prático disto é que precisa-se somente calcular os valores singulares do ganho de malha  $GK(j\omega)$ , e não os de  $S(j\omega)$  e  $T(j\omega)$ , que satisfazem  $S(j\omega) + T(j\omega) = I$ ,

$$e(s) = S(s)(r(s) - n(s)) + T(s)n(s). \quad (2.93)$$

Para baixas frequências, considera-se o ruído da Equação (2.93) igual a zero. Para manter o erro pequeno é necessário assegurar que, em cada frequência,  $\sigma_{max}(S(j\omega))$  é pequeno, o que pode ser garantido se for selecionado

$$\sigma_{min}(GK(j\omega)) \gg 1. \quad (2.94)$$

Já para altas frequências, o ruído do sensor faz-se presente acima de alguns valores de frequências conhecidas, logo de acordo com a Eq. (2.93), para manter o erro pequeno deve-se assegurar que

$$\sigma_{max}(GK(j\omega)) \ll 1. \quad (2.95)$$

como pode se observar na Figura 2.

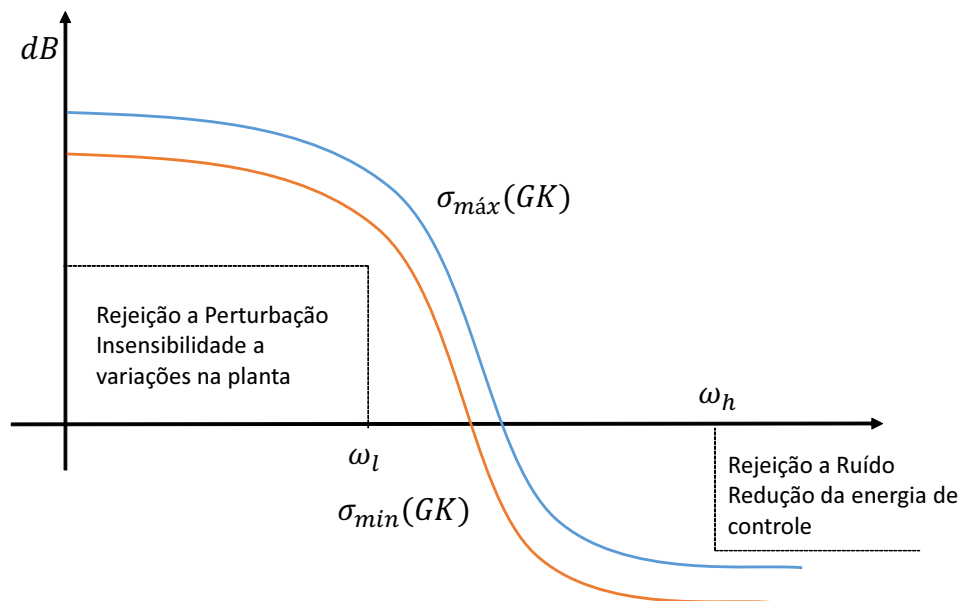


Figura 2 – Resposta em Frequência desejável para um sistema multivariável (PINTO, 2012).

## Capítulo 3

# Redes Neurais Artificiais SLP e Estruturadas para Alocação Online de Autoestruturas

Os sistemas de equações lineares desempenham um papel importante em muitas aplicações, tais como processamento de imagens, controle automático e robótica. Entretanto, na prática de engenharia é difícil um sistema ter equações lineares bem-condicionadas, os métodos iterativos raramente são usados devido a algumas restrições na matriz de coeficientes. Motivado pelas discussões acima, o objetivo neste capítulo é apresentar um método para resolver o problema de AAE formulado com base em sistemas de equações lineares. Numerosos problemas científicos e de engenharia exigem resolução de equações lineares simultâneas. Quando o número de variáveis desconhecidos é grande, a tarefa se torna computacionalmente intensiva e alguns métodos existentes não são eficientes (HANG et al., 1997).

Neste capítulo, é proposto um método bioinspirado para alocação online de autoestruturas, as redes neurais apresentadas tem por base a resolução de sistemas de equações lineares. A ideia básica dessa abordagem é representar um determinado problema em uma arquitetura de rede estruturada; em seguida, treinar a RNA-E para corresponder a alguns padrões desejados, e finalmente, obter a solução para o problema a partir dos pesos resultantes (WANG; MENDEL, 1990).

### 3.1 Redes Neurais Artificiais

As redes neurais são definidas por (HAYKIN et al., 2009) como um processador maciçamente paralelamente distribuído constituído de unidades de processamento simples, que tem a propensão natural para armazenar conhecimento experimental e torná-lo dispo-



nível para o uso. Ao contrário dos modelos matemáticos tradicionais, que são programados, as RNA aprendem as relações entre entradas e saídas selecionadas.

O neurônio é a unidade básica de uma RNA, estes são células eletricamente excitáveis e podem gerar sinais elétricos através do seu corpo celular, estes sinais são multiplicados por um coeficiente (peso) e então somados ao corpo da célula, de modo que, quando este somatório atinge um valor limiar, o neurônio "dispara" e um sinal conhecido como o potencial de ação é enviado através do axônio para as terminações sinápticas. (KAIADI, 2006).

O cérebro é um computador complexo, não linear e paralelo. Ele tem a capacidade de executar tarefas como reconhecimento de padrões, percepção e controle motor muito mais rápido do que qualquer computador - mesmo que os eventos ocorram na faixa de nanossegundos para portas de silício e milissegundos para sistemas neurais. Além dessas características, outras, como a capacidade de aprender, memorizar e ainda generalizar, estimularam pesquisas em modelagem algorítmica de sistemas neurais biológicos - denominadas redes neurais artificiais (RNA) (ENGELBRECHT, 2007).

### 3.1.1 Neurônio Artificial

O "neurônio básico" é abstraído de um neurônio motor da medula espinhal de mamíferos. Do soma (corpo celular) sobressai um número de ramificações chamados dendritos; o soma e os dendritos constituem a superfície de entrada do neurônio. Há também a extrusão do corpo da célula, em um ponto chamado de axônio (que fica próximo ao segmento inicial), uma fibra longa chamada axônio, cujos ramos formam a arborização axonal. As pontas dos ramos do axônio, chamadas terminais nervosas, atingem outros neurônios ou efetores. O locus da interação entre uma terminação nervosa e a célula na qual ele colide é chamado de sinapse, e dizemos que a célula com a terminação nervosa faz as sinapses na célula com a qual a conexão é feita. De fato, os ramos axonais de alguns neurônios podem ter muitas varicosidades, correspondentes às sinapses, ao longo de seu comprimento, não apenas no final do ramo (ARBIB, 2002).

Um neurônio é uma unidade de processamento de uma rede neuronal. Não existe um neurônio típico, ou seja, uma RNA pode ser interligada de diferentes formas, onde as conexões oriundas dos dados de entrada ou de saída de outros neurônios são a base para novas estruturas, estes são denominados pesos sinápticos que podem ser fixos ou ajustáveis, em um somador para efetuar a soma de todas as conexões de entrada do neurônio e em uma função de ativação responsável por limitar a saída do neurônio (HAYKIN et al., 2009). A Figura 3 mostra uma analogia entre um neurônio biológico e um neurônio artificial, com os seus quatro componentes principais.

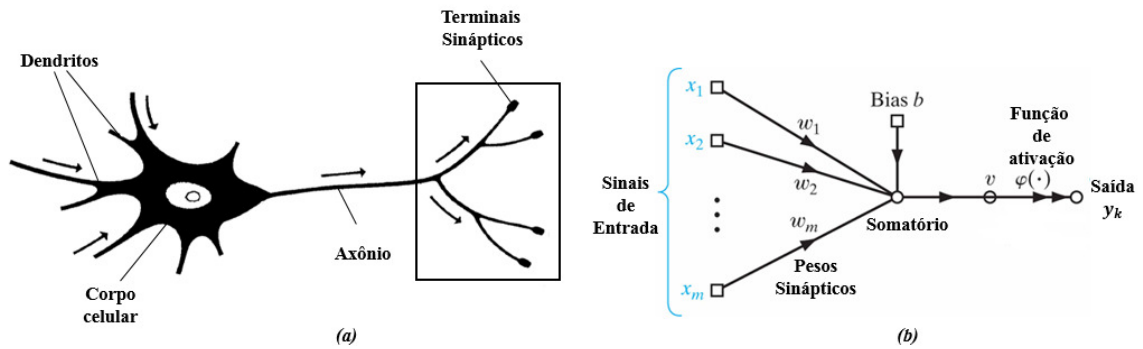


Figura 3 – Estrutura generalizada de um neurônio. (a) Neurônio Natural (ARBIB, 2002). (b) Neurônio artificial (HAYKIN et al., 2009).

As entradas da RNA são representadas por  $x_m$ , e cada uma delas é multiplicada pelos pesos das conexões especificadas, que são representados por  $w_m$ . Há uma entrada extra denominada de *Bias* ( $b_k$ ), esta é responsável pela polarização, permitindo que o neurônio tenha uma saída diferente de zero. Na próxima etapa são realizadas algumas operações matemáticas e os produtos desta operação são alimentados através de funções de ativação não-linear ( $\varphi(\cdot)$ ) para gerar uma saída, se a entrada combinada exceder um limite, ela irá ativar e enviar uma saída (KAIADI, 2006), esta saída geralmente é escolhida entre os intervalos 0 e 1 ou -1 e 1. Como a derivada da função de ativação é frequentemente usada no treinamento da rede, é conveniente que a derivada seja expressa em termos do valor da função original, pois assim os cálculos adicionais são minimizados de forma a calcular a derivada desta função (HAYKIN et al., 2009).

O resultado produzido pelo neurônio artificial é sintetizado por duas equações que são dadas por

$$v_k = \sum_{i=1}^n w_i x_i + b_k, \quad (3.1)$$

e

$$y_k = \varphi(v_k). \quad (3.2)$$

As Eqs. (3.1) e (3.2) representam a rede neuronal *Single Layer Perceptron* que implementam as equações da lei de controle  $u = Kx$ .

### 3.1.2 Treinamento de uma Rede Neuronal

A aprendizagem consiste em ajustar os valores de peso e limiar até que um determinado critério (ou vários critérios) sejam satisfeitos (ENGELBRECHT, 2007). A característica mais importante de uma RNA é a de poder aprender modelos matemáticos experimentalmente, sem ter que determinar explicitamente as relações matemáticas que estão vinculadas com as soluções dos problema, conseqüentemente, a RNA não é pro-

gramada, mas sim "treinada" através de um processo de aprendizagem baseado em dados empíricos (BISHOP, 2006).

Existem três tipos principais de aprendizado: aprendizagem supervisionada, aprendizagem não supervisionada e aprendizagem por reforço.

Na aprendizagem supervisionada o neurônio é fornecido com um conjunto de dados que consiste em vetores de entrada e um alvo (saída desejada) associado a cada vetor de entrada. Esse conjunto de dados é referido como o conjunto de treinamento. O objetivo do treinamento supervisionado é então ajustar os valores de peso de modo que o erro entre a saída real, do neurônio e a saída alvo seja minimizado (HAYKIN et al., 2009) (BISHOP, 2006) (ENGELBRECHT, 2007).

A aprendizagem não supervisionada tem como objetivo descobrir padrões ou recursos nos dados de entrada sem assistência de uma fonte externa. Muitos algoritmos de aprendizado não supervisionados basicamente realizam um agrupamento dos padrões de treinamento.

Na aprendizagem por reforço o objetivo é recompensar o neurônio (ou partes de um neurônio) por um bom desempenho e penalizar o neurônio por mau desempenho.

### 3.1.3 Algoritmos de Aprendizado

As redes neurais utilizam algoritmos de aprendizado para aprender com os dados que foram armazenados. Os principais algoritmos de aprendizado são: descida mais íngreme (*Steepest Descent*), método de Newton, Gradiente Conjugado, método Quasi-Newton, algoritmo Levenberg-Marquardt e *back-propagation*. Dentre os principais algoritmos, o mais utilizado é o algoritmo *back-propagation*.

A fim de minimizar o custo de uma predição, é necessário analisar as entradas, a função custo e os pesos da rede, visto que contribuem para o cálculo do custo, em que alterações na entrada para obter uma saída correta está fora de questão, uma vez que o objetivo é fazer com que a rede use a resposta real do sistema para dar uma resposta aceitável. A função de ativação está fora do nosso controle quando a aprendizagem está em andamento, portanto, é o peso que precisa ser alterado para minimizar o custo (MARSLAND, 2015).

O algoritmo *back-propagation* geralmente é implementado de forma iterativa, o resultado desse processo de aprendizagem é passado para o neurônio de saída, onde este resultado de aprendizagem é comparado ao resultado desejado, se o erro não for suficientemente pequeno, o processo de aprendizagem na camada oculta continuará até que os resultados sejam satisfatórios (HAYKIN et al., 2009) (MARSLAND, 2015) (BISHOP, 2006).

Os métodos de aprendizagem resultam em uma regra que ajusta os pesos das conexões entre unidades, de acordo com alguma regra de modificação. Grande parte dos algoritmos de aprendizagem são derivados da regra de aprendizagem Hebbiana, esta foi sugerida por Hebb (HEBB, 1964). De acordo com a regra de Hebb, o ajuste aplicado no peso sináptico  $w_{kj}$  em um instante de tempo é dado por

$$\Delta w_{jk} = \mu y_k(n) x_j(n), \quad (3.3)$$

sendo  $y_k$  representa as atividades pós-sinápticas,  $x_j$  as atividades pré-sinápticas e  $\mu$  é uma constante positiva a qual determina a taxa de aprendizagem (HAYKIN et al., 2009). Outra regra de treinamento comumente empregada utiliza a diferença entre a entrada atual (pré-sináptico) e o sinal desejado ao invés de somente as atividades pré-sinápticas ( $x_j$ ), e é dada por

$$\Delta w_{jk} = \mu y_k(n) (d_j(n) - x_j(n)), \quad (3.4)$$

sendo  $d_j$  o sinal de ativação desejado fornecido por um "professor". Isto é chamado de Regra Delta e dentre uma de suas variações obtem-se a seguinte expressão

$$\Delta w_{jk} = \mu (d_j(n) - x_j(n)) y_k(n) x_j(n). \quad (3.5)$$

## 3.2 Redes Neurais Estruturadas

Muitos problemas encontrados em ciência e engenharia envolvem a solução de sistemas de equações algébricas lineares, por exemplo. em processamento de sinal e robótica. Em princípio, resolver sistemas de equações é equivalente a calcular a inversa (ou pseudo-inversa) de uma matriz. Nesta seção, pretende-se estender algumas das ideias para resolver sistemas de equações e no desenvolvimento de estratégias de processamento em tempo real ou online. Muitos excelentes métodos numéricos já existem. Normalmente, as restrições de tempo para resolver sistemas de equações offline não são importantes, mas boa solução é fundamental. No entanto, se for necessário resolver sistemas de equações online repetidamente, ou em tempo real, e se as restrições de tempo para resolver essas equações são mais exigentes do que um computador digital típico pode fornecer, é necessário explorar métodos alternativos. Uma possibilidade é usar arranjos sistólicos ou *wavefront* (LI; WAH, 1985)-(KUMAR, 1994). Outra abordagem é usar redes neurais artificiais analógicas por causa de suas arquiteturas paralelas inerentes (CICHOCKI; UNBEHAUEN, 1993),(CICHOCKI; UNBEHAUEN, 1992b) e (CICHOCKI; UNBEHAUEN, 1992a).

Motivado pelo exposto no parágrafo anterior, o objetivo nesta seção é apresentar o desenvolvimento de estratégias de aprendizado para solução online de sistemas de equações lineares. A filosofia básica é resolver certos problemas de álgebra matricial por meio redes neurais. Isto é, usar uma abordagem de neurocomputação para resolver um sistema de equações lineares pode ser computacionalmente vantajoso para sistemas que devem ser

repetidamente resolvidos quando certos parâmetros do sistema mudam muito lentamente com o tempo (que é a razão pela qual o sistema de equações deve ser repetidamente resolvido). Devido às pequenas variações de parâmetros do sistema, a "nova" solução não está "longe" da solução "antiga" e, como resultado, poucas etapas de "aprendizado" serão necessárias para alcançar a nova solução (CICHOCKI; UNBEHAUEN, 1993).

### 3.2.1 Sistemas de Equações Algébricas Lineares Simultâneas

Dado o conjunto de equações algébricas lineares com coeficientes constantes

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned}, \quad (3.6)$$

é desejado resolver para as incógnitas  $x_1, x_2, \dots, x_n$ , dados os coeficientes  $a_{ij}$  for  $i = 1, 2, \dots, m$  and  $j = 1, 2, \dots, n$  and  $b_{ij}$  for  $i = 1, 2, \dots, m$ . A Eq. (3.6) é escrita de forma mais conveniente (compacta), ou ainda na forma matricial

$$Ax = b, \quad (3.7)$$

sendo que  $A \in \mathfrak{R}^{m \times n}$ ,  $x \in \mathfrak{R}^{n \times 1}$ ,  $b \in \mathfrak{R}^{m \times 1}$ , e

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad (3.8)$$

$$x = [x_1, x_2, \cdots, x_n]^T \quad (3.9)$$

e

$$b = [b_1, b_2, \cdots, b_m]^T. \quad (3.10)$$

Existem três casos possíveis que podem ocorrer: (1) Se  $m > n$ , esta é uma situação comum (mais equações do que desconhecidas), e o sistema de equações é referido como sendo sobredeterminado (GOLUB; LOAN, 2013). (2) Se  $n > m$  (mais incógnitas do que equações), o sistema de equações é chamado de indeterminado (GOLUB; LOAN, 2013). (3) Quando  $m = n$ , existem tantas equações quanto incógnitas (sistema exatamente determinado). A Matriz  $A$  é algumas vezes chamada de matriz de dados, e o vetor  $b$  é ocasionalmente chamado de vetor de observação (GOLUB; LOAN, 2013).

### 3.2.2 Solução de Equações Lineares pelos Mínimos Quadrados

Assumindo o conjunto geral de equações lineares dado na Eq. (3.7) com  $m \geq n$  (sistema sobredeterminado), considere primeiro o problema de mínimos quadrados de posto completo (GOLUB; LOAN, 2013). Em outras palavras, assumimos que a matriz  $A$  possui posto completo,  $\rho(A) = n$ , e busca-se por uma solução  $x$ . Para alcançar isso, deve-se minimizar a função de custo escalar que é dada por

$$\varepsilon(x) = \|Ax - b\|_p, \quad (3.11)$$

para alguma norma  $p$  escolhida apropriadamente. Diferentes normas resultam em diferentes soluções ótimas. Se a norma 1 ( $L_1$ ) e norma  $\infty$  ( $L_\infty$ ) ou norma *Chebyshev* forem usadas, a minimização de (3.11) é difícil porque a função  $\varepsilon(x) = \|Ax - b\|_p$  não é diferenciável para estes valores de  $p$  (ZHANG, 1993). Quando  $p = 1$  é escolhido é chamado de desvios absolutos mínimos. Para  $p = 2$ , isso é chamado de problema dos mínimos quadrados lineares. E quando  $p = \infty$ , é chamado de problema *minimax*. O tipo de norma usado em (3.11) depende muito da distribuição de erros nos dados e do tipo de aplicação (HUBER, 1996).

Para o caso em que os erros têm uma distribuição exponencial dupla (ou distribuição de Laplace) (MOOD; GRAYBILL; BOES, 1974), a norma  $L_1$  pode ser usada. Quando a distribuição de erro tiver transições nítidas, por exemplo, uma distribuição uniforme, a norma  $L_\infty$  (ou norma *Chebyshev*) deve ser usada. No entanto, se a distribuição de erro for gaussiana (normal), é melhor usar a norma euclidiana (ou norma  $L_2$ ). Usando a norma euclidiana, o problema dos mínimos quadrados resultante

$$\min_{\forall x} \varepsilon(x) = \min_{\forall x} \|Ax - b\|_2, \quad (3.12)$$

é mais tratável que os outros casos. Se  $A$  tem posto completo, a Eq. (3.7) tem uma única solução (GOLUB; LOAN, 2013). A função de custo escalar que é normalmente usada pode ser escrita como

$$\varepsilon(x) = \frac{1}{2} \|Ax - b\|_2^2 = \frac{1}{2} \|e\|_2^2, \quad (3.13)$$

sendo

$$e = Ax - b \quad (3.14)$$

a solução do vetor de erro. Calculando o gradiente da Eq. (3.13) em relação ao vetor  $x$  e estabelecendo o resultado igual a zero, ou seja,  $\nabla_x \varepsilon(x) = 0$ , tem-se que

$$A^T Ax - A^T b = 0. \quad (3.15)$$

Resolvendo  $x$  diretamente a partir da Eq. (3.15), tem-se que

$$x = A^+ b, \quad (3.16)$$

em que

$$A^+ = (A^T A)^{-1} A^T, \quad (3.17)$$

é definida como a pseudoinversa de  $A$ . A solução apresentada na Eq. (3.16) é a solução "em batelada" para o problema (3.7). Se  $m < n$  (problema é subdeterminado) e  $\rho(A) = m$  (ou seja,  $A$  tem posto completo), a solução para a Eq. (3.7) é dada por

$$x = A^+ b, \quad (3.18)$$

sendo

$$A^+ = A^T (A A^T)^{-1}, \quad (3.19)$$

a pseudoinversa de  $A$ .

### 3.2.3 Resolução de Problemas Lineares usando a Abordagem Neural baseada em Mínimos Quadrados

Supõe-se que precisa-se de uma solução para a Eq. (3.7) que não seja baseada no método em batelada, mas sim em uma abordagem que possa "aprender" a solução. Um método simples é baseado no método de descida gradiente mais íngreme (HAM; KOSTANIC, 2001). A regra de aprendizado de tempo contínuo é dada por

$$\frac{dx}{dt} = -\mu \nabla_x \varepsilon(x), \quad (3.20)$$

sendo  $\mu \in \mathfrak{R}^{n \times n}$  ( $\mu = [\mu_{ij}]$  para  $i, j = 1, 2, \dots, n$ ) uma matriz positiva definida, geralmente escolhida como diagonal, ou seja,  $\mu_{ij} = \mu_j \delta_{ij}$ , sendo  $\mu_j > 0$ ,  $\delta_{ij}$  é o delta *Kronecker* e  $x(0) = x_0$ . O lado esquerdo da Eq. (3.16) é o cálculo do gradiente mostrado na Eq. (3.15). Substituindo-se este gradiente na Eq. (3.20), tem-se a regra de aprendizado para resolver um sistema de equações

$$\frac{dx}{dt} = -\mu (A^T A x - A^T b) = -\mu A^T (A x - b). \quad (3.21)$$

Substituindo a Eq.(3.14) na Eq.(3.21) obtém-se a regra de aprendizado em tempo contínuo

$$\frac{dx}{dt} = -\mu A^T e, \quad (3.22)$$

para resolver o sistema de Eqs. (3.7), com  $x(0) = x_0$  como as condições iniciais. As Eqs. (3.14) e (3.22) juntas são adequadas para resolver qualquer sistema de equações, independentemente do sistema ser exatamente determinado  $m = n$ , sobredeterminado ( $m < n$ ), subdeterminado ( $m > n$ ), ou se  $A$  tem posto incompleto, ou mesmo se  $A$  é mal-condicionada (HAM; KOSTANIC, 2001) (WANG; MENDEL, 1990). Muitas vezes é conveniente usar uma equação matricial normalizada  $\hat{A}x = \hat{b}$ , onde

$$\hat{a}_{ij} = \frac{a_{ij}}{\|a_i^T\|_2} \quad b_i = \frac{b_i}{\|a_i^T\|_2} \quad i = 1, 2, \dots, m \text{ e } j = 1, 2, \dots, n. \quad (3.23)$$

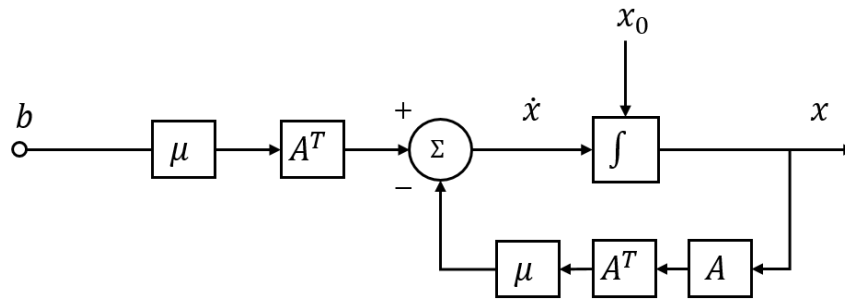


Figura 4 – Diagrama de blocos para realizar o cálculo dos mínimos quadrados usando uma abordagem de neurocomputação para resolver o sistema de equações lineares (HAM; KOSTANIC, 2001).

Uma arquitetura da rede neural estruturada é desenvolvida usando uma abordagem de mínimos quadrados para resolver sistemas de equações lineares. Um diagrama de blocos de um sistema diferencial é mostrado na Fig. 4, que é a base para a realização de um sistema contínuo da abordagem de neurocomputação.

Dado o vetor de erro definido na Eq. (3.14), essa formulação de vetor de erro é aplicável apenas para o caso determinado (ou seja,  $m = n$ ). Reformular-se a expressão de erro como

$$e = A^T Ax - A^T b \quad (3.24)$$

a fim de avaliar os erros durante o processo de aprendizagem para qualquer caso. Note que o lado direito da Eq. (3.24) sendo igual a zero resulta na Eq. (3.15). Usando a Eq. (3.24) na função de custo do erro

$$\varepsilon(x) = \frac{1}{2} \|e\|_2^2 = \frac{1}{2} e^T e. \quad (3.25)$$

e calculando o gradiente da Eq. (3.25) em relação a  $x$  fornece a nova regra de aprendizado em tempo contínuo da Eq. (3.20) como

$$\frac{dx}{dt} = -\mu e, \quad (3.26)$$

com  $x(0) = x_0$  como as condições iniciais. Portanto, as Eqs. (3.24) e (3.26) juntas fornecem uma regra geral de aprendizado para resolver um sistema de equações algébricas lineares simultâneas. Além disso, não é necessário fazer a suposição de posto completo, isto é,  $\rho(A) = \min(m, n)$ . Isso é óbvio a partir da própria regra de aprendizado.

A escolha dos elementos na matriz  $\mu$  é crítica por razões de estabilidade. Obviamente, é desejável assegurar que a Eq. (3.26), diferencial vetorial, seja estável. A escolha dos elementos em  $\mu$  também determina a velocidade de convergência para o estado de equilíbrio (isto é, a solução de  $Ax = b$ ) (HAM; KOSTANIC, 2001) (WANG; MENDEL, 1990). Na verdade, deriva-se diretamente as condições para a estabilidade da regra de aprendizado, isto é, a Eq. (3.26) vetorial-diferencial. Portanto, a derivada total da função de energia



pode ser calculada e é dada por

$$\dot{\varepsilon}(x) = \nabla_x^T \varepsilon(x) \frac{dx}{dt}. \quad (3.27)$$

No entanto, substituindo a forma geral da regra de aprendizagem em tempo contínuo dada através da Eq. (3.20) na Eq. (3.27), tem-se que

$$\dot{\varepsilon}(x) = \nabla_x^T \varepsilon(x) \mu \nabla_x \varepsilon(x). \quad (3.28)$$

Usando  $\nabla_x \varepsilon(x) = e$ , como explicado anteriormente na Eq. (3.28), obtém-se

$$\dot{\varepsilon}(x) = -e^T \mu e. \quad (3.29)$$

Assumimos que a matriz  $\mu$  é positiva definida, isto é,  $\mu > 0$ ; portanto

$$\dot{\varepsilon}(x) = e^T \mu e < 0. \quad \varepsilon \neq 0 \quad (3.30)$$

A Eq. (3.30) declara a derivada total da função de energia é sempre definida negativa. Portanto, a Eq. (3.26) assintoticamente estável.

A forma escalar da regra de aprendizagem pode ser escrita diretamente da Eq. (3.26) e (3.24) como

$$\frac{dx_i}{dt} = - \sum_{h=1}^n \mu_{ih} e_h \quad x_i(0) = x_{i0}, \quad (3.31)$$

para  $i = 1, 2, \dots, n$ . Novamente, a escolha dos parâmetros  $\mu_h$  deve ser feita cuidadosamente para assegurar a estabilidade das equações diferenciais e a velocidade de convergência razoável para o estado de equilíbrio (isto é, a solução de  $Ax = b$ ). Os termos  $e_h$  da Eq. (3.31) são os elementos individuais do vetor de erro dado na Eq. (3.24). Estes elementos escalares do vetor de erro podem ser escritos definindo primeiro a matriz

$$W \triangleq A^T A \quad W \in \mathfrak{R}^{n \times n}. \quad (3.32)$$

Na forma escalar a Eq.(3.32) é dada por

$$w_{hr} = \sum_{q=1}^m a_{qh} a_{qr}, \quad (3.33)$$

para  $h = 1, 2, \dots, n$  e  $r = 1, 2, \dots, n$ . Usando a Eq. (3.33), os elementos do vetor de erros dado na Eq. (3.24) pode ser escrito como

$$e_h = \sum_{r=1}^n w_{hr} x_r - \sum_{j=1}^m a_{jh} b_j. \quad (3.34)$$

### 3.3 Proposta de RNA-E para AAE

Nesta seção propõe-se duas RNA-E para alocação online de autoestrutura baseadas nos princípios e soluções de sistemas lineares descritos nas seções 3.2.2 e 3.2.3. Além da RNA para o controle  $u = Kx$ , o treinamento RNA-SLP é feito por meio da RNA-E. As entradas da rede são os estados do sistema e suas saídas são as leis de controle, o par entrada-saída é usado para gerar o erro do sistema.

A Fig. 5 mostra a configuração geral de um sistema em espaço de estados em malha fechada com uma RNA-SLP, esta rede irá fornecer de forma online uma matriz  $K$  de retroação de estado capaz de alocar a autoestrutura especificada. O algoritmo de treinamento depende da estimação paramétrica das matrizes que representam a dinâmica do sistema e também da autoestrutura desejada.

A arquitetura da rede neuronal que representa o controle online e bioinspirado para a alocação da autoestrutura por realimentação de estado é baseada na Eq. (3.35). A partir desta equação, um diagrama de blocos da arquitetura das  $n$  RNA-SLP é construído, o que é mostrado na Fig. 6.

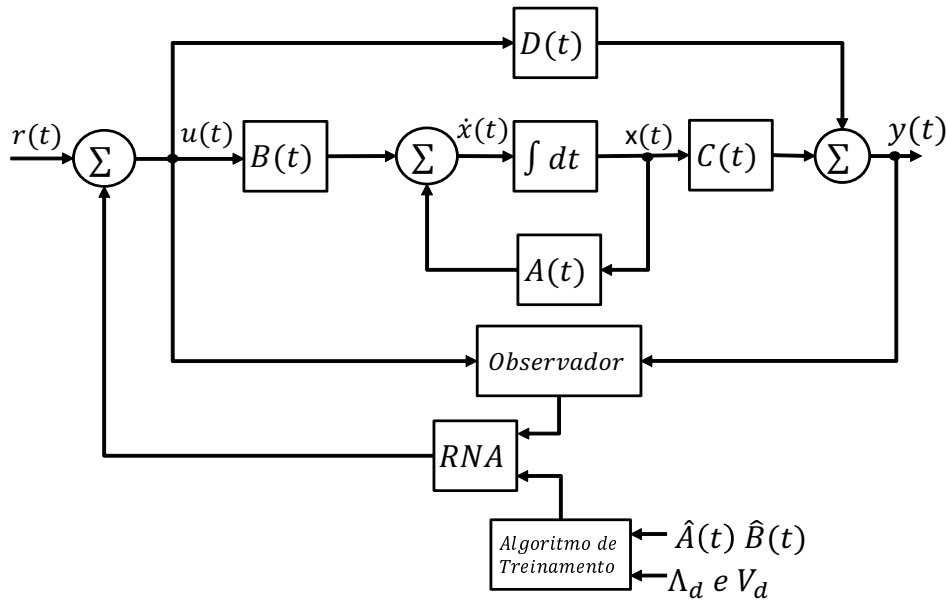


Figura 5 – Diagrama de blocos do sistema em espaço de estados, em malha fechada com a rede neuronal.

As entradas de controle da Eq. (2.6) são representadas em termos de equações escalares que são dadas por

$$u_i = -\langle K_i, x \rangle, \quad i = 1, \dots, q, \quad (3.35)$$

sendo  $K = [ K_{i1} \ K_{i2} \ \dots \ K_{in} ]^T \quad i = 1, \dots, q.$

A arquitetura da Rede Neuronal do Dispositivo de Controle da Fig. 6 é uma abstração para o desenvolvimento de sistema embarcado (GAJSKI et al., 2009) para

alocação online de autoestruturas baseada em aprendizado bio-inspirado para treinamento de redes neurais perceptrons de camada única. Nesta rede as entradas são os estados amostrados para intervalos de tempo  $T_s$ . Ainda, observa-se que a sua saída é comparada com a entrada de controle especificada  $U_{di}$ . Esta entrada é montada a partir de cada par  $(V_{di}, \lambda_i)$  da autoestrutura especificada. O algoritmo de treinamento da rede recebe o resultado da comparação entre o valor atual e o valor desejado. O subsídio para ajuste dos pesos sinápticos da rede é fornecido pela comparação entre esses dois sinais, considerando que o sinal de saída para um dado valor da função de ativação representa a atividade pós-sináptica e o sinal resultante do combinador linear é chamado de sinal pré-sináptico, conforme a teoria de RNA, que é apoiada pelos os postulados de Hebb (HAYKIN, 1999).

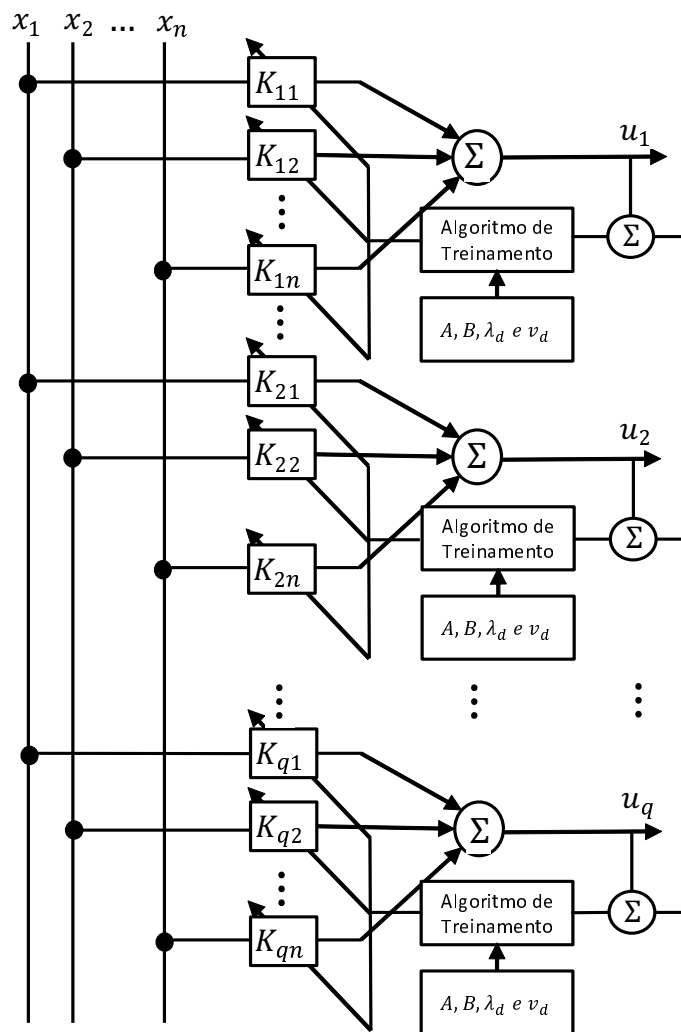


Figura 6 – Arquitetura do Dispositivo de Controle da RNA-E para Alocação de Autoestruturas.

Desta forma, percebe-se a demanda por novas arquiteturas e topologias que superem ou agreguem valor ao projeto de sistema de controle em malha fechada. Estes novos sistemas devem satisfazer a incerteza da operação (erros humanos), redução de custos na produção do produto final e variações do processo. Na Fig. 6 apresenta-se a arquitetura do sistema

de controle que propõe atender e agregar as qualidades de alto desempenho nos sistemas de controle.

O treinamento da RNA-SLP, que realiza a estrutura de controle por realimentação de estados, baseia-se na abordagem de rede neuronal estruturada RNA-E. A solução da RNA-E deve resolver  $n$  sistemas lineares do tipo

$$(A - BK)v_i = \lambda_i v_i. \quad (3.36)$$

A Eq.(3.36) em termos de um problema de espaço nulo (CHEN, 2013) é dada por

$$[(A - \lambda I) - B] \begin{bmatrix} v_i \\ W_i \end{bmatrix} = 0 \quad (3.37)$$

e

$$W_i = K_i v_i, \quad (3.38)$$

sendo  $W$  o vetor intermediário que permite linearizar o sistema de equações da estrutura da Eq.(3.36). Levando em consideração que a  $i$ -ésima autoestrutura especificada é representada pelo par

$$V \lambda_{di} = (v_{di}, \lambda_{di}), \quad (3.39)$$

sendo  $\{\Lambda_d, V_d\}$  a  $i$ -ésima autoestrutura especificada, e substituindo os valores, a equação da  $AE$  especificada em termos de  $W_i$  é dada por

$$(A - \lambda_{di} I)v_{di} - BW_i = 0. \quad (3.40)$$

Na rede neuronal estruturada, as entradas transformadas são dadas por  $W_i$  e as saídas são dadas pela transformação dos pares  $(v_{di}, \lambda_{di})$ .

A partir da Eq. (3.40), o processo iterativo para o treinamento da rede do dispositivo de controle, doravante denominado de ator no sentido que executa a decisão baseado no treinamento da rede neuronal do tipo *feedforward* via uma RNA-E. Uma das mais importantes equações para determinação da matriz de realimentação de estados é dada por

$$Bw_i = (A - \lambda_{di} I)v_{di}, \quad (3.41)$$

sendo  $B \in \mathfrak{R}^{n \times q}$  e  $w_i \in \mathfrak{R}^{q \times 1}$ .

A arquitetura desta rede de treinamento é proposta a partir da Eq. (3.41). A forma matricial, em termos dos seus coeficientes, é dada por

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nq} \end{bmatrix} \begin{bmatrix} W_{11} \\ W_{21} \\ \vdots \\ W_{q1} \end{bmatrix} = \begin{bmatrix} \bar{a}_{11} \\ \bar{a}_{21} \\ \vdots \\ \bar{a}_{2n} \end{bmatrix}, \quad (3.42)$$

sendo  $\bar{a} = (A - \lambda_d I)$ .

Na forma simplificada o sistema de equações lineares em  $W$  é representado por

$$BW = \bar{a}. \quad (3.43)$$

A partir da equação matricial (3.43), que representa a rede para treinamento dos pesos sinápticos do dispositivo de controle, constrói-se a arquitetura e apresenta-se a proposta de um algoritmo para treinamento desta rede. A rede estruturada que tem como entrada o vetor de singularidades do par  $(V_{di}, \lambda_i)$  e como saída o vetor de controle  $K$ .

A regra de aprendizado da RNA estruturada proposta baseai-se no método do gradiente (*Steep Descent*), a regra de aprendizado para esta rede é dada pela variação de  $\Delta W$  que é dada por (HAYKIN, 1999)

$$\frac{dW(t)}{dt} = -\mu \nabla_W \varepsilon(W), \quad (3.44)$$

sendo  $\mu$  a taxa de treinamento que pertence ao espaço de matrizes  $\mu \in \mathfrak{R}^{q \times n}$  e  $\mu$  é positiva definida. De acordo com as Equações (3.44) e (3.42), o gradiente para o nosso problema, conforme estipulado na Eq. (3.44) é dado por

$$\frac{dW(t)}{dt} = -\mu B^T B W - B^T \bar{a}_i \quad (3.45)$$

De acordo com as Eqs. (3.43) - (3.45) o erro expresso em termos dos parâmetros do sistema e suas transformações é dado por

$$\frac{dW(t)}{dt} = -\mu B^T e, \quad (3.46)$$

sendo  $e$  o erro que é dado por

$$e = BW - \bar{a}_i, \quad (3.47)$$

e as condições iniciais dadas por  $W(0) = W_0$ .

A Fig. 7 tem por base a estrutura de resolução dos mínimos quadrados, conforme ilustrado na Fig. 4. A Fig. 7 apresenta uma estrutura em diagrama de blocos da rede neuronal estruturada para o cálculo da matriz  $W$ .

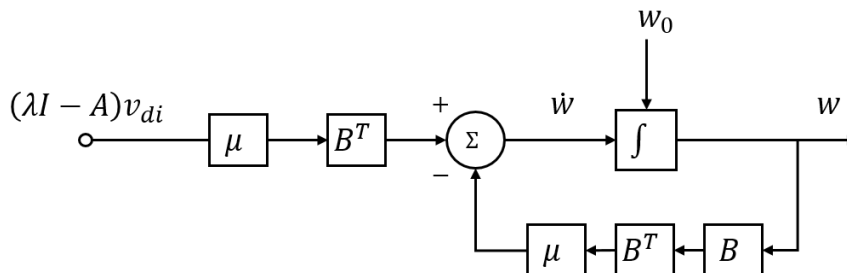


Figura 7 – Diagrama de blocos da rede neuronal para o cálculo do matriz de ganhos  $W$

Aplicando a transposta conjugada na Eq. (2.16), a formulação do problema para determinação do ganho é dada por

$$V_d^T K^T = W^T, \quad (3.48)$$

A determinação do ganho  $K$  é realizado através da Eq. (3.48) que depende do cálculo de  $W$ . A determinação de  $K$  pelo  $LS$  em batelada envolve o cálculo da pseudoinversa, que devido a problemas de mal condicionamento torna o processo não confiável, podendo não apresentar solução. Para contornar estes problemas, a determinação do ganho  $K$  é representado na forma de sistemas lineares pela direita.

Na forma de elementos, a matriz de ganhos é dada por

$$K = [ K_{ij} ], \quad i = 1, \dots, q, \quad j = 1, \dots, n,$$

sendo

$$K \in \mathfrak{R}^{q \times n}. \quad (3.49)$$

O mesmo procedimento utilizado para a formulação do erro da primeira rede neuronal é utilizado para o cálculo do erro da segunda rede, o mesmo é dado por

$$\varepsilon_K = a_K K_{rna} - b_{Ki}, \quad (3.50)$$

sendo  $a_K = V_d^T$  e  $b_{Ki} = w_{rni}$

O gradiente do erro é dado por

$$\nabla \varepsilon_{iK} = a_K^T \varepsilon_K, \quad (3.51)$$

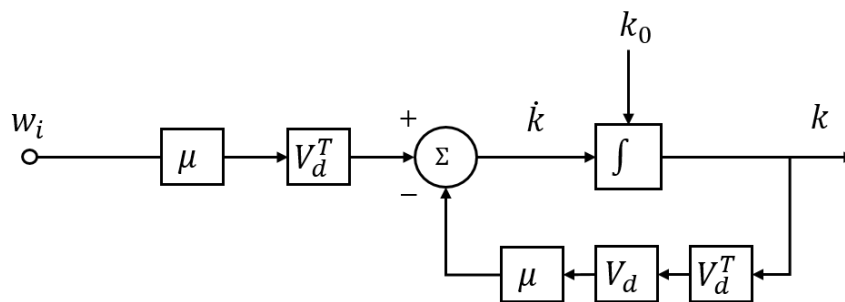


Figura 8 – Diagrama de blocos da rede neuronal para o cálculo do matriz de ganhos  $K$

### 3.3.1 Algoritmo RNA-E

O algoritmo  $EA-RNA-E$  para alocação de autoestuturas é construído por blocos funcionais, estes blocos são desenvolvidos visando as abordagens de sistemas embarcados de (GAJSKI et al., 2009) e (LEE; SESHIA, 2016).

ALGORITMO - *EA – RNA – E*

---

1 ▷ Modelo do Sistema Dinâmico

---

2 ▷ Autoestrutura Desejada

---

3 ▷ *EA – RNA – Core*

---

Primeiramente são inseridas as condições iniciais e as matrizes em espaço de estado do sistema ( $A$ ,  $B$ ,  $C$  e  $D$ ), em seguida é especificada a Autoestrutura desejada ( $V_d$ ,  $\Lambda_d$ ) e então é executado o algoritmo *EA – RNA – Core*.

O algoritmo *EA – RNA – Core* é constituído por duas RNA-E. A primeira rede determina a matriz  $W$  e a outra determina o ganho da realimentação de estados  $K$ .

Devem ser inicializados todos os parâmetros necessários para a execução das redes neuronais. A primeira rede neuronal que obtém a matriz de ganhos  $W$ . Os parâmetros a serem inicializados são: as matrizes da dinâmica da planta  $A$  e do controle  $B$ , a taxa de aprendizado  $\mu$  da RNA-E, o vetor inicial do peso da rede  $w_{rna}(0)$ , o erro inicial  $\varepsilon_0$  e o gradiente do erro  $\nabla\varepsilon_0$ .

O primeiro laço do processo iterativo é iniciado na linha 8 , sendo executado até o número  $n_\lambda$ , ou seja, o número de autovalores do sistema, na linha 10 é calculado o parâmetro  $\bar{a}_i$ , o mesmo depende do primeiro par da estrutura especificada e também da matriz que representa a dinâmica do sistema. Na linha 12, o segundo processo iterativo começa. Depois disso, os pesos sinápticos da rede, o erro e o gradiente são calculados, e então o peso sináptico da rede neuronal é atualizado até que o erro seja minimizado, obtendo assim  $W_{rna}$ , que é a representação do autovetor desejado em relação a matriz de ganhos de realimentação de estados  $K$ .

Para inicializar a segunda rede neuronal, devem-se carregar os parâmetros iniciais que são: variável auxiliar  $a_K$ , taxa de aprendizado da rede  $\mu$ , vetor inicial dos pesos da rede  $K_{rna}(0)^T$ , erro inicial  $\varepsilon_{0K}$  e gradiente do erro  $\nabla\varepsilon_{0K}$ .

Na sequência, na linha 25 do algoritmo, um processo iterativo é iniciado, sendo executado até o número de entradas do sistema  $n_{ent}$ . Na linha 27, o parâmetro  $b_{Ki}$  é armazenado, assim como as colunas da matriz  $W_{rna}$  em cada iteração. Na linha 30 começa o último processo iterativo, este é responsável por obter a matriz de ganhos  $K_{rna}$ . Em seguida são calculados os pesos sinápticos da rede, o erro, o gradiente e, em seguida, o peso sináptico da rede neuronal é atualizado até que o erro seja minimizado, obtendo assim  $K_{rna}$ .

Algoritmo EA – RNA – Core

---

```

1  ▷ Configuração - Inicialização
2  ▷ Inserir matrizes dinâmicas - A e B
3  ▷ Inicializar -  $\mu$ ,  $W_{rna}(0)$ ,  $V_d$  e  $\Lambda_d$ 
4  ▷ Inicializar -  $\bar{a}_i \leftarrow (\lambda_{di}I - A)v_{di}$ 
5  ▷ Inicializar -  $\varepsilon_0 \leftarrow BW_{rna_i} - \bar{a}_i$ 
6  ▷ Inicializar -  $\nabla\varepsilon_i \leftarrow B^T\varepsilon_0$ 

```

---

```

7  ▷ Processo Iterativo
8  para  $n \leftarrow 1 : n_\lambda$ 
9      faça
10          $\bar{a}_n \leftarrow (\lambda_{dn}I - A)v_{dn}$ 
11         para  $i \leftarrow 1 : n_{iter_w}$ 
12             faça
13                  $w_{rna}(i) \leftarrow w_{rna}(i-1) + \mu\nabla\varepsilon_i$ 
14                  $\varepsilon_i \leftarrow Bw_{rna}(i) - \bar{a}_i$ 
15                  $\nabla\varepsilon_i \leftarrow B^T\varepsilon_i$ 
16             fim
17 fim
18 ▷ Fim do Processo Iterativo
19  $W_{rna} = [ w_{ij} ]$ ,  $i = 1, \dots, q$ ,  $j = 1, \dots, n$ 

```

---

```

20 ▷ Inicializar -  $a_K \leftarrow V_d^T$ 
21 ▷ Inicializar -  $\mu$ ,  $K_{rna}(0)^T$ 
22 ▷ Inicializar -  $\varepsilon_{K_0} \leftarrow a_K K_{rna} - b_{K_n}$ 
23 ▷ Inicializar -  $\nabla\varepsilon_{K_0} \leftarrow a_{K_0}^T \varepsilon_{K_0}$ 

```

---

```

24 ▷ Processo Iterativo
25 para  $n \leftarrow 1 : n_{iter_k}$ 
26     faça
27          $b_{K_n} \leftarrow W_{rna_n}$ 
28         para  $i \leftarrow 1 : n_{iter_k}$ 
29             faça
30                  $k_{rna}(i) \leftarrow k_{rna}(i-1) + \mu\nabla\varepsilon_i$ 
31                  $\varepsilon_K \leftarrow \bar{a}_{K_i} k_{rna} - b_{K_i}$ 
32                  $\nabla\varepsilon_{K_i} \leftarrow \bar{a}_{K_i}^T \varepsilon_{K_i}$ 
33             fim
34 fim
35 ▷ Fim do Processo Iterativo
36  $K_{rna} = [ k_{ij} ]$ ,  $i = 1, \dots, q$ ,  $j = 1, \dots, n$ 

```



## Capítulo 4

# Estudos de Caso e Experimentos Computacionais

Os procedimentos para avaliar o desempenho dos algoritmos para alocação de autoestruturas são apresentados neste capítulo, levando em consideração a inicialização do processo iterativo, avaliando os resultados obtidos através dos métodos clássicos, tendo como referência a complexidade computacional e a estabilidade numérica. Assim, os testes computacionais foram realizados via *MATLAB* e, em seguida, foram realizadas comparações para avaliar a precisão das soluções fornecidas por esses algoritmos.

Os experimentos são realizados por meio de simulações e são aplicados em dois sistemas dinâmicos MIMO para avaliar a funcionalidade dos algoritmos. No primeiro sistema, os resultados do controle das tensões nas fontes de alimentação de um circuito elétrico de quarta ordem (DESOER; KUH, 1988) serão enfatizados. No segundo, os resultados do controle lateral-direcional de uma aeronave, especificamente, o projeto de uma aeronave F-16 (ALBOSTAN; GÖKAŞAN, 2018) que representa um sistema multivariável de sexta ordem. Os sistemas são avaliados através da análise modal e decomposição em valores singulares sob a ótica de sistemas multivariáveis.

A Fig. 9 apresenta a metodologia utilizada para a realização dos experimentos computacionais, sendo que primeiramente são calculados por meio do método da projeção o ganho de realimentação que aloca a autoestrutura desejada, a partir dos algoritmos AAE-LS-MP1 e AAE-LS-MP2, sendo que o primeiro está relacionado ao caso em que a autoestrutura apresenta somente autovalores reais, enquanto que o segundo algoritmo é utilizado para o caso em que a autoestrutura apresenta autovalores complexos. Após serem alocados pelo método da projeção, utiliza-se o a alocação por redes neuronais estruturadas e SLP. E por fim, os sistemas multivariáveis são analisados pelos métodos de sensibilidade individual dos autovalores, decomposição em valores singulares e análise modal.

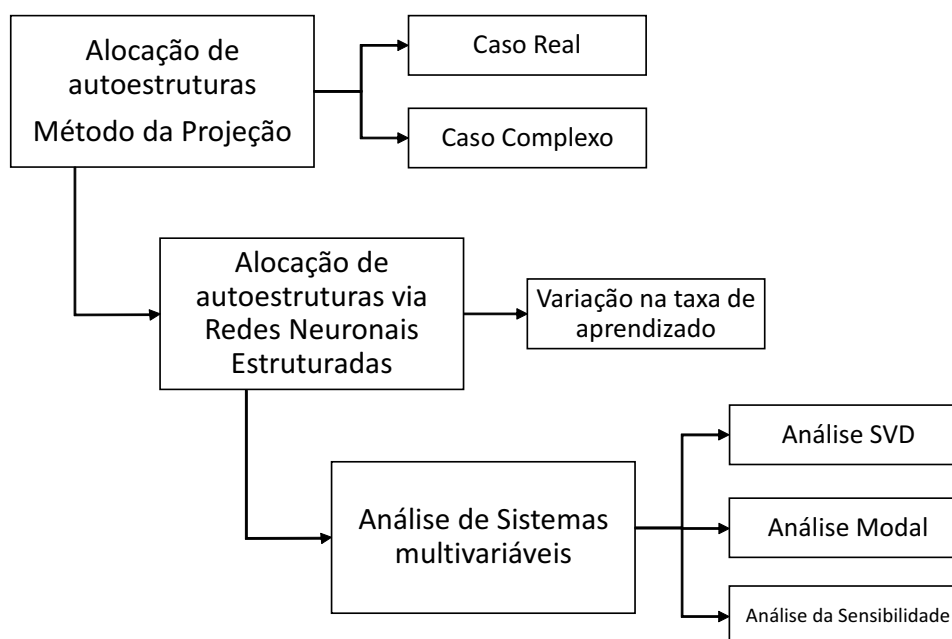


Figura 9 – Metodologia aplicada aos experimentos computacionais.

## 4.1 Circuito Elétrico de 4ª ordem

O circuito elétrico passivo de 4ª ordem e a disposição dos seus elementos são apresentados na Fig. 10. Este circuito é constituído por capacitores, indutores e resistores.

A lei das correntes é aplicada para relacionar as variáveis de tensão e corrente, as fontes de energia do sistema e os elementos do circuito de parâmetros concentrados. O comportamento do sistema é modelado por um conjunto de equações diferenciais e algébricas, (DESOER; KUH, 1988).

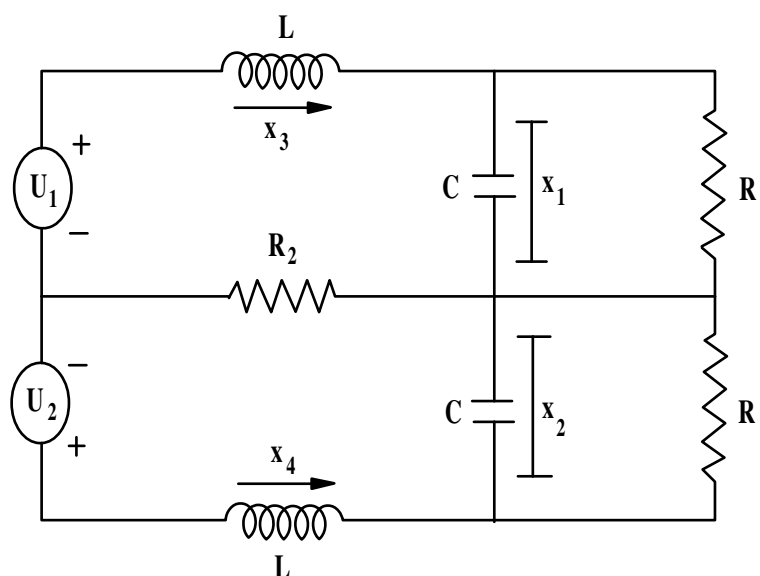


Figura 10 – Diagrama esquemático do circuito elétrico de 4ª.

Aplicando as leis de *Kirchhoff* das correntes e das tensões, obtém-se as equações diferenciais que representam o referido circuito elétrico. Denotando por  $x_1$  e  $x_2$  as tensões nos capacitores,  $x_3$  e  $x_4$  as correntes nos indutores, e aplicando a lei de *Kirchhoff* das correntes, obtém-se

Nó 3:

$$\begin{aligned}x_3 - C \frac{dx_1}{dt} - \frac{x_1}{R_1} &= 0 \\C \frac{dx_1}{dt} &= -\frac{x_1}{R_1} + x_3 \\ \frac{dx_1}{dt} &= -\frac{x_1}{CR_1} + \frac{x_3}{C}\end{aligned}\tag{4.1}$$

Nó 4:

$$\begin{aligned}x_4 - C \frac{dx_2}{dt} - \frac{x_2}{R_1} &= 0 \\C \frac{dx_2}{dt} &= -\frac{x_2}{R_1} - x_4 \\ \frac{dx_2}{dt} &= -\frac{x_2}{CR_1} - \frac{x_4}{C}\end{aligned}\tag{4.2}$$

Aplicando a lei de *Kirchhoff* das tensões, tem-se

Laço 1:

$$\begin{aligned}u_1 - L \frac{dx_3}{dt} - x_1 - R_2(x_3 + x_4) &= 0 \\ \frac{dx_3}{dt} &= -\frac{x_1}{L} - \frac{R_2}{L}x_3 - \frac{R_2}{L}x_4 + \frac{u_1}{L} = 0\end{aligned}\tag{4.3}$$

Laço 2:

$$\begin{aligned}u_2 - L \frac{dx_4}{dt} - x_2 - R_2(x_3 + x_4) &= 0 \\ \frac{dx_4}{dt} &= -\frac{x_2}{L} - \frac{R_2}{L}x_3 - \frac{R_2}{L}x_4 + \frac{u_2}{L} = 0\end{aligned}\tag{4.4}$$

#### 4.1.1 Descrição em Espaço de Estados

O modelo do sistema multivariável descrito em espaço de estados por meio das Eqs. (4.1)-(4.4) é dado por

$$\dot{x} = Ax + Bu, \quad t \geq t_0,\tag{4.5}$$

sendo  $A^{4 \times 4}$  a matriz de estado é dada por

$$A = \begin{bmatrix} -1/CR_1 & 0 & 1/C & 0 \\ 0 & -1/CR_1 & 0 & -1/C \\ -1/L & 0 & -R_2/L & -R_2/L \\ 0 & 1/L & -R_2/L & -R_2/L \end{bmatrix},\tag{4.6}$$

sendo  $R_1 = 1\Omega$ ,  $R_1 = R_2$ ,  $C = 0.1\mu F$  e  $L = 1H$ .

A matriz de entradas  $B^{4 \times 2}$  dada por

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/L & 0 \\ 0 & 1/L \end{bmatrix}, \quad (4.7)$$

e, a partir da Eq. (2.5), constrói-se a matriz de saída  $C^{2 \times 4}$ ,

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad (4.8)$$

as saídas do sistema são as tensões nos capacitores.

#### 4.1.2 Alocação da Autoestrutura através do Método da Projeção

Com a finalidade de validação dos resultados obtidos através das redes neuronais, são calculadas por meio do método dos mínimos quadrados e projeção a matriz de representação dos autovetores admissíveis  $W_{LS}$  e a matriz de ganhos de realimentação de estados  $K_{LS}$ , conforme algoritmo AAE-LS-MP1, apresentado na Seção 2.2.

As etapas necessárias para AAE por meio do método da projeção são as inicializações das matrizes dinâmicas dadas nas Eqs. (4.6) e (4.7), em seguida é especificada a autoestrutura desejada que são dadas por

$$V = \begin{bmatrix} 0,4962 & 0,5012 & -0,5038 & 0,4987 \\ -0,4962 & 0,5012 & 0,5038 & 0,4987 \\ -0,5038 & -0,4987 & -0,4962 & 0,5012 \\ -0,5038 & 0,4987 & -0,4962 & 0,5012 \end{bmatrix}, \quad (4.9)$$

e

$$\Lambda_d = \text{diag} \left[ -3,2859 \quad -4,4819 \quad -8,4165 \quad -8,1285 \right]. \quad (4.10)$$

O próximo passo do algoritmo é calcular as bases para os subespaços dos autovetores admissíveis, como visto na linha 5 do algoritmo, estas são dadas a seguir:

$$P_{v_1} = \begin{bmatrix} 3,2449 & 4,0739 \\ -4,0739 & -3,2449 \\ 2,1786 & 2,7353 \\ 2,7353 & 2,1786 \end{bmatrix}, \quad P_{v_2} = \begin{bmatrix} -1,6925 & -1,0136 \\ 1,0136 & 1,6925 \\ -0,9339 & -0,5593 \\ -0,5593 & -0,9339 \end{bmatrix}, \quad (4.11)$$

$$P_{v_3} = \begin{bmatrix} -32,6499 & -29,6446 \\ 29,6446 & 32,6499 \\ -5,1701 & -4,6942 \\ -4,6942 & -5,1701 \end{bmatrix} \quad \text{e} \quad P_{v_4} = \begin{bmatrix} -4,3618 & -2,4433 \\ 2,4433 & 4,3618 \\ -0,8163 & -0,4573 \\ -0,4573 & -0,8163 \end{bmatrix}. \quad (4.12)$$

Na sequência, são calculadas as representações dos autovetores admissíveis nas suas respectivas bases. E então, forma-se a matriz  $W_{LS}$  que é dada por

$$W_{LS} = \begin{bmatrix} 0,7036 & -0,1831 & -0,2271 & 0,0043 \\ -0,7111 & -0,2739 & 0,2309 & -0,2003 \end{bmatrix}. \quad (4.13)$$

E, finalmente, obtém-se a matriz de ganhos de realimentação de estados que aloca a autoestrutura desejada, dada por

$$K_{LS} = \begin{bmatrix} 0,0379 & 0,0501 & 1,1371 & -0,4894 \\ 0,0384 & -0,0560 & -0,6358 & 1,1757 \end{bmatrix}. \quad (4.14)$$

A seguir são apresentados os resultados computacionais obtidos através das redes neuronais estruturadas dedicadas em alocar a autoestrutura especificada para um sistema elétrico de 4ª ordem.

### 4.1.3 Alocação Completa da Autoestrutura Desejada do Circuito Elétrico RLC

Com o intuito de investigar e verificar o modelo de rede neural proposto para a solução do problema de alocação de autoestrutura, foram realizadas simulações computacionais via fórmula de atualização dos pesos. Para comprovação dos métodos anteriormente demonstrados matematicamente é aplicado em uma planta de um circuito elétrico RLC.

Em conformidade com o algoritmo *EA-RNA-Core*, inicializa-se todos os parâmetros necessários para a obtenção de uma matriz  $K$  que em malha fechada com a planta aloque a autoestrutura desejada  $\{\Lambda_d, V_d\}$ . Esta *AE* é apresentada nas Eqs. (4.9) e (4.10).

As etapas necessárias para obtenção da matriz de realimentação são expostas na Seção 3.3.1. A seguir são apresentados os resultados obtidos por meio de simulações computacionais das redes neuronais artificiais. As *RNAs* que calculam as matrizes  $W_{rna}$  e  $K_{rna}$  são denotadas pelas terminologias *RNA- $E_W$*  e *RNA- $E_K$* , respectivamente.

#### 4.1.3.1 Simulação com $\mu = 0,6$ para *RNA- $E_W$* e *RNA- $E_K$*

Apresenta-se na Fig. 11 a evolução do processo iterativo do peso sináptico  $W_{rna}$  e a sua convergência. Observa-se que com apenas 6 iterações houve a convergência do peso sináptico.

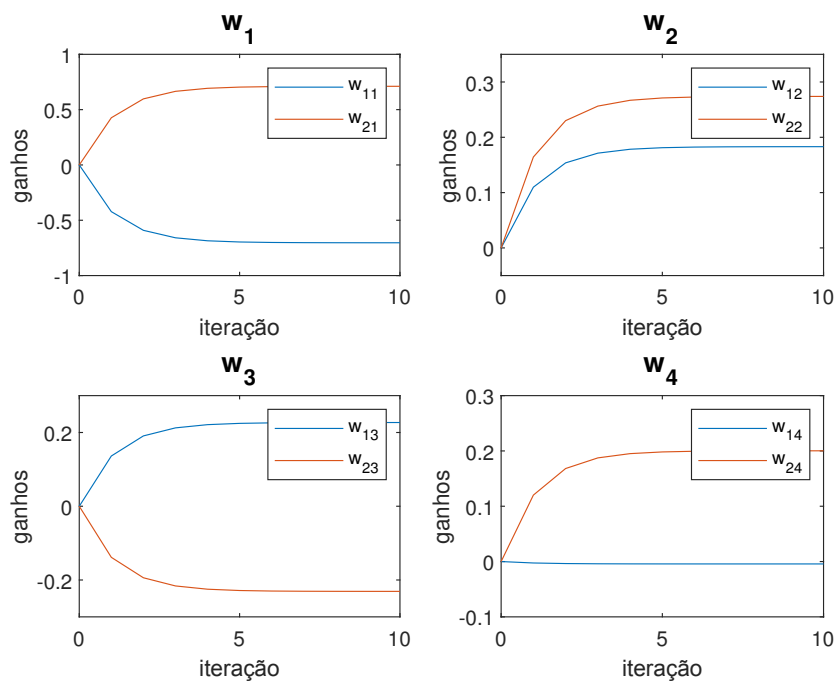


Figura 11 – Evolução do processo iterativo das colunas da matriz  $W_{rna}$  com taxa de aprendizado  $\mu = 0,6$ . Algoritmo  $EA-RNA-Core-RNA-E_W$ .

Apresenta-se a evolução do processo iterativo do ganho  $K_{rna}$  e a sua convergência na Fig. 12, ou seja, os ganhos para a primeira e segunda entradas de controle. A taxa de aprendizado  $\mu$  utilizada foi de 0,6.

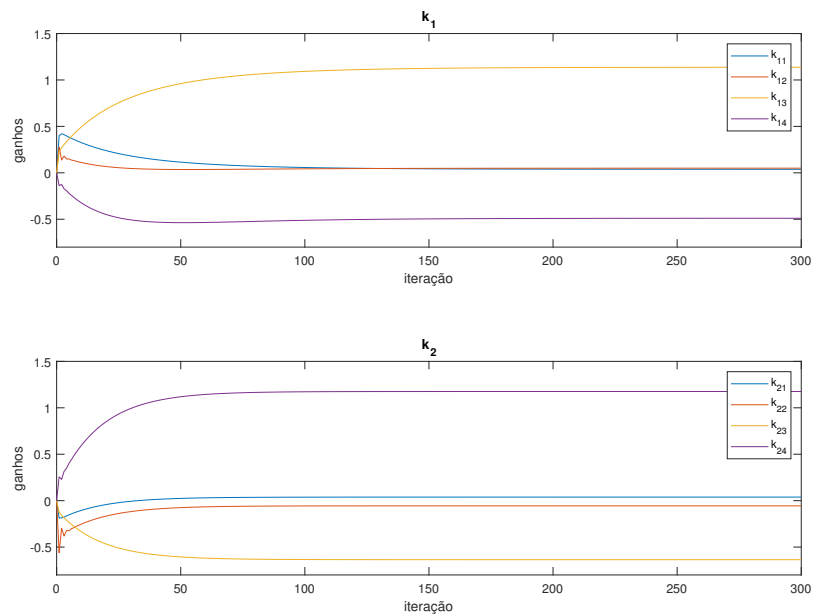


Figura 12 – Evolução do processo iterativo das colunas da matriz  $K_{rna}$  com taxa de aprendizado  $\mu = 0,6$ . Algoritmo  $EA-RNA-Core-RNA-E_K$ .

4.1.3.2 Simulação com  $\mu = 0,7$  para  $RNA-E_W$  e  $RNA-E_K$ 

Nas Fig. 13 e 14 , apresentam-se as evoluções dos processos iterativos das redes neuronais estruturadas, com ciclo de 10 iterações para a  $RNA-E_W$ , ciclo de 100 iterações para a  $RNA-E_K$ , e taxa de aprendizado  $\mu = 0,7$  para ambas as redes. As curvas da Fig. 13 representam o comportamento dos elementos  $w_{11}$ ,  $w_{12}$ ,  $w_{13}$ ,  $w_{14}$ ,  $w_{21}$ ,  $w_{22}$ ,  $w_{23}$  e  $w_{24}$  associados a matriz  $W_{rna}$ . As curvas da Fig.14 mostram a evolução dos elementos  $k_{11}$ ,  $k_{12}$ ,  $k_{13}$ ,  $k_{14}$ ,  $k_{21}$ ,  $k_{22}$ ,  $k_{23}$  e  $k_{24}$  associados a matriz  $K_{rna}$ .

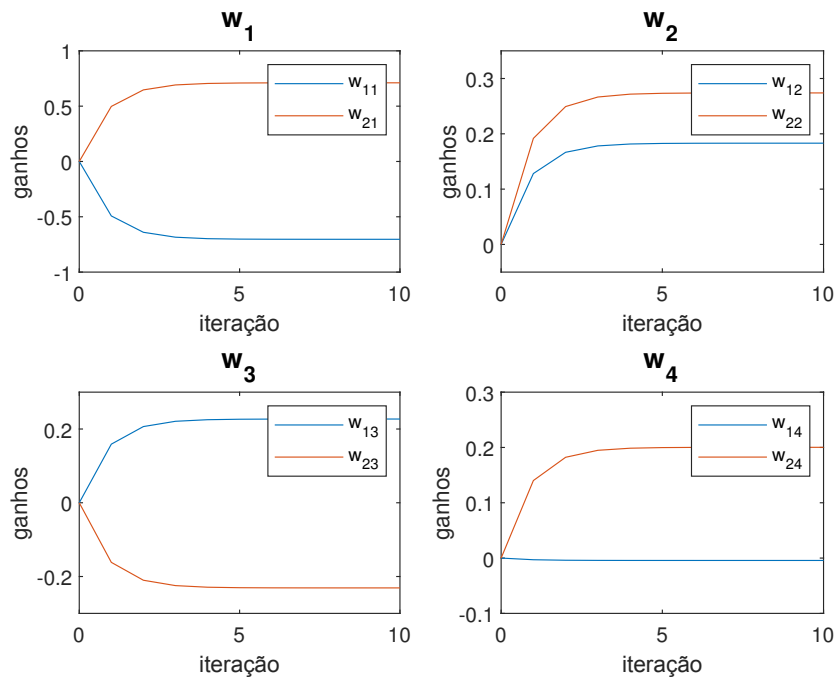


Figura 13 – Evolução do processo iterativo das colunas da matriz  $W_{rna}$  com taxa de aprendizado  $\mu = 0,7$ . Algoritmo  $EA-RNA-Core-RNA-E_W$ .

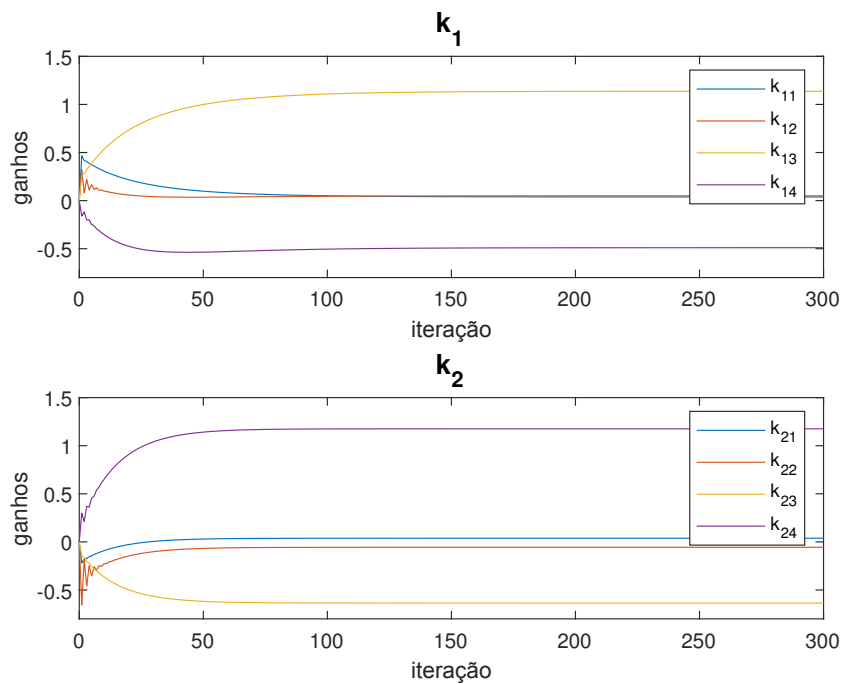


Figura 14 – Evolução do processo iterativo das colunas da matriz  $K_{rna}$  com taxa de aprendizado  $\mu = 0,7$ . Algoritmo  $EA-RNA-Core-RNA-E_K$ .

#### 4.1.3.3 Simulação com $\mu = 0,8$ para $RNA-E_W$ e $RNA-E_K$

Nas Fig. 15 e 16, apresentam-se as evoluções dos processos iterativos das redes neuronais estruturadas, com ciclo de 10 iterações para a  $RNA-E_W$ , ciclo de 100 iterações para a  $RNA-E_K$ , e taxa de aprendizado  $\mu = 0,8$  para ambas as redes. As curvas da Fig. 15 representam o comportamento dos elementos  $w_{11}$ ,  $w_{12}$ ,  $w_{13}$ ,  $w_{14}$ ,  $w_{21}$ ,  $w_{22}$ ,  $w_{23}$  e  $w_{24}$  associados a matriz  $W_{rna}$ . As curvas da Fig.16 mostram a evolução dos elementos  $k_{11}$ ,  $k_{12}$ ,  $k_{13}$ ,  $k_{14}$ ,  $k_{21}$ ,  $k_{22}$ ,  $k_{23}$  e  $k_{24}$  associados a matriz  $K_{rna}$ .



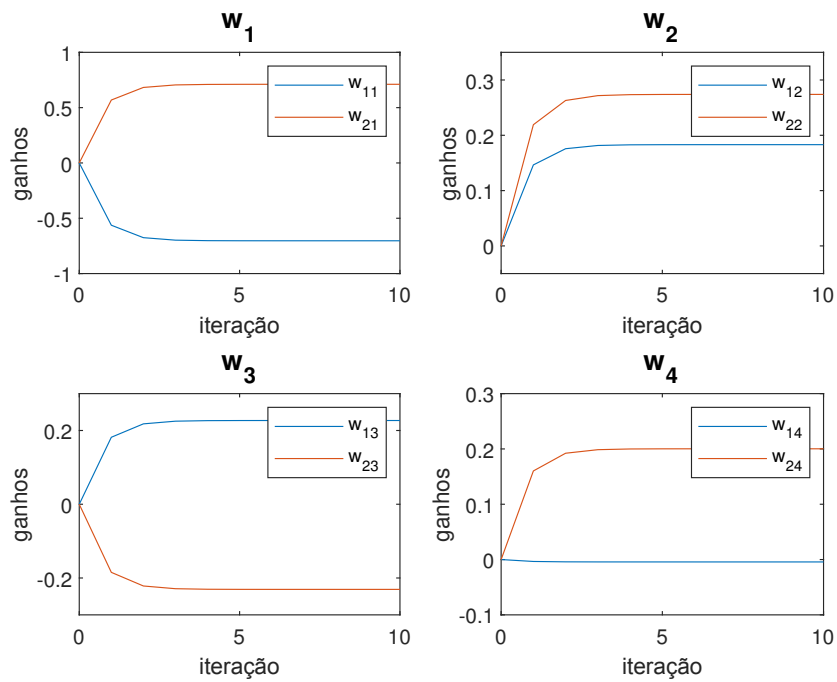


Figura 15 – Evolução do processo iterativo das colunas da matriz  $W_{rna}$  com taxa de aprendizado  $\mu = 0,8$ . Algoritmo  $EA-RNA-Core-RNA-E_W$ .

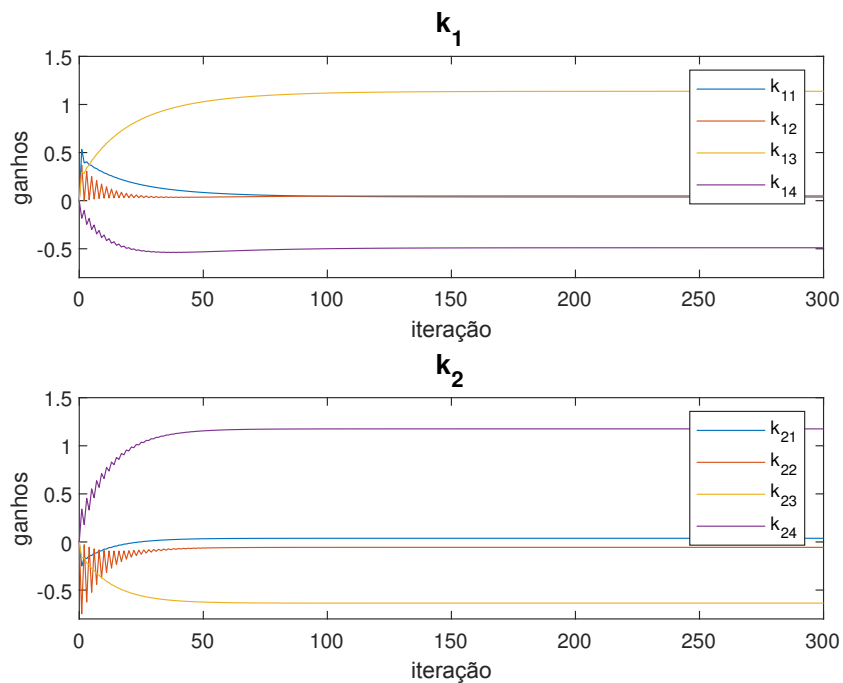


Figura 16 – Evolução do processo iterativo das colunas da matriz  $K_{rna}$  com taxa de aprendizado  $\mu = 0,8$ . Algoritmo  $EA-RNA-Core-RNA-E_K$ .

4.1.3.4 Simulação com  $\mu = 0,9$  para  $RNA-E_W$  e  $RNA-E_K$ 

Nas Fig. 17 e 18 , apresentam-se as evoluções dos processos iterativos das redes neuronais estruturadas, com ciclo de 10 iterações para a  $RNA-E_W$ , ciclo de 100 iterações para a  $RNA-E_K$ , e taxa de aprendizado  $\mu = 0,9$  para ambas as redes. As curvas da Fig. 17 representam o comportamento dos elementos  $w_{11}$ ,  $w_{12}$ ,  $w_{13}$ ,  $w_{14}$ ,  $w_{21}$ ,  $w_{22}$ ,  $w_{23}$  e  $w_{24}$  associados a matriz  $W_{rna}$ . As curvas da Fig.18 mostram a evolução dos elementos  $k_{11}$ ,  $k_{12}$ ,  $k_{13}$ ,  $k_{14}$ ,  $k_{21}$ ,  $k_{22}$ ,  $k_{23}$  e  $k_{24}$  associados a matriz  $K_{rna}$ .

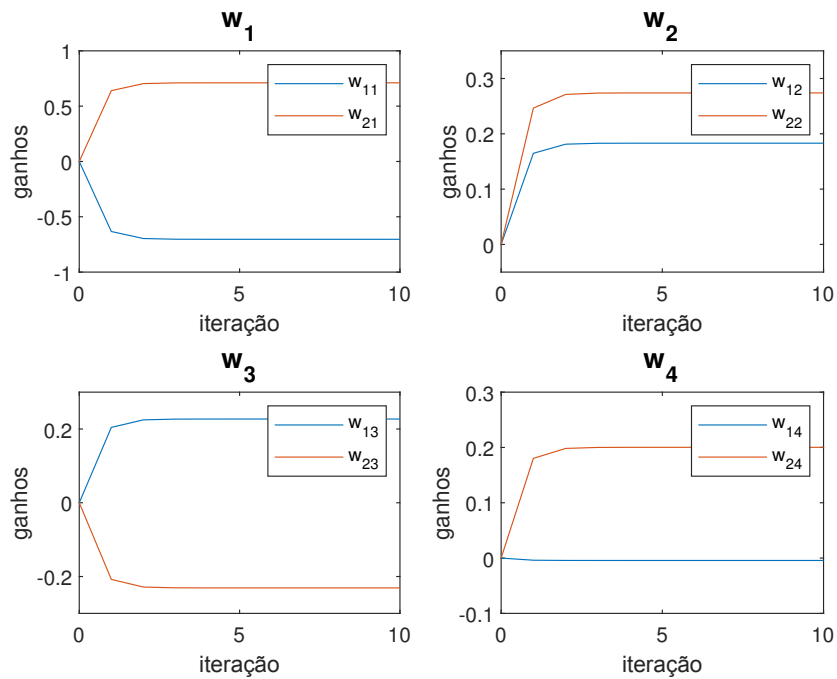


Figura 17 – Evolução do processo iterativo das colunas da matriz  $W_{rna}$  com taxa de aprendizado  $\mu = 0,9$ . Algoritmo  $EA-RNA-Core-RNA-E_W$ .

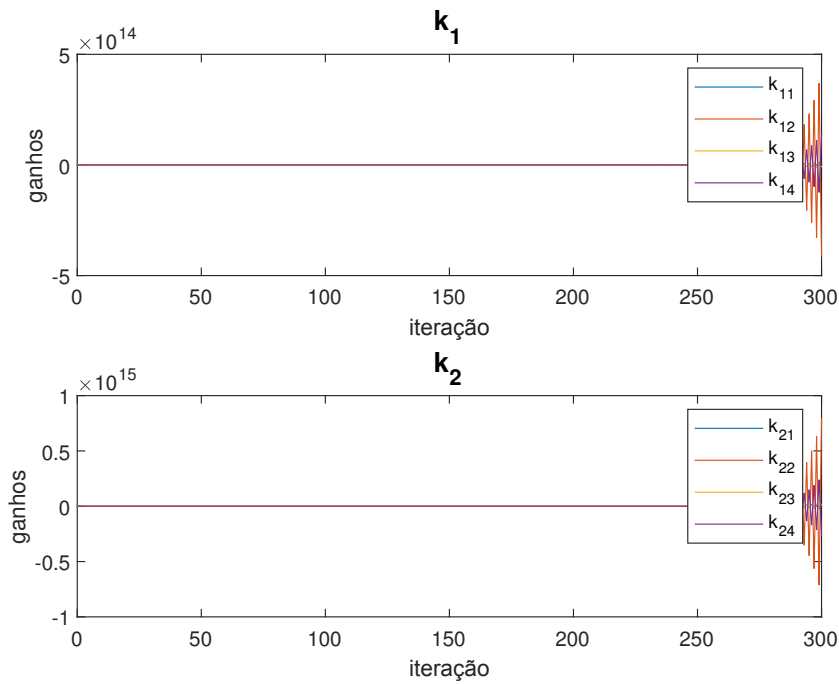


Figura 18 – Evolução do processo iterativo das colunas da matriz  $K_{rna}$  com taxa de aprendizado  $\mu = 0,9$ . Algoritmo  $EA-RNA-Core-RNA-E_K$ .

As matrizes obtidas através das redes neurais com a variação da taxa de aprendizado com os valores entre o intervalo  $[0,6-0,9]$  são dispostas nas Tabelas 1 e 2.

Tabela 1 – Matrizes  $W_{rna}$  obtidas através da mudança da taxa de aprendizado.

$\mu$	$n_{iter_w}$	$W_{rna}$			
0,6	10	- 0,7035	0,1830	0,2271	- 0,0043
		0,7110	0,2739	- 0,2309	0,2002
0,7	10	- 0,7036	0,1831	0,2271	- 0,0043
		0,7111	0,2739	- 0,2309	0,2003
0,8	10	- 0,7036	0,1831	0,2271	- 0,0043
		0,7111	0,2739	- 0,2309	0,2003
0,9	10	- 0,7036	0,1831	0,2271	- 0,0043
		0,7111	0,2739	- 0,2309	0,2003

A variação da taxa de aprendizado entre os valores definidos anteriormente não apresentaram significativa alteração na convergência dos pesos sinápticos da  $RNA-E_w$ , ou seja, todas as matrizes obtidas convergiram para valores bem próximos aos calculados através do método dos mínimos quadrados. Para valores menores que esse intervalo de  $\mu$ , por exemplo  $\mu = 0,2$ , foram observados que aumentou o número de iterações necessárias para a convergência dos parâmetros.

Tabela 2 – Matrizes  $K_{rna}$  de ganhos obtidos através da mudança da taxa de aprendizado.

$\mu$	$n_{iter_k}$	$K_{rna}$			
0,6	300	0,0380	0,0501	1,1367	- 0,4895
		0,0384	- 0,0560	- 0,6357	1,1756
0,7	300	0,0380	0,0501	1,1367	- 0,4895
		0,0384	- 0,0560	- 0,6357	1,1757
0,8	300	0,0379	0,0501	1,1367	- 0,4894
		0,0384	- 0,0560	- 0,6358	1,1757
0,9	300	$1,0e + 14 \times \begin{bmatrix} 0,0379 & 0,0501 & 1,1371 & - 0,4894 \\ 0,0384 & - 0,0560 & - 0,6358 & 1,1757 \end{bmatrix}$			

Observa-se que a RNA-E converge para os mesmos valores encontrados através do método dos mínimos quadrados. Percebe-se também que os sinais dos elementos estão invertidos devido à forma que é calculada a matriz de malha fechada do sistema, isto é,  $(A + BK)$ .

A variação na taxa de aprendizado  $\mu$  apresentou uma leve alteração na convergência de alguns pesos da rede, a saber,  $k_{11}$  e  $k_{22}$ , quando  $\mu = 0,7$ . Para  $\mu = 0,8$  observa-se uma oscilação um pouco mais acentuada na convergência dos pesos  $k_{11}$ ,  $k_{22}$  e  $k_{24}$ . Agora, para  $\mu = 0,9$  a rede neuronal não consegue obter a convergência dos pesos sinápticos.

Como pode ser observado nas Tabelas 1 e 2 as RNA-E se mostraram eficientes para a alocação completa da autoestrutura desejada, mas é necessário escolher de forma adequada a taxa de aprendizado, tanto para aumentar a precisão dos resultados, como para aumentar a velocidade de convergência.

#### 4.1.3.5 Método Iterativo de Richardson

O método iterativo de Richardson foi considerado como método de escolha da taxa de aprendizado, a ideia básica é realizar iterações até a solução da derivada de primeira ordem tornar-se zero, isto é,

$$-\frac{k(i+1) - k(i)}{\beta} = V_d^T k - w_{rna_i}, \quad (4.15)$$

sendo  $\varepsilon(i) = V_d^T k(i) - w_{rna_i}$  e  $i \geq 0$ . A expressão pode ser reorganizada para

$$k(i+1) = -k(i) - \beta(i)(V_d^T k - w_{rna_i}), \quad (4.16)$$

sendo  $\beta = \beta(i)$ ,  $k(0)$  é o vetor de condições iniciais, e a iteração ótima pode ser determinada por

$$\beta(i) = \frac{\varepsilon^T(i)\varepsilon(i)}{\varepsilon^T(i)V_d\varepsilon(i)} \quad (4.17)$$

Substituindo  $\mu$  por  $\beta$  na linha 15 do algoritmo (*EA-RNA-Core*), obtém-se a evolução do processo iterativo da matriz de realimentação de estados, apresentada na Fig. 19.

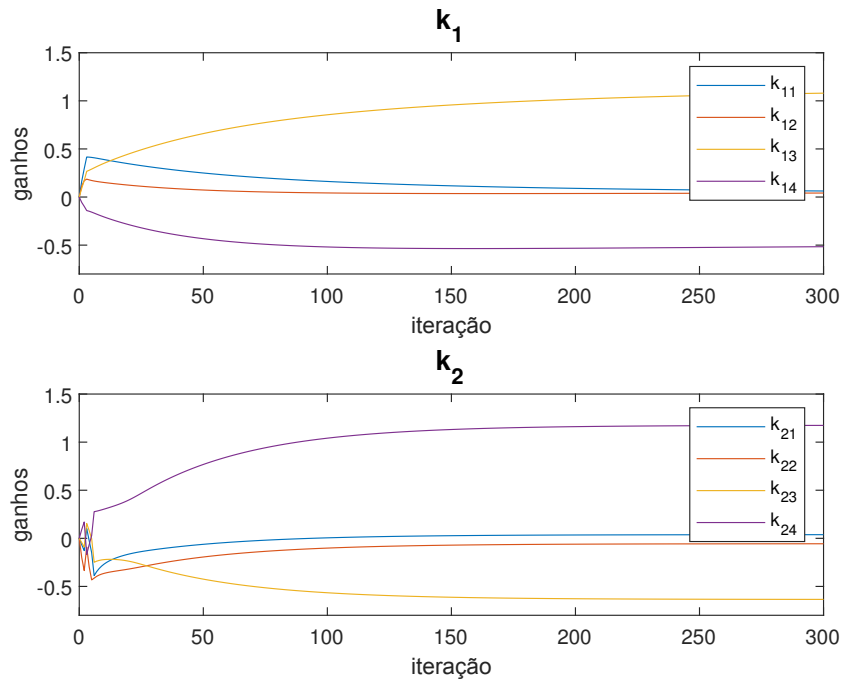


Figura 19 – Evolução do processo iterativo das colunas da matriz  $K_{rna}$  com taxa de aprendizado de  $\beta$ .

Este método de escolha da taxa de aprendizado variável  $\beta$  mostrou-se menos eficiente quando comparado com taxa de aprendizado fixa  $\mu$ . Observando-se a evolução do processo iterativo da Fig. 19, nota-se que são necessárias 150 iterações para a convergência de seus elementos. E com taxa de aprendizado fixa, a rede neuronal obtém a convergência de seus parâmetros com 50 iterações.

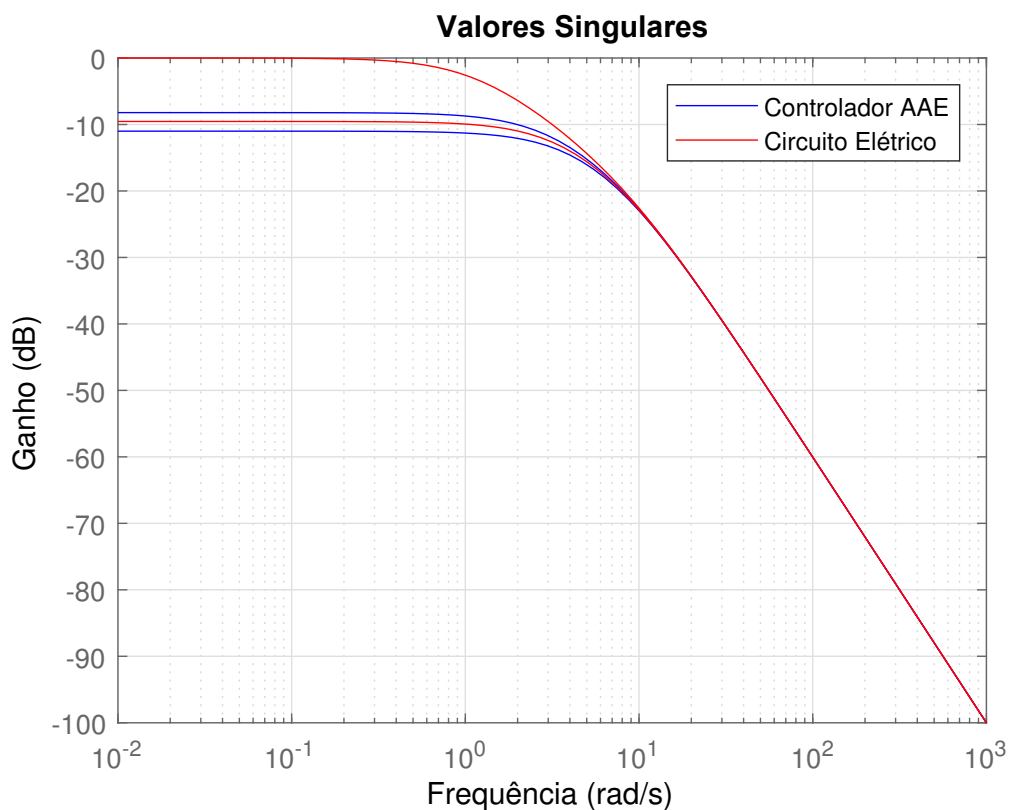
#### 4.1.4 Análise dos Valores Singulares

Conforme apresentado na Seção 2.3.3 a decomposição em valores singulares é uma ferramenta utilizada para análise e verificação de desempenho e estabilidade no domínio da frequência.

Na Tabela 3 apresentam-se os ganhos de realimentação de estados relacionados com os seus respectivos valores singulares máximos e mínimos.

Tabela 3 – Valores singulares máximos e mínimos dos sistemas em malha fechada.

Sistema	$K_{rna}$				$\sigma_{máx}$	$\sigma_{mín}$
0	0,0379	0,0501	1,1371	-0,4894	14,1865	1,9583
	0,0384	-0,0560	-0,6358	1,1757		
1	0,1438	-0,0391	2,8400	-0,6378	14,3145	3,2166
	0,0391	-0,1438	-0,6378	2,8400		
2	0,1446	-0,0342	2,8273	-0,5055	14,2873	1,7531
	0,0342	0,0095	-0,5055	0,6018		
3	-0,0034	-0,0182	0,5277	-0,2260	14,1722	1,2247
	0,0182	0,0034	-0,2260	0,5277		
4	-0,0243	-0,0138	0,3379	-0,1494	14,1682	1,0244
	0,0138	0,0243	0,1494	0,3379		

Figura 20 – Gráfico dos valores singulares do sistema em malha aberta e em malha fechada com a matriz de ganhos  $K$ .

Na figura 20 são mostradas as curvas provenientes dos valores singulares do sistema em malha aberta e em malha fechada com a matriz de ganhos  $K_{rna}$ . Observa-se que com a alocação de autoestrutura houve um aumento na faixa de estabilidade. Nota-se que houve uma melhora significativa em relação a rejeição ao ruído e conseqüentemente redução da energia de controle.

Na Fig. 22 são mostrados os valores singulares dos sistemas apresentados na Tabela 3.

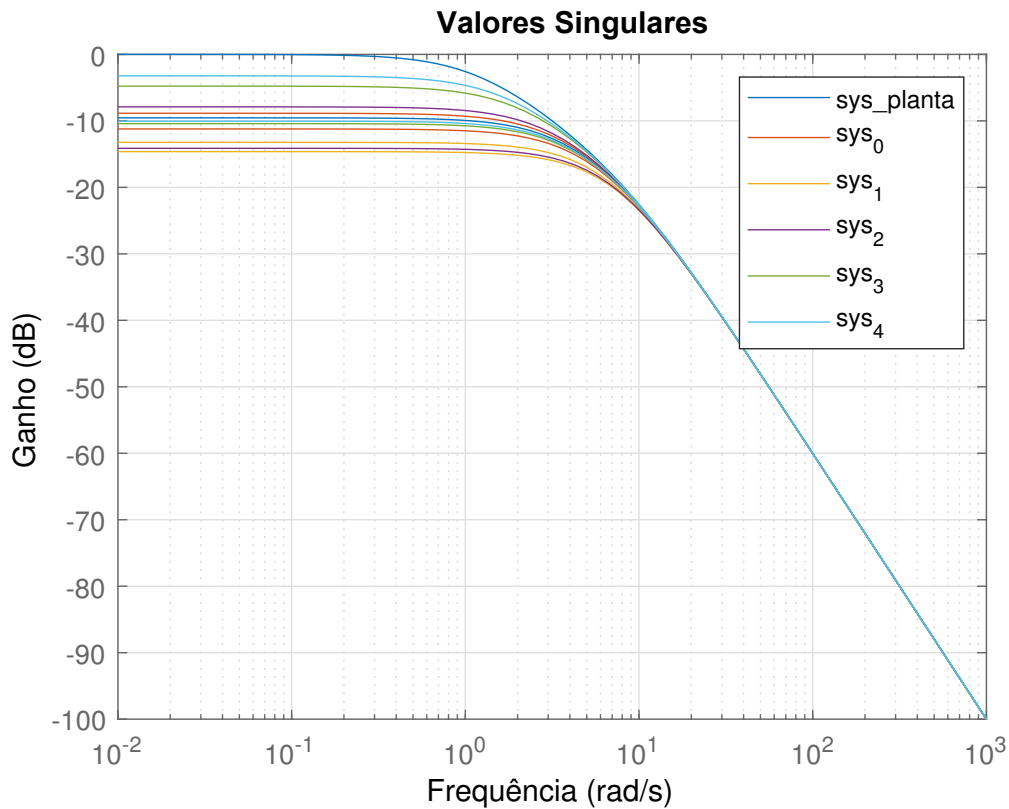


Figura 21 – SVD

Figura 22 – Gráfico Valores Singulares sistemas

#### 4.1.5 Análise Modal

Os autovalores do sistema teste são apresentados na Tabela 4 e os autovetores correspondentes estão na Tabela 5.

Tabela 4 – Autovalores do circuito elétrico.

N <sup>o</sup>	Autovalores $\lambda_i$
1	-8,4164
2	-8,1283
3	-9,9993
4	-9,9994

O comportamento dinâmico da resposta temporal do sistema pode ser melhor compreendido através da análise modal, que significa verificar os efeitos provocados pelas interações entre a autoestrutura, as condições iniciais e as matrizes de entrada e de saída

do sistema. A determinação dos coeficientes de amortecimento, das frequências naturais amortecidas e das contribuições dos componentes dos vetores modais na entrada e na saída do sistema são os parâmetros utilizados para mostrar de forma direta a influência da autoestrutura na dinâmica do sistema. Os autovalores reais e os seus autovetores associados são chamadas de modos de primeira ordem e os autovalores complexos conjugados e seus autovetores associados são chamados de modos complexos ou de segunda ordem (NETO, 2000).

Tabela 5 – Autovetores do Circuito Elétrico associados aos Autovalores 1 e 2.

Componente	Autovetores			
	$v_1$	$v_2$	$v_3$	$v_4$
1	0,9877	-0,0004	1,0000	0,0004
2	-0,0004	-0,9829	-0,0004	1,0000
3	0,1564	-0,0001	0,0001	0,0000
4	0,0001	0,1840	0,0000	-0,0001

Somente a análise dos autovalores não é suficiente para caracterizar o comportamento do sistema dinâmico. Sabe-se que cada componente do vetor de estado é formado pela contribuição de cada modo. Unindo os resultados apresentados na Tabela 4, autovalores, e na Tabela 5 autovetores, verifica-se a influência de cada modo na resposta temporal do sistema dinâmico.

Cada autovalor da Tabela 4 está associado a um autovetor da Tabela 5, e cada componente de um autovetor é associado a somente um estado. Por exemplo, o modo de 1<sup>a</sup> ordem, que corresponde ao autovalor 1 e ao autovetor 1 possui sua maior interação com o estado 1 porque seu maior elemento está na primeira posição, já para os estados 2 e 4 injeta contribuição muito pequena. Observa-se na terceira linha da Tabela 5 que o modo 4 de 1<sup>a</sup> ordem não influencia no estado 3.

Os acoplamentos entre os modos e as saídas e entre as entradas e os modos podem ser determinados pelos produtos  $CV$  e  $WB$ , respectivamente apresentados nas Eqs. (2.74) e (2.76). Então, o produto dos autovetores pela matriz de saída  $C$  fornece o grau da distribuição dos efeitos dos modos sobre a saída, Tabela 6, e o produto dos autovetores recíprocos pela matriz  $B$  mostra o grau da distribuição dos efeitos do controle sobre os modos, Tabela 11.

As colunas da Tabela 6 representam vetores e os valores numéricos de seus componentes são interpretados como o grau de acoplamento entre os modos e a saída. Por exemplo, para o modo de 1<sup>a</sup> ordem, o acoplamento representado pelo primeiro componente da coluna 1, é o que exerce maior influência na saída 1; enquanto que o segundo elemento da coluna 1 tem a menor influência para a saída 2.



Tabela 6 – Acoplamento CV do circuito elétrico - Sistema 0.

Saída	Vetores			
	1	2	3	4
1	-0,6137	0,5875	0,5692	0,4705
2	-0,5591	-0,6492	0,8072	-0,8630

Tabela 7 – Acoplamento CV do circuito elétrico - Sistema 1.

Saída	Vetores			
	1	2	3	4
1	0,6687	0,6687	0,6710	0,6710
2	-0,6687	-0,6687	0,6710	0,6710

Tabela 8 – Acoplamento CV do circuito elétrico - Sistema 2.

Saída	Vetores			
	1	2	3	4
1	0,1691	0,9263	0,9263	-0,2056
2	0,7968	-0,1966	-0,1966	-0,9690

Tabela 9 – Acoplamento CV do circuito elétrico - Sistema 3.

Saída	Vetores			
	1	2	3	4
1	-0,5514	-0,6061	-0,6972	-0,7019
2	-0,5514	0,6061	0,6972	-0,7019

Tabela 10 – Acoplamento CV do circuito elétrico - Sistema 4.

Saída	Vetores			
	1	2	3	4
1	-0,5424	-0,6004	-0,6983	-0,7025
2	-0,5424	0,6004	0,6983	-0,7025

A contribuição das entradas para os estados é observada através dos componentes dos vetores coluna da Tabela 11. Como pode ser observado as todas as entradas tem influencia direta nos modos do sistema dinâmico.

Observa-se que a entrada 4 pode exercer uma grande influência no estado 4 porque o último componente do vetor 1 possui um valor um pouco mais elevado quando comparado com os valores dos outros componentes. Contudo a entrada 2 exerce maior influência sobre o estado 2. Observando os componentes restantes do vetor de distribuição de entradas  $WB$ , verifica-se que as entradas exercem maior influência nos estados 2 e 4; para os outros estados a influência é menor.

Tabela 11 – Acoplamento WB do circuito elétrico - Sistema 0.

Estados	Vetores	
	1	2
1	-1.7029	1.6048
2	2.1486	2.2975
3	-2.0469	0.9290
4	-2.4276	-1.8991

Tabela 12 – Acoplamento WB do circuito elétrico - Sistema 1.

Estados	Vetores	
	1	2
1	- 2,0200i	- 2,0200i
2	+ 2,0200i	+ 2,0200i
3	- 5,7983i	+ 5,7983i
4	+ 5,7983i	- 5,7983i

Tabela 13 – Acoplamento WB do circuito elétrico - Sistema 2.

Estados	Vetores	
	1	2
1	0,4438	- 2,0915
2	- 3,3949i	- 0,7204i
3	+ 3,3949i	+ 0,7204i
4	0,3649	- 1,7197

Tabela 14 – Acoplamento WB do circuito elétrico - Sistema 3.

Estados	Vetores	
	1	2
1	-1,3318	1,3318
2	-1,9094	-1,9094
3	1,6598	1,6598
4	1,0462	-1,0462

Tabela 15 – Acoplamento WB do circuito elétrico - Sistema 4.

Estados	Autovetores	
	1	2
1	-1,2781	1,2781
2	-1,7985	-1,7985
3	1,5464	1,5464
4	0,9869	-0,9869

#### 4.1.6 Análise da Sensibilidade Individual dos Autovalores

Além das alocações de autoestrutura apresentada na Seção 4.1.3, foram realizadas diversas outras alocações para obtenção de uma matriz  $K$  que aloque a autoestrutura desejada de forma a garantir certas condições de desempenho, a saber, baixa sensibilidade de autovalores.

A Tabela 16 relaciona os autovalores de malha fechada e suas respectivas sensibilidades para os 5 sistemas propostos, sendo o sistema apresentado na Seção 4.1.3 é o sistema 1.

Tabela 16 – Sensibilidades individuais dos autovalores.

Sistemas	Autovalores e Sensibilidades				
0	$\Lambda_{sis_0}$	-3,2859	-4,4819	-8,4165	-8,1285
	$\eta_{sis_0}$	35,6652	5,1403	4,7098	2,4169
1	$\Lambda_{sis_1}$	-7,1011+1,8509i	-7,1011-1,8509i	-6,7389+0,6426i	-6,7389-0,6426i
	$\eta_{sis_1}$	4,1047	4,1047	3,3573	3,3573
2	$\Lambda_{sis_2}$	-2,8774	-6,9661+1,5214i	-6,9661-1,5214i	-8,6195
	$\eta_{sis_2}$	1,7822	3,6735	2,9611	4,0944
3	$\Lambda_{sis_3}$	-1,9725	-3,9906	-8,3112	-8,7812
	$\eta_{sis_3}$	6,8483	3,8523	4,2204	17,5021
4	$\Lambda_{sis_4}$	-1,6375	-3,7792	-8,4094	-8,8498
	$\eta_{sis_4}$	7,5621	4,2036	4,4497	17,8286

Observa-se na Tabela 16 que o modo de 1<sup>a</sup> ordem correspondente ao quarto autovalor do sistema 0 apresentou a menor sensibilidade dentre todos os sistemas apresentados. Conforme exposto na Seção 2.3.2, quanto mais próximo a sensibilidade do autovalor de uma unidade menor o movimento do autovalor quando sujeito à perturbações.

## 4.2 Aeronave F-16

Para evidenciar as qualidades da metodologia proposta para alocação online da autoestrutura escolheu-se para sistema teste o modelo de uma aeronave F-16. Este sistema foi apresentado por (ALBOSTAN; GÖKAŞAN, 2018) para realizar alocação de autoestrutura, consiste em fixar os autovalores em uma determinada faixa e alocar os autovetores de tal forma que ocorra o desacoplamento de modos.

A seguir apresenta-se as equações em espaço de estados do sistema multivariável F-16. A modelagem matemática está detalhadamente no Anexo A. As equações dinâmicas lateral-direcional são linearizadas entre as condições de 10kft e 200 kts e transformado em uma referência do eixo de estabilidade (STENGEL, 2015).

$$A = \begin{bmatrix} -0.383 & 4.88 & 0.172 & 0 & 1.487 & -1.53 \\ -0.994 & -0.147 & 0.0024 & 0.094 & 0.0074 & 0 \\ 1.0017 & -13.84 & -1.476 & 0 & -12.01 & 0 \\ 0 & 0 & 1.0134 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -20.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -20.2 \end{bmatrix} \quad (4.18)$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 20.2 & 0 \\ 0 & 20.2 \end{bmatrix} \quad (4.19)$$

### 4.2.1 Alocação da Autoestrutura através do Método da Projeção

Com a finalidade de validação dos resultados obtidos através das redes neurais, são calculadas por meio do método dos mínimos quadrados e projeção a matriz de representação dos autovetores admissíveis  $W_{LS}$  e a matriz de ganhos de realimentação de estados  $K_{LS}$ , conforme algoritmo AAE-LS-MP2 apresentado na seção 2.2.

As etapas necessárias para AAE por meio do método da projeção são as inicializações das matrizes dinâmicas dadas nas Eqs. (4.18) e (4.19), em seguida é especificada a

autoestrutura desejada que são dadas pelas equações

$$V_d = \begin{bmatrix} 0,0171 & -0,0360 & -0,1389 & 0,3814 - 0,0793i & 0,3814 + 0,0793i & -0,0906 \\ 0,0003 & -0,0011 & -0,0270 & 0,2626 + 0,1068i & 0,2626 - 0,1068i & -0,0261 \\ 0,1381 & 0,2911 & 0,9245 & -0,3520 - 0,0001i & -0,3520 + 0,0001i & 0,0261 \\ -0,0035 & -0,0072 & -0,1981 & 0,2085 + 0,1341i & 0,2085 - 0,1341i & -0,9942 \\ 0,4419 & 0,9560 & 0,2700 & -0,2630 - 0,1069i & -0,2630 + 0,1069i & 0,0194 \\ 0,8862 & 0,0019 & -0,1145 & 0,7084 & 0,7084 & -0,0404 \end{bmatrix}, \quad (4.20)$$

$$\Lambda_d = \begin{bmatrix} -41 & -40 & -5 & -1,2105 + 0,7776i & -1,2105 - 0,7776i & -0,0874 \end{bmatrix}. \quad (4.21)$$

O próximo passo do algoritmo é calcular as bases para os subespaços dos autovetores admissíveis, em seguida são calculadas as representações dos autovetores admissíveis nas suas respectivas bases, então forma-se a matriz  $W_{LS}$ , dada por

$$W_{LS} = \begin{bmatrix} 0,4292 & 0,9840 & -0,2067 & 0,2432 + 0,1106i & 0,2432 - 0,1106i & -0,0193 \\ 0,8610 & 0,0020 & 0,0876 & -0,6659 - 0,0272i & -0,6659 + 0,0272i & 0,0402 \end{bmatrix} \quad (4.22)$$

E, finalmente, obtém-se a matriz de ganhos de realimentação de estados que aloca a autoestrutura desejada, dada por

$$K_{LS} = \begin{bmatrix} -0,3783 & 1,9315 & -0,5711 & 0,0120 & 1,1913 & -0,0140 \\ -2,3983 & -2,3226 & -0,1468 & 0,1910 & -0,0469 & 1,0657 \end{bmatrix} \quad (4.23)$$

A seguir são apresentados os resultados computacionais obtidos através das redes neurais estruturadas dedicadas em alocar a autoestrutura especificada para um sistema de uma aeronave F-16 de 6ª ordem.

#### 4.2.2 Alocação completa da Autoestrutura Desejada do Sistema F-16

Com o intuito de investigar e verificar o modelo de rede neuronal proposta para a solução do problema de alocação de autoestrutura, foram realizadas simulações computacionais via fórmula de atualização dos pesos. Para comprovação dos métodos anteriormente demonstrados matematicamente, é aplicado em uma planta de uma aeronave F-16, no qual pretende-se controlar o movimento lateral por meio da alocação da autoestrutura.

Em conformidade com o algoritmo *EA-RNA-Core*, inicializa-se todos os parâmetros necessários para a obtenção de uma matriz  $K$ , que em malha fechada com a planta, aloque a autoestrutura desejada  $\{\Lambda_d, V_d\}$ . Esta *AE* é apresentada nas Eqs. (4.20) e (4.21).

As etapas necessárias para obtenção da matriz de realimentação são expostas na Seção 3.3.1. A seguir são retratados os resultados obtidos por meio de simulações computacionais das redes neuronais artificiais. As RNAs que obtêm as matrizes  $W_{rna}$  e  $K_{rna}$  são denotadas pelas terminologias  $RNA-E_W$  e  $RNA-E_K$ , respectivamente.

#### 4.2.2.1 Simulação com $\mu = 0,001$ para $RNA-E_W$ e $\mu = 0,4$ para $RNA-E_K$

Na Fig. 23 apresenta-se a evolução do processo iterativo do peso sináptico  $W_{rna}$ . Observa-se que com apenas 8 iterações houve a convergência do pesos sinápticos.

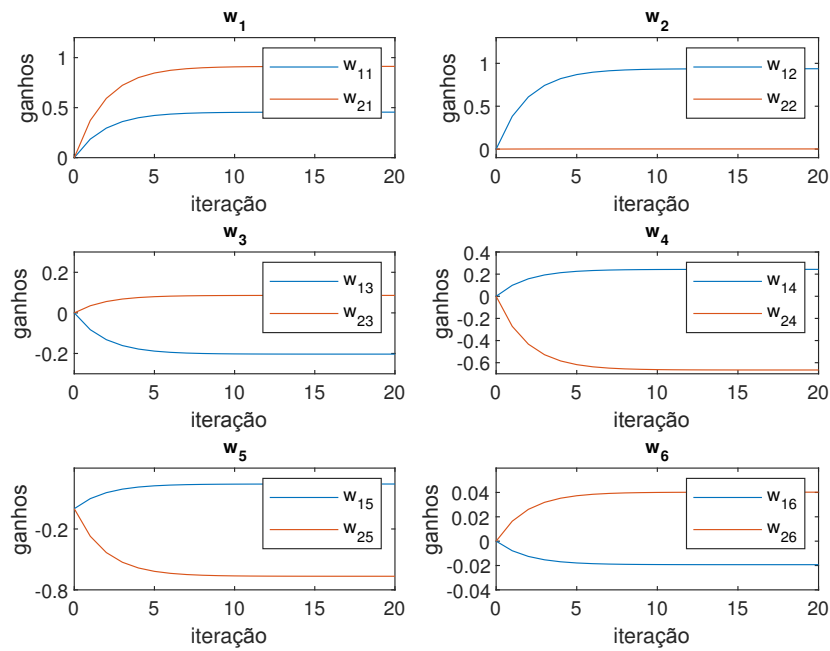


Figura 23 – Evolução do processo iterativo das colunas da matriz  $W_{rna}$  com taxa de aprendizado  $\mu = 0,001$ . Algoritmo  $EA-RNA-Core-RNA-E_W$ .

Apresenta-se o comportamento de convergência do ganho  $K_{rna}$  na Fig. 24, ou seja, os ganhos para a primeira e segunda entradas de controle. A taxa de aprendizado  $\mu$  utilizada foi de 0,4. Na Fig. 24 observa-se a convergência dos ganhos  $K_{rna}$  para as entrada de controle, e o último ganho a converge na 300<sup>a</sup> iteração.

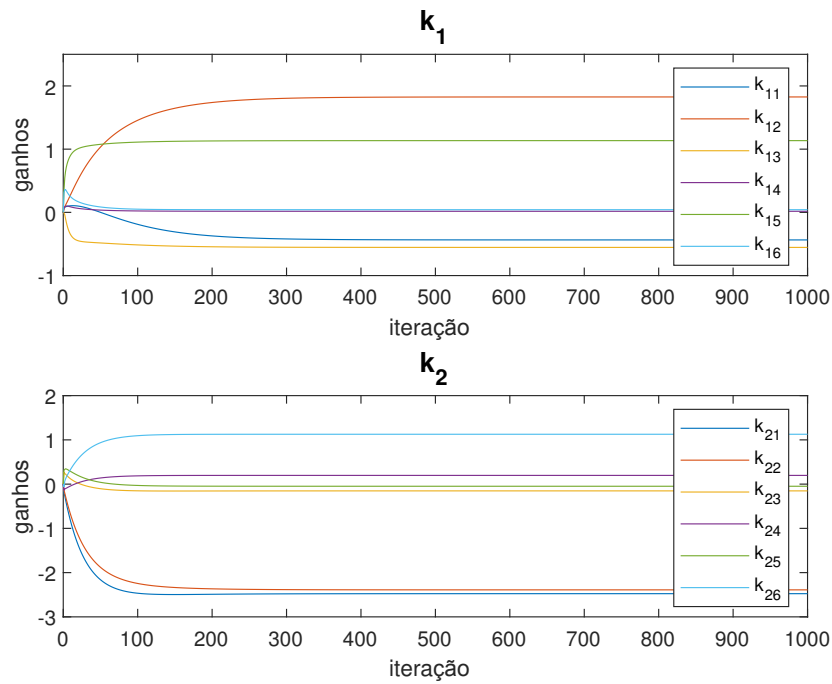


Figura 24 – Evolução do processo iterativo das colunas da matriz  $K_{rna}$  com taxa de aprendizado  $\mu = 0,4$ . Algoritmo  $EA-RNA-Core-RNA-E_K$ .

#### 4.2.2.2 Simulação com $\mu = 0,002$ para $RNA-E_W$ e $\mu = 0,5$ para $RNA-E_K$

Nas Fig. 25 e 26, apresentam-se as evoluções dos processos iterativos das redes neurais estruturadas, com ciclo de 20 iterações para a  $RNA-E_W$  e taxa de aprendizado  $\mu = 0,002$ . Para a  $RNA-E_K$  ciclo de 1000 iterações e taxa de aprendizado  $\mu = 0,5$ . As curvas da Fig. 25 representam o comportamento dos elementos  $w_{11}$ ,  $w_{12}$ ,  $w_{13}$ ,  $w_{14}$ ,  $w_{15}$ ,  $w_{16}$ ,  $w_{21}$ ,  $w_{22}$ ,  $w_{23}$ ,  $w_{24}$ ,  $w_{25}$  e  $w_{26}$  associados a matriz  $W_{rna}$ . As curvas da Fig. 26 mostram a evolução dos elementos  $k_{11}$ ,  $k_{12}$ ,  $k_{13}$ ,  $k_{14}$ ,  $k_{15}$ ,  $k_{16}$ ,  $k_{21}$ ,  $k_{22}$ ,  $k_{23}$ ,  $k_{24}$ ,  $k_{25}$  e  $k_{26}$  associados a matriz  $K_{rna}$ .

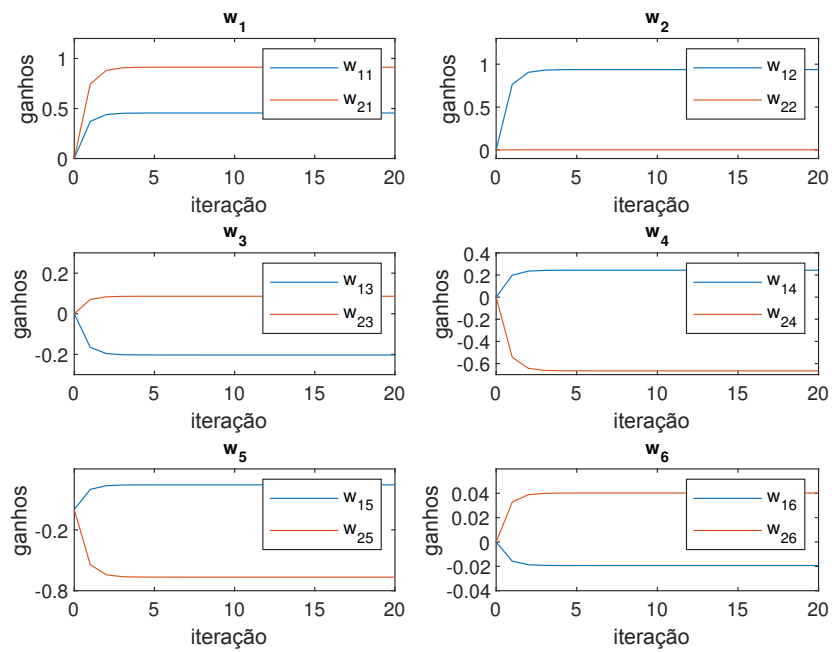


Figura 25 – Evolução do processo iterativo das colunas da matriz  $W_r$  na com taxa de aprendizado  $\mu = 0,002$ . Algoritmo  $EA-RNA-Core-RNA-E_W$ .

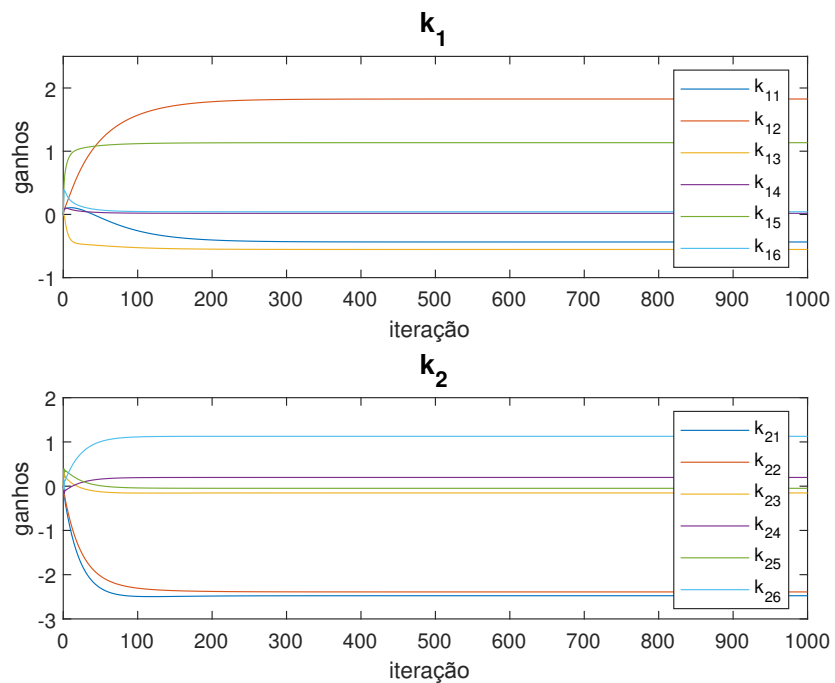


Figura 26 – Evolução do processo iterativo das colunas da matriz  $K_r$  na com taxa de aprendizado  $\mu = 0,5$ . Algoritmo  $EA-RNA-Core-RNA-E_K$ .



4.2.2.3 Simulação com  $\mu = 0,003$  para  $RNA-E_W$  e  $\mu = 0,6$  para  $RNA-E_K$

Nas Fig. 27 e 28 , apresentam-se as evoluções dos processos iterativos das redes neuronais estruturadas, com ciclo de 20 iterações para a  $RNA-E_W$  e taxa de aprendizado  $\mu = 0,003$ . Para a  $RNA-E_K$  ciclo de 1000 iterações e taxa de aprendizado  $\mu = 0,6$ . As curvas da Fig. 27 representam o comportamento dos elementos  $w_{11}, w_{12}, w_{13}, w_{14}, w_{15}, w_{16}, w_{21}, w_{22}, w_{23}, w_{24}, w_{25}$  e  $w_{26}$  associados a matriz  $W_{rna}$ . As curvas da Fig.28 mostram a evolução dos elementos  $k_{11}, k_{12}, k_{13}, k_{14}, k_{15}, k_{16}, k_{21}, k_{22}, k_{23}, k_{24}, k_{25}$  e  $k_{26}$  associados a matriz  $K_{rna}$ .

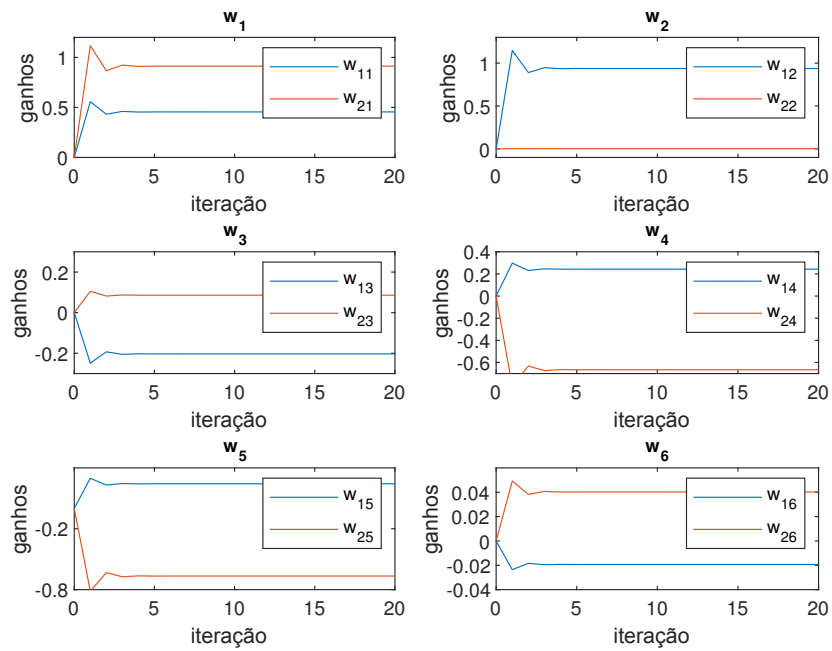


Figura 27 – Evolução do processo iterativo das colunas da matriz  $W_{rna}$  com taxa de aprendizado  $\mu = 0,003$ . Algoritmo  $EA-RNA-Core-RNA-E_W$ .

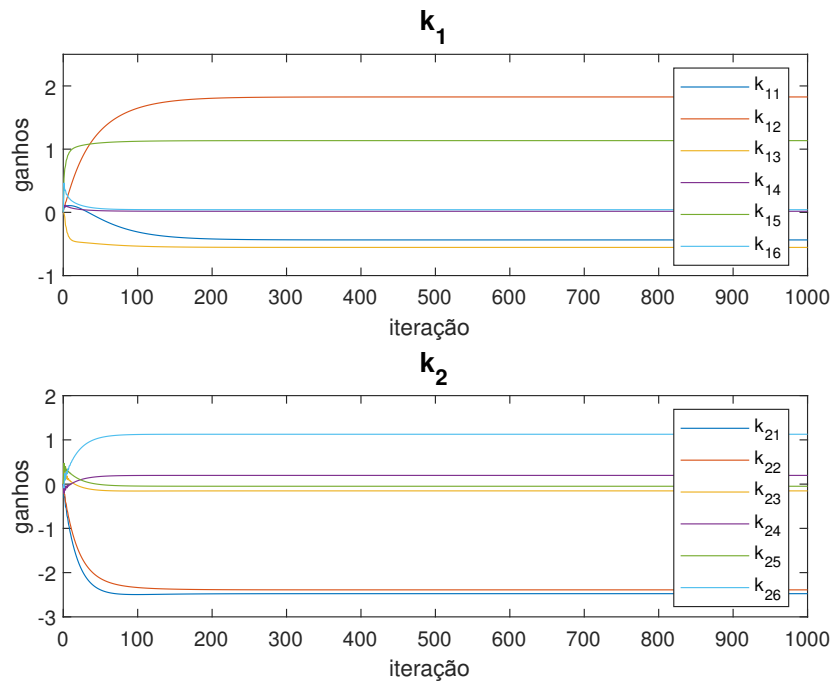


Figura 28 – Evolução do processo iterativo das colunas da matriz  $K_{rna}$  com taxa de aprendizado  $\mu = 0,6$ . Algoritmo *EA-RNA-Core-RNA-E<sub>K</sub>*.

#### 4.2.2.4 Simulação com $\mu = 0,005$ para *RNA-E<sub>W</sub>* e $\mu = 0,7$ para *RNA-E<sub>K</sub>*

Nas Fig. 29 e 30, apresentam-se as evoluções dos processos iterativos das redes neurais estruturadas, com ciclo de 100 iterações para a *RNA-E<sub>W</sub>* e taxa de aprendizado  $\mu = 0,005$ . Para a *RNA-E<sub>K</sub>* ciclo de 1000 iterações e taxa de aprendizado  $\mu = 0,7$ . As curvas da Fig. 29 representam o comportamento dos elementos  $w_{11}, w_{12}, w_{13}, w_{14}, w_{15}, w_{16}, w_{21}, w_{22}, w_{23}, w_{24}, w_{25}$  e  $w_{26}$  associados a matriz  $W_{rna}$ . As curvas da Fig.30 mostram a evolução dos elementos  $k_{11}, k_{12}, k_{13}, k_{14}, k_{15}, k_{16}, k_{21}, k_{22}, k_{23}, k_{24}, k_{25}$  e  $k_{26}$  associados a matriz  $K_{rna}$ .

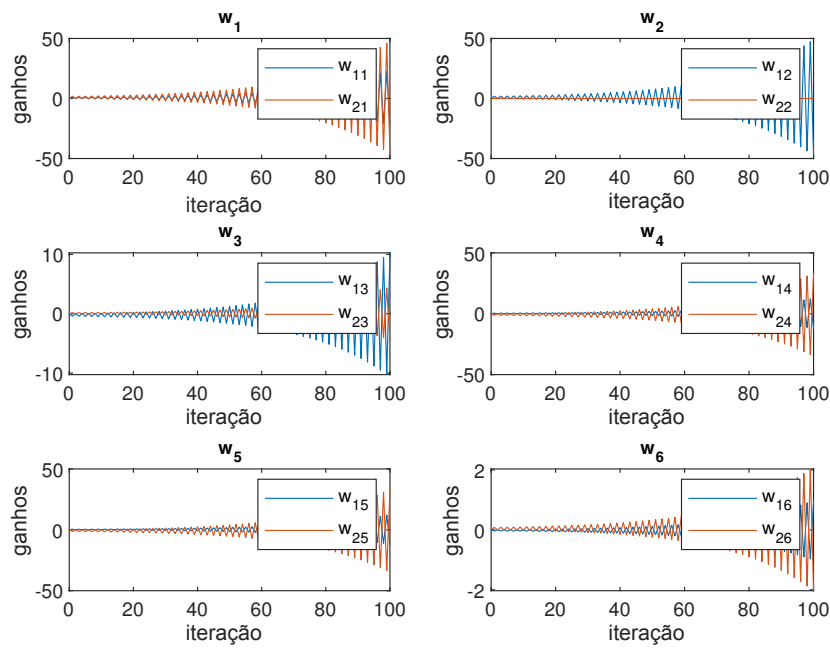


Figura 29 – Evolução do processo iterativo das colunas da matriz  $W_rna$  com taxa de aprendizado  $\mu = 0,005$ . Algoritmo  $EA-RNA-Core-RNA-EW$ .

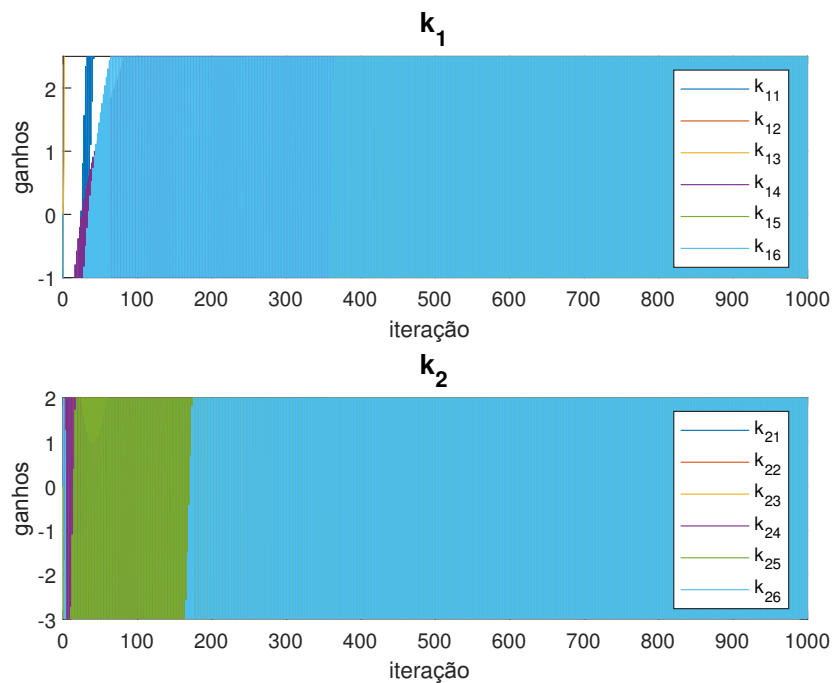


Figura 30 – Evolução do processo iterativo das colunas da matriz  $K_rna$  com taxa de aprendizado  $\mu = 0,7$ . Algoritmo  $EA-RNA-Core-RNA-EK$ .

As matrizes obtidas por meio das redes neuronais com a variação na taxa de aprendizado (intervalo de  $[0,001-0,005]$  para  $RNA-EW$  e intervalo de  $[0,4-0,7]$  para  $RNA-$

$E_K$ ), estão dispostas nas Tabelas 17 e 18.

Tabela 17 – Matrizes  $W_{rna}$  obtidas através da mudança da taxa de aprendizado.

$\mu$	$n_{iter_w}$	$W_{rna}$					
0.001	20	0,4550	0,9371	-0,2032	$0,2431 + 0,1106i$	$0,2431 - 0,1106i$	-0,0193
		0,9125	0,0019	0,0862	$-0,6659 - 0,0273i$	$-0,6659 + 0,0273i$	0,0402
0.002	20	0,4550	0,9371	-0,2032	$0,2431 + 0,1106i$	$0,2431 - 0,1106i$	-0,0193
		0,9125	0,0019	0,0862	$-0,6659 - 0,0273i$	$-0,6659 + 0,0273i$	0,0402
0.003	20	0,4550	0,9371	-0,2032	$0,2431 + 0,1106i$	$0,2431 - 0,1106i$	-0,0193
		0,9125	0,0019	0,0862	$-0,6659 - 0,0273i$	$-0,6659 + 0,0273i$	0,0402
0.005	100	-22,9722	-47,3084	10,2571	$-12,2743 - 5,5846i$	$-12,2743 + 5,5846i$	0,9752
		-46,0692	-0,0940	-4,3498	$33,6207 + 1,3767i$	$33,6207 - 1,3767i$	-2,0308

Tabela 18 – Matrizes  $K_{rna}$  de ganhos obtidos através da mudança da taxa de aprendizado.

$\mu$	$n_{iter_w}$	$K_{rna}$					
0.6	1000	-0,4361	1,8258	-0,5545	0,0171	1,1348	0,0419
		-2,4758	-2,3897	-0,1526	0,1971	-0,0483	1,1269
0.7	1000	-0,4361	1,8258	-0,5545	0,0171	1,1348	0,0419
		-2,4758	-2,3897	-0,1526	0,1971	-0,0483	1,1269
0.8	1000	-0,4361	1,8258	-0,5545	0,0171	1,1348	0,0419
		-2,4758	-2,3897	-0,1526	0,1971	-0,0483	1,1269
0.9	1000	$1 \times 10^7 \begin{bmatrix} 0,2773 & 0,1728 & -0,4781 & 0,3136 & -0,3560 & 0,5346 \\ -1,1453 & -0,7137 & 1,9744 & -1,2952 & 1,4698 & -2,2077 \end{bmatrix}$					

Observa-se que para as taxas de aprendizado  $\mu$  entre [0,4-0,7] a rede obtém a matriz de ganhos esperados, ou seja, a matriz de ganhos de realimentação de estados  $K$  aloca a autoestrutura especificada. Entretanto, quando se escolhe  $\mu = 0,005$  para  $RNA-E_W$  e  $\mu = 0,7$  para  $RNA-E_K$ , a evolução do processo iterativo mostra que seus elementos não convergem com o passar do processo iterativo, isto é, a rede com estes valores de taxa de aprendizado não alocam a autoestrutura desejada.

#### 4.2.2.5 Método Iterativo de Richardson

Agora aplicando o método de iterativo de Richardson descrito na seção 4.1.3.5, substituindo  $\mu$  por  $\beta$  na linha 15 do algoritmo ( $EA-RNA-Core$ ) obtém-se a evolução do processo iterativo da matriz de realimentação de estados, apresentada na Fig. 31.

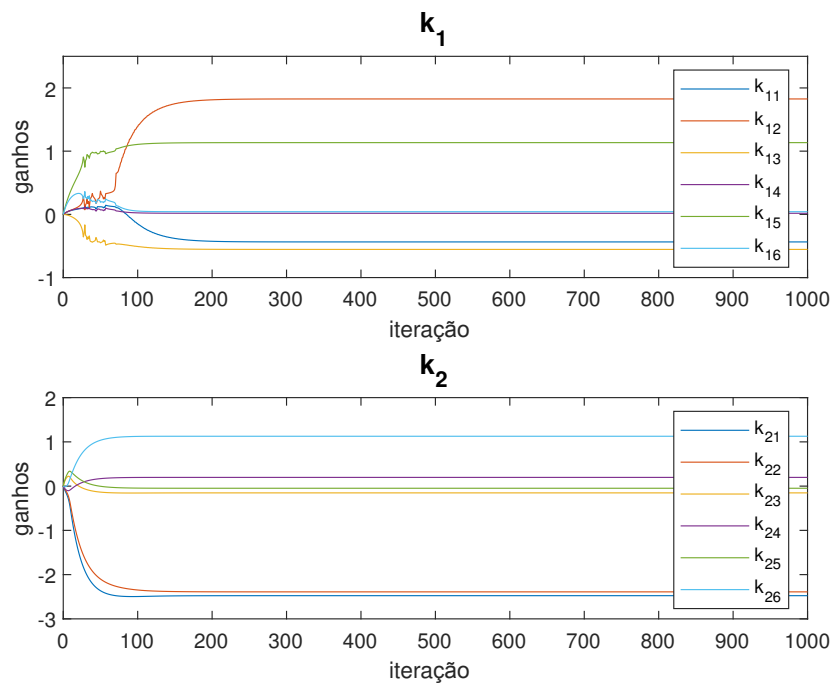


Figura 31 – Evolução do processo iterativo das colunas da matriz  $K_{rna}$  com taxa de aprendizado  $\beta$ . Algoritmo *EA-RNA-Core-RNA- $E_K$* .

Este método mostrou-se eficiente em comparação com o método de escolha da taxa de aprendizado  $\mu$ , neste método a taxa de aprendizado  $\beta$  se altera na medida que o gradiente do erro tende a zero. Observou-se que com 200 iterações os pesos sinápticos convergem.

### 4.2.3 Análise dos Valores Singulares

Como apresentado na Seção 2.3.3 a decomposição em valores singulares é uma ferramenta utilizada para análise e verificação de desempenho e estabilidade no domínio da frequência.

Os ganhos de realimentação de estados obtidos através das redes neuronais para alocação online de autoestruturas, estão relacionados com os seus respectivos valores singulares máximos e mínimos. Esta relação é apresentada na Tabela 19.

Em relação aos valores singulares máximos e mínimos apresentados na Tabela 19 observa-se que o sistema 4 apresentou o maior dos valores singulares mínimos e o sistema 3 apresentou o menor dos valores singulares máximos.

O diagrama de valores singulares fornece uma análise de desempenho e estabilidade robusta para sistemas MIMO.

Tabela 19 – Valores singulares máximos e mínimos dos sistemas em malha fechada.

Sistema	$K_{rna}$						$\sigma_{m\acute{a}x}$	$\sigma_{m\acute{i}n}$
0	-0,6674	2,0845	-0,7150	0,0418	1,1802	-0,0140	83,1757	0,0056
	-2,5204	-2,1496	-0,8980	0,2419	0,1818	1,0768		
1	-0,6649	2,0518	-0,6236	0,0437	1,1517	-0,0091	73,8134	0,0004
	-2,5056	-2,3405	-0,3640	0,2532	0,0154	1,1053		
2	-0,6649	2,0518	-0,6236	0,0437	1,1517	-0,0091	73,3304	0,0182
	-2,5056	2,3405	-0,3640	0,2532	0,0154	1,1053		
3	-0,6664	2,0719	-0,6797	0,0440	1,1692	-0,0121	72,6358	0,0001
	-2,5147	-2,2232	-0,6921	0,2547	0,1177	1,0878		
4	-0,3783	1,9315	-0,5711	0,0120	1,1913	-0,0140	75,8305	0,2191
	-2,3983	-2,3226	-0,1468	0,1910	-0,0469	1,0657		

Na Figura 32 são mostradas as curvas provenientes dos valores singulares dos sistemas em malha fechada com a matriz de ganhos  $K_{rna}$ . Observa-se que com a alocação de autoestrutura houve um aumento na faixa de estabilidade. Nota-se que houve uma melhora significativa em relação a rejeição ao ruído e conseqüentemente redução da energia de controle, também houve melhoria em relação à rejeição de perturbações.

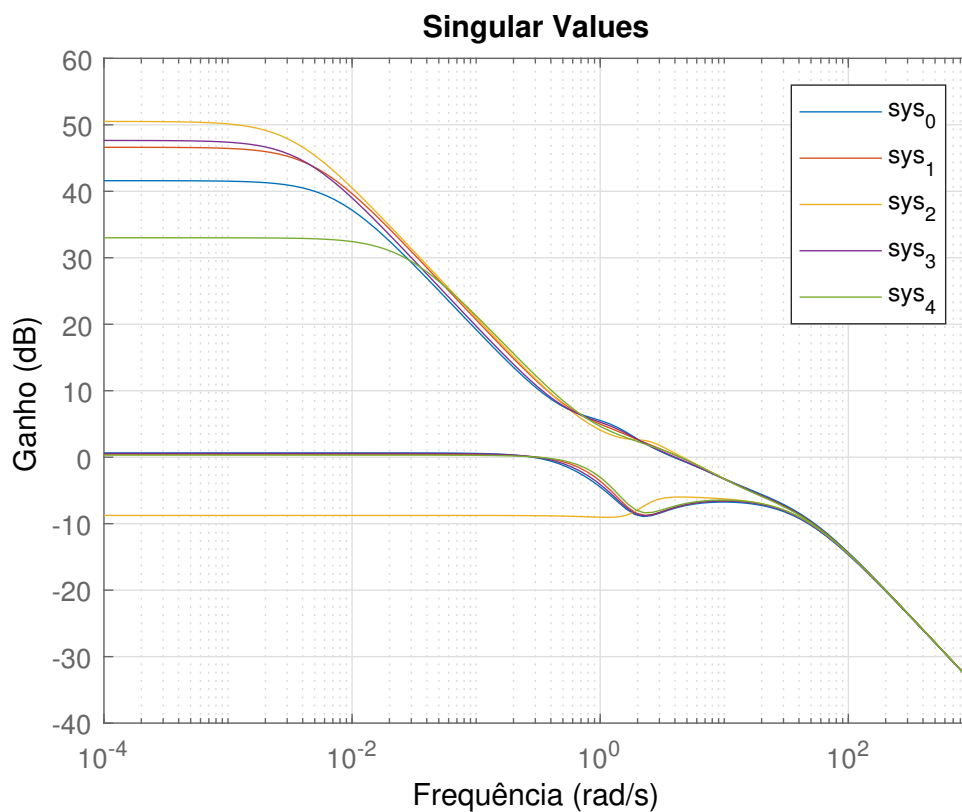


Figura 32 – Gráfico dos Valores Singulares dos Sistemas

De acordo com as formulações e teorias apresentadas na Seção 2.3.3 o sistema 4 apresentou melhor comportamento em baixas frequências, pois a curva desse sistema está mais acima que as outras, ou seja, houve uma pequena melhoria em relação a rejeição às perturbações e insensibilidade a variações da planta. Na Fig. 33 mostra os valores singulares dos sistemas apresentados na Tabela 19.

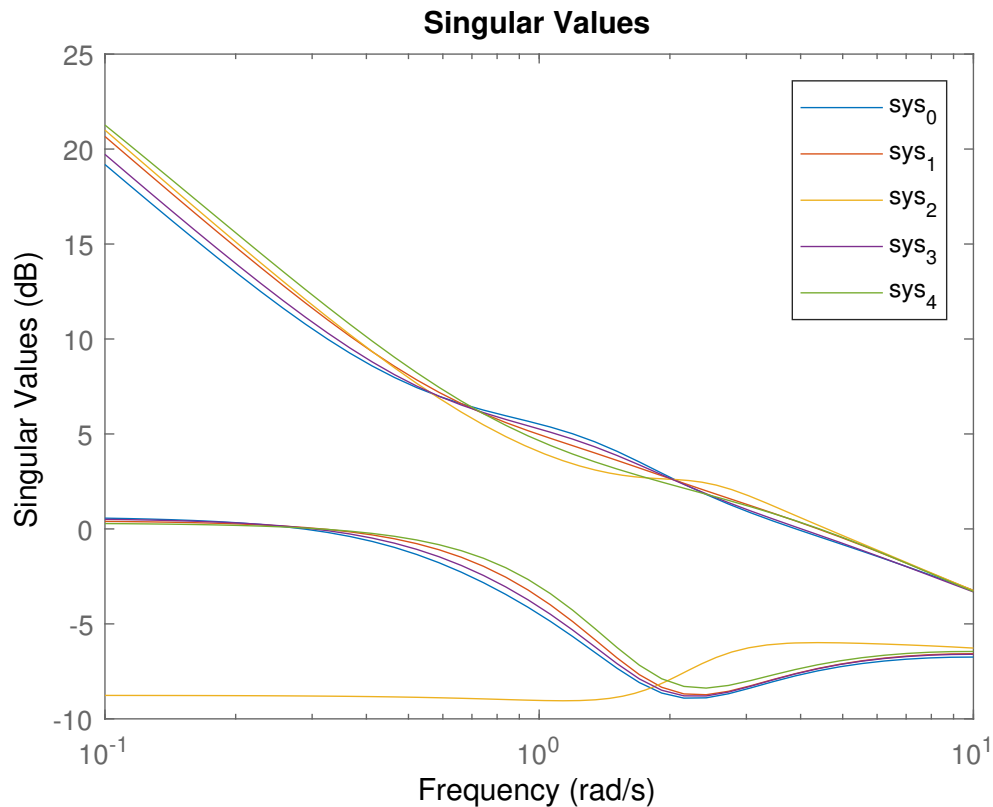


Figura 33 – Gráfico dos valores Singulares dos Sistemas

Observa-se tanto pelos dados apresentados na Tabela 19 e análise gráfica das Figuras 32 e 33 que o sistema 4 apresenta melhores resultados, pois neste houve um aumento no valor singular mínimo e sem aumento significativo do valor singular máximo, garantindo o critério de robustez. Entretanto o comportamento do sistema 2 também é aceitável, pois a este apresenta comportamento similar ao sistema 4.

#### 4.2.4 Análise Modal

Cada autovalor da Tabela 20 está associado a um autovetor da Tabela 21, e cada componente de um autovetor é associado a somente um estado. Por exemplo, o modo de 1ª ordem que corresponde ao autovalor  $\lambda_1$  e ao autovetor  $v_1$  possui sua maior interação com o estado 6 porque seu maior elemento está na sexta posição, já para os estados 1, 2 e 4 injeta menor contribuição. Observando o modo de 2ª ordem na quarta linha da Tabela

21 verifica-se que este injeta sua maior influência no estado 6, já para os estados 2, 4 e 5 injetam menor influência.

Tabela 20 – Autovalores do F-16.

Nº	Autovalores $\lambda_i$
1	-41,0437
2	-39,8983
3	-4,7029
4	-1,2105 + 0,7774i
5	-1,2105 - 0,7774i
6	-0,0266

Tabela 21 – Autovetores do F-16 associados aos Autovalores.

Componente	Autovetores					
	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	$v_6$
1	-0,0151	-0,0379	-0,1387	0,3814 - 0,0793i	0,3814 + 0,0793i	-0,0906
2	-0,0003	-0,0011	-0,0271	0,2626 + 0,1068i	0,2626 - 0,1068i	-0,0261
3	-0,1416	0,2986	0,9250	-0,3521 - 0,0001i	-0,3521 + 0,0001i	0,0261
4	0,0035	-0,0076	-0,1993	0,2087 + 0,1341i	0,2087 - 0,1341i	-0,9942
5	-0,4674	0,9533	0,2681	-0,2630 - 0,1068i	-0,2630 + 0,1068i	0,0194
6	-0,8725	-0,0225	-0,1133	0,7084	0,7084	-0,0404

Os acoplamentos entre os modos e as saídas e entre as entradas e os modos podem ser determinados pelos produtos  $CV$  e  $WB$ , respectivamente apresentados nas Eqs. (2.74) e (2.76). Então, o produto dos autovetores pela matriz de saída  $C$  fornece o grau da distribuição dos efeitos dos modos sobre a saída, Tabelas 22 - 26, e o produto dos autovetores recíprocos pela matriz  $B$  mostra o grau da distribuição dos efeitos do controle sobre os modos, Tabelas 27 - 31.

Tabela 22 – Acoplamento CV do F-16 - Sistema 0.

Saída	Vetores					
	1	2	3	4	5	6
1	-0,0291	-0,0391	0,0006	-0,4125 + 0,1483i	-0,4125 - 0,1483i	-0,0929
2	-0,0009	-0,0011	-0,0013	-0,2755 - 0,1531i	-0,2755 + 0,1531i	-0,0086
3	0,2936	0,2971	-0,8401	0,3485 - 0,0906i	0,3485 + 0,0906i	0,0074
4	- 0,0075	-0,0075	0,1387	-0,2148 - 0,1342i	-0,2148 + 0,1342i	-0,9956
5	0,9361	0,9523	-0,3246	0,2611 + 0,1621i	0,2611 - 0,1621i	0,0013
6	0,1911	-0,0573	-0,4119	-0,6482	-0,6482	-0,0027



Tabela 23 – Acoplamento CV do F-16 - Sistema 1.

Saída	Vetores					
	1	2	3	4	5	6
1	-0,0520	0,0363	0,0889	-0,3870 + 0,1024i	-0,3870 - 0,1024i	-0,0933
2	-0,0014	0,0011	0,0151	-0,2594 - 0,1205i	-0,2594 + 0,1205i	-0,0061
3	0,2322	-0,2980	-0,9309	0,3809 - 0,0876i	0,3809 + 0,0876i	0,0049
4	-0,0058	0,0076	0,1827	-0,2482 - 0,1166i	-0,2482 + 0,1166i	-0,9956
5	0,7533	-0,9537	-0,2958	0,2493 + 0,1216i	0,2493 - 0,1216i	-0,0013
6	-0,6131	-0,0181	-0,0662	-0,6731	-0,6731	0,0028

Tabela 24 – Acoplamento CV do F-16 - Sistema 2.

Saída	Vetores					
	1	2	3	4	5	6
1	0,0523	-0,0307	0,0743	0,5977	0,5977	-0,0939
2	0,0014	-0,0009	0,0121	0,1034 + 0,2292i	0,1034 - 0,2292i	-0,0022
3	-0,2033	0,2952	-0,9271	-0,3933 + 0,0965i	-0,3933 - 0,0965i	0,0033
4	0,0051	-0,0075	0,1798	0,1109 + 0,1312i	0,1109 - 0,1312i	-0,9955
5	-0,6623	0,9426	-0,2971	-0,0386 - 0,1974i	-0,0386 + 0,1974i	-0,0057
6	0,7192	0,1531	-0,1195	0,5139 - 0,2855i	0,5139 + 0,2855i	0,0111

Tabela 25 – Acoplamento CV do F-16 - Sistema 3.

Saída	Autovetores					
	1	2	3	4	5	6
1	-0,0445	0,0349	0,0289	-0,4041 + 0,1318i	-0,4041 - 0,1318i	-0,0934
2	-0,0013	0,0010	0,0033	-0,2698 - 0,1426i	-0,2698 + 0,1426i	-0,0052
3	0,2891	-0,2978	-0,8861	0,3596 - 0,0904i	0,3596 + 0,0904i	0,0039
4	-0,0073	0,0076	0,1554	-0,2262 - 0,1294i	-0,2262 + 0,1294i	-0,9956
5	0,9294	-0,9525	-0,3188	0,2566 + 0,1487i	0,2566 - 0,1487i	-0,0023
6	-0,2249	-0,0534	-0,2972	-0,6579	-0,6579	0,0049

As colunas da Tabela 26 representam vetores e os valores numéricos de seus componentes são interpretados como o grau de acoplamento entre os modos e a saída. Por exemplo, para o modo de 1<sup>a</sup> ordem, o acoplamento representado pelo primeiro componente da coluna 6, é o que exerce maior influência na saída 1; enquanto que o primeiro elemento da coluna 1 e 2 tem a menor influência para a saída 1. De forma similar é realizada a análise da influência para as outras saídas.

Tabela 26 – Acoplamento CV do F-16 - Sistema 4.

Saída	Vetores					
	1	2	3	4	5	6
1	0,0171	-0,0360	-0,1389	0,3814 - 0,0793i	0,3814 + 0,0793i	-0,0906
2	0,0003	-0,0011	-0,0270	0,2626 + 0,1068i	0,2626 - 0,1068i	-0,0261
3	0,1381	0,2911	0,9245	-0,3520 - 0,0001i	-0,3520 + 0,0001i	0,0261
4	-0,0035	-0,0072	-0,1981	0,2085 + 0,1341i	0,2085 - 0,1341i	-0,9942
5	0,4419	0,9560	0,2700	-0,2630 - 0,1069i	-0,2630 + 0,1069i	0,0194
6	0,8862	0,0019	-0,1145	0,7084	0,7084	-0,0404

A contribuição das entradas para os estados é observada através dos componentes dos vetores coluna da Tabela 27. Observa-se que a entrada 2 exerce uma grande influência no estado 1 porque o primeiro componente do vetor 1 possui um valor mais elevado quando comparado com os valores dos outros componentes. Entretanto, a influência da entrada 1 neste é bastante reduzida. Observando os componentes restantes do vetor de distribuição de entradas  $WB$ , verifica-se que as entradas exercem maior influência somente nos estados 1, 2 e 3 para os outros estados a influência relativamente pequena.

Tabela 27 – Acoplamento WB do F-16 - Sistema 0.

Estados	Vetores	
	1	2
1	14,7963	86,4598
2	9,2855	-84,6390
3	7,6767	0,9879
4	-0,6706 - 1,0697i	0,5868 + 1,0015i
5	-0,6706 + 1,0697i	0,5868 - 1,0015i
6	0,8891	0,1447

Tabela 28 – Acoplamento WB do F-16 - Sistema 1.

Estados	Vetores	
	1	2
1	0,3667	-33,8758
2	-23,0268	-27,0780
3	7,1549	1,0896
4	-0,2080 - 0,7627i	0,7335 + 1,4371i
5	-0,2080 + 0,7627i	0,7335 - 1,4371i
6	1,0610	0,1628

Tabela 29 – Acoplamento WB do F-16 - Sistema 2.

Estados	Vetores	
	1	2
1	-3,9889	25,5384
2	20,7950	18,2654
3	7,1418	0,8607
4	0,3531 + 0,2871i	-0,6974 - 0,2518i
5	0,3531 - 0,2871i	-0,6974 + 0,2518i
6	1,1160	0,0584

Tabela 30 – Acoplamento WB do F-16 - Sistema 3.

Estados	Vetores	
	1	2
1	-0,7311	-76,9206
2	-24,3500	-75,4034
3	7,3137	0,9923
4	-0,5352 - 0,9938i	0,6297 + 1,1159i
5	-0,5352 + 0,9938i	0,6297 - 1,1159i
6	0,9471	0,1497

Tabela 31 – Acoplamento WB do F-16 - Sistema 4.

Estados	Vetores	
	1	2
1	-0,7097	23,9880
2	23,3746	-10,9154
3	-7,4031	-0,7517
4	-0,1525 + 0,2972i	-0,7845 - 1,7802i
5	-0,1525 - 0,2972i	-0,7845 + 1,7802i
6	1,1642	0,2956

#### 4.2.5 Análise da Sensibilidade Individual dos Autovalores

Além das alocações de autoestrutura apresentada na Seção 4.2.2, foram realizadas diversas outras alocações para obtenção de uma matriz  $K$  que aloque a autoestrutura desejada de forma a garantir certas condições de desempenho, a saber, baixa sensibilidade de autovalores.

A Tabela 32 relaciona os autovalores de malha fechada e suas respectivas sensibilidades para os 5 sistemas propostos, sendo que o sistema apresentado na Seção 4.2.2 é o sistema 4.

Tabela 32 – Sensibilidades individuais dos autovalores.

Sistemas		Autovalores e Sensibilidades					
0	$\Lambda_{sis_0}$	39,8233	-40,0461	-6,1393	-0,9906 + 1,0461i	-0,9906 - 1,0461i	-0,0075
	$\eta_{sis_0}$	5,2990	3,8208	2,6670	2,1135	2,7667	15,2385
1	$\Lambda_{sis_1}$	-40,5736	-39,9814	-5,1642	-1,1366 + 0,8918i	-1,1366 - 0,8918i	-0,0050
	$\eta_{sis_1}$	2,5200	1,7092	1,1837	1,3294	2,9915	14,6426
2	$\Lambda_{sis_2}$	-40,7641	-39,8781	-5,2252	-1,0634 + 2,1386i	-1,0634 - 2,1386i	-0,0033
	$\eta_{sis_2}$	4,5360	1,9904	1,1898	1,8126	3,8947	17,9238
3	$\Lambda_{sis_3}$	-40,1865	-40,1865	-5,7778	-1,0392 + 0,9992i	-1,0392 - 0,9992i	-0,0040
	$\eta_{sis_3}$	4,0660	2,8793	2,0765	1,2520	2,7870	15,3069
4	$\Lambda_{sis_4}$	-39,8259	-40,9945	-4,7293	-1,2105 + 0,7776i	-1,2105 - 0,7776i	-0,0266
	$\eta_{sis_4}$	3,7938	1,6608	1,1171	1,5166	2,5539	12,3813

Observa-se na Tabela 32 que o modo de 1<sup>a</sup> ordem correspondente ao terceiro autovalor do sistema 4 apresentou a menor sensibilidade dentre todos os sistemas apresentados, e quanto menor a sensibilidade do autovalor menor é o movimento do autovalor quando sujeito à perturbações.

# Capítulo 5

## Conclusão

Neste capítulo apresenta-se a conclusão e comentários gerais da dissertação juntamente com suas principais contribuições e propostas. Ainda, cita-se sugestões para desenvolvimentos futuros relacionados à alocação online de autoestruturas.

### 5.1 Conclusões Gerais

Apresentou-se uma abordagem de controle inteligente orientada para alocação de autoestrutura em sistemas dinâmicos multivariáveis. O método está apoiado em inteligência computacional, especificamente, redes neuronais artificiais. Um algoritmo bio-inspirado, do tipo RNA para o projeto online de sistemas de controle que são dedicados a alocação de autoestrutura completa por realimentação de estados foi apresentado nesta dissertação de mestrado. Duas redes neuronais foram propostas para resolver o problema de *AAE* em sistemas de controle MIMO. A primeira rede é responsável pela obtenção da matriz de representação e a uma segunda rede calcula a matriz de realimentação de ganhos que aloca a autoestrutura especificada.

As redes neuronais estruturadas utilizadas baseiam-se na metodologia de solução de sistemas lineares. Os resultados computacionais das simulações foram comparados com o método da projeção e mínimos quadrados para alocação de autoestruturas. Percebeu-se que a escolha adequada da taxa de aprendizado influencia significativamente no comportamento das redes neuronais.

Além da análise da evolução iterativa dos parâmetros das redes neuronais foram feitas análises da resposta do sistema dinâmicos, ou seja, acoplamento entre entradas e saídas. Também foram realizadas análises relacionadas à decomposição em valores singulares dos sistemas dinâmicos.

O desenvolvimento desta proposta apresentou soluções alternativas para alocação online de autoestrutura aplicados no controle de uma planta circuito elétrico e num sistema

que representa uma aeronave, de tal forma que se acredita que a referida proposta possa contribuir para a melhoria do desempenho da estabilidade dinâmica e transitória do sistema. Os resultados apresentados são vistos como uma boa alternativa para aplicações do mundo real.

Percebeu-se também que a análise da autoestrutura é importante, pois mostra claramente a influência das relações entre os autovalores, autovetores, condições iniciais e entrada na composição da resposta temporal do sistema.

## 5.2 Contribuições

As principais contribuições desta pesquisa estão em torno dos seguintes contextos:

- Desenvolvimento de uma metodologia para Alocação completa e parcial da Autoestrutura especificada de forma online através de redes neurais artificiais que baseiam-se em variações paramétricas.

## 5.3 Trabalhos Futuros

Como trabalhos futuros, enumera-se os seguintes tópicos em Alocação de Autoestruturas que são:

- Aplicação do sistema em plantas de sistemas autônomos, e em outros processos compostos por sistemas MIMO;
- Alocação de autoestrutura utilizando projeto de controladores  $H_\infty$ ;
- Alocação online de autoestruturas baseada em Computação Evolutiva (PSO e GA);
- Alocação online de autoestruturas garantindo as condições de otimalidade, robustez e adaptabilidade dos sistemas dinâmicos;

# Referências

ALBERTOS, P.; ANTONIO, S. *Multivariable Control Systems: An Engineering Approach*. Springer London, 2006. (Advanced Textbooks in Control and Signal Processing). ISBN 9781852338435. Disponível em: <<https://books.google.com.br/books?id=mCEGCAAQBAJ>>. Citado na página 37.

ALBOSTAN, O.; GÖKAŞAN, M. Mode decoupling robust eigenstructure assignment applied to the lateral-directional dynamics of the f-16 aircraft. *Aerospace Science and Technology*, v. 77, p. 677 – 687, 2018. ISSN 1270-9638. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1270963817321302>>. Citado 4 vezes nas páginas 18, 56, 75 e 99.

ANDRY, A. N.; SHAPIRO, E. Y.; CHUNG, J. c. Eigenstructure assignment for linear systems. *IEEE Transactions on Aerospace and Electronic Systems*, AES-19, n. 5, p. 711–729, Sep. 1983. ISSN 0018-9251. Citado na página 27.

ARBIB, M. A. (Ed.). *The Handbook of Brain Theory and Neural Networks*. 2nd. ed. Cambridge, MA, USA: MIT Press, 2002. ISBN 0262011972. Citado 3 vezes nas páginas 9, 40 e 41.

BELOTTI, R.; RICHIEDEI, D. Dynamic structural modification of vibrating systems oriented to eigenstructure assignment through active control: A concurrent approach. *Journal of Sound and Vibration*, v. 422, p. 358 – 372, 2018. ISSN 0022-460X. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0022460X18301263>>. Citado na página 18.

BISHOP, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN 0387310738. Citado na página 42.

CHEN, C. *Linear System Theory and Design*. Oxford University Press, 2013. (Oxford series in electrical and computer engineering). ISBN 9780199959570. Disponível em: <<https://books.google.com.br/books?id=UTNcNAEACAAJ>>. Citado na página 51.

CICHOCKI, A.; UNBEHAUEN, R. Neural networks for solving systems of linear equations and related problems. In: . [S.l.: s.n.], 1992. v. 39, n. 2, p. 124–138. ISSN 1057-7122. Citado 2 vezes nas páginas 17 e 43.

CICHOCKI, A.; UNBEHAUEN, R. Neural networks for solving systems of linear equations. ii. minimax and least absolute value problems. In: . [S.l.: s.n.], 1992. v. 39, n. 9, p. 619–633. ISSN 1057-7130. Citado 2 vezes nas páginas 17 e 43.

CICHOCKI, A.; UNBEHAUEN, R. *Neural networks for optimization and signal processing*. J. Wiley, 1993. ISBN 9780471930105. Disponível em: <<https://books.google.com.br/books?id=Ba1QAAAAMAAJ>>. Citado 2 vezes nas páginas 43 e 44.

DESOER, C.; KUH, E. *Teoria basica de circuitos*. Guanabara, 1988. Disponível em: <<https://books.google.com.br/books?id=19XrkQEACAAJ>>. Citado 2 vezes nas páginas 56 e 57.

ENGELBRECHT, A. *Computational Intelligence: An Introduction*. Wiley, 2007. ISBN 9780470512500. Disponível em: <<https://books.google.com.br/books?id=IZosIcgJMjUC>>. Citado 3 vezes nas páginas 40, 41 e 42.

FERREIRA, C. C. T.; NETO, J. V. F. Projeto do Filtro de Kalman via Alocação de Autoestrutura e Algoritmo Genético. *VI Simpósio Brasileiro de Automação Inteligente*, Setembro 2003. Citado na página 16.

FERREIRA, C. C. T.; NETO, J. V. F.; TORRICO, F. A. Alocação de Autoestrutura via Controle LQG/LTR e Computação Evolutiva. *VI Simpósio Brasileiro de Automação Inteligente*, Setembro 2003. Citado na página 16.

GAJSKI, D. et al. *Embedded System Design: Modeling, Synthesis and Verification*. Springer US, 2009. ISBN 9781441905048. Disponível em: <<https://books.google.com.br/books?id=XYZAAAAAQBAJ>>. Citado 2 vezes nas páginas 49 e 53.

GAN, C.; DANAI, K. Fault diagnosis of the ifac benchmark problem with a model-based recurrent neural network. In: *Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No.99CH36328)*. [S.l.: s.n.], 1999. v. 2, p. 1755–1760 vol. 2. Citado na página 17.

GOLUB, G.; LOAN, C. V. *Matrix Computations*. Johns Hopkins University Press, 2013. (Johns Hopkins Studies in the Mathematical Sciences). ISBN 9781421407944. Disponível em: <<https://books.google.com.br/books?id=X5YfsuCWpxMC>>. Citado 2 vezes nas páginas 44 e 45.

HAM, F.; KOSTANIC, I. *Principles of Neurocomputing for Science and Engineering*. McGraw Hill, 2001. ISBN 9780070259669. Disponível em: <<https://books.google.com.br/books?id=Xp8ZQAQAAIAAJ>>. Citado 4 vezes nas páginas 9, 16, 46 e 47.

HANG, D. L. et al. Ub1 - a recurrent neural network based parallel machine for solving simultaneous linear equations. In: *Proceedings 4th Brazilian Symposium on Neural Networks*. [S.l.: s.n.], 1997. p. 14–18. Citado 2 vezes nas páginas 17 e 39.

HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999. (International edition). ISBN 9780139083853. Disponível em: <<https://books.google.com.br/books?id=M5abQgAACAAJ>>. Citado 2 vezes nas páginas 50 e 52.



HAYKIN, S. S. et al. *Neural networks and learning machines*. [S.l.]: Pearson Upper Saddle River, NJ, USA.; 2009. v. 3. Citado 6 vezes nas páginas 9, 39, 40, 41, 42 e 43.

HEBB, D. *The Organization of Behavior: A Neuropsychological Theory*. John Wiley, 1964. (Science editions). Disponível em: <<https://books.google.com.br/books?id=ngsPtWEACAAJ>>. Citado na página 43.

HUBER, P. *Robust Statistical Procedures: Second Edition*. Society for Industrial and Applied Mathematics, 1996. (CBMS-NSF Regional Conference Series in Applied Mathematics). ISBN 9780898713794. Disponível em: <<https://books.google.com.br/books?id=9xyux0TiJ60C>>. Citado na página 45.

KAIADI, M. *Artificial Neural Networks Modelling for Monitoring and Performance Analysis of a Heat and Power Plant*. Dissertação (Thesis for the Degree of Master of Science) — Department of Energy Sciences(LUND UNIVERSITY), 2006. Citado 2 vezes nas páginas 40 e 41.

KIMURA, H. On pole assignment by output feedback. *International Journal of Control*, Taylor Francis, v. 28, n. 1, p. 11–22, 1978. Disponível em: <<https://doi.org/10.1080/00207177808922432>>. Citado na página 25.

KONARA, A. I.; ANNAKAGE, U. D. Robust power system stabilizer design using eigenstructure assignment. *IEEE Transactions on Power Systems*, v. 31, n. 3, p. 1845–1853, May 2016. ISSN 0885-8950. Citado na página 18.

KUMAR, V. *Introduction to parallel computing: design and analysis of algorithms*. Benjamin/Cummings Pub. Co., 1994. ISBN 9780805331707. Disponível em: <<https://books.google.com.br/books?id=UpNQAAAAMAAJ>>. Citado na página 43.

LEE, E.; SESHIA, S. *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*. MIT Press, 2016. (Electrical Engineering Computer Sciences). ISBN 9780262533812. Disponível em: <<https://books.google.com.br/books?id=chPiDQAAQBAJ>>. Citado na página 53.

LI, G.-J.; WAH. The design of optimal systolic arrays. *IEEE Transactions on Computers*, C-34, n. 1, p. 66–77, Jan 1985. ISSN 0018-9340. Citado na página 43.

LIU, D.; LU, Z. A new synthesis approach for feedback neural networks based on the perceptron training algorithm. *IEEE Transactions on Neural Networks*, v. 8, n. 6, p. 1468–1482, Nov 1997. ISSN 1045-9227. Citado na página 18.

LIU, G.; PATTON, R. *Eigenstructure assignment for control system design*. J. Wiley, 1998. ISBN 9780471975496. Disponível em: <<https://books.google.com.br/books?id=P9hSAAAAMAAJ>>. Citado 4 vezes nas páginas 15, 21, 29 e 36.

LU, J.; CHIANG, H. D.; THORP, J. S. Eigenstructure assignment by decentralized feedback control. *IEEE Transactions on Automatic Control*, v. 38, n. 4, p. 587–594, April 1993. ISSN 0018-9286. Citado na página 15.

MARSLAND, S. *Machine Learning: An Algorithmic Perspective, Second Edition*. CRC Press, 2015. ISBN 9781498759786. Disponível em: <<https://books.google.com.br/books?id=y\oYCwAAQBAJ>>. Citado na página 42.

- MOOD, A.; GRAYBILL, F.; BOES, D. *Introduction to the Theory of Statistics*. McGraw-Hill, 1974. (International Student edition). ISBN 9780070428645. Disponível em: <<https://books.google.com.br/books?id=Viu2AAAAIAAJ>>. Citado na página 45.
- MOORE, B. On the flexibility offered by state feedback in multivariable systems beyond closed loop eigenvalue assignment. *IEEE Transactions on Automatic Control*, v. 21, n. 5, p. 689–692, October 1976. ISSN 0018-9286. Citado 2 vezes nas páginas 24 e 25.
- NETO, J. V. d. F. *Alocação Computacional Inteligente de Autoestruturas para Controle Multivariável*. Tese (Doutorado) — UNICAMP, Março 2000. Citado 3 vezes nas páginas 16, 31 e 71.
- PINTO, V. P. *Controladores Robustos LQG/LTR com Ação Integral Aplicados ao Controle de Geradores Eólicos Interligados ao Sistema de Distribuição de Energia Elétrica Baseados em Máquina de Indução Duplamente Alimentada*. Tese (Doutorado) — UFC, Janeiro 2012. Citado 3 vezes nas páginas 9, 36 e 38.
- SKOGESTAD, S.; POSTLETHWAITE, I. *Multivariable Feedback Control: Analysis and Design*. USA: John Wiley & Sons, Inc., 2005. ISBN 0470011688. Citado na página 37.
- SOBEL, K. M.; LALLMAN, F. J. Eigenstructure assignment for the control of highly augmented aircraft. In: *1988 American Control Conference*. [S.l.: s.n.], 1988. p. 1266–1276. Citado na página 15.
- SOBEL, K. M.; SHAPIRO, E. Y. Robustness/performance tradeoffs in eigenstructure assignment with flight control application. In: *1987 American Control Conference*. [S.l.: s.n.], 1987. p. 380–385. Citado na página 28.
- SOBEL, K. M.; SHAPIRO, E. Y.; JR, A. N. A. Eigenstructure assignment. *International Journal of Control*, Taylor Francis, v. 59, n. 1, p. 13–37, 1994. Disponível em: <<https://doi.org/10.1080/00207179408923068>>. Citado 3 vezes nas páginas 15, 25 e 27.
- STENGEL, R. *Flight Dynamics*. Princeton University Press, 2015. ISBN 9781400866816. Disponível em: <<https://books.google.com.br/books?id=-OjZBQAAQBAJ>>. Citado 2 vezes nas páginas 75 e 99.
- WANG, L.; MENDEL, J. M. Structured trainable networks for matrix algebra. In: *1990 IJCNN International Joint Conference on Neural Networks*. [S.l.: s.n.], 1990. p. 125–132 vol.2. Citado 4 vezes nas páginas 17, 39, 46 e 47.
- WEI, X.; MOTTERSHEAD, J. E. Block-decoupling vibration control using eigenstructure assignment. *Mechanical Systems and Signal Processing*, v. 74, p. 11 – 28, 2016. ISSN 0888-3270. Special Issue in Honor of Professor Simon Braun. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0888327015001569>>. Citado na página 18.
- YAICI, M.; HARICHE, K. On eigenstructure assignment using block poles placement. *European Journal of Control*, v. 20, n. 5, p. 217 – 226, 2014. ISSN 0947-3580. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0947358014000557>>. Citado na página 17.

---

ZHANG, Y. A primal-dual interior point approach for computing  $l_1$  and  $l$  solutions of overdetermined linear systems. *Journal of Optimization Theory and Applications - J OPTIMIZ THEOR APPL*, v. 77, p. 323–341, 05 1993. Citado 2 vezes nas páginas 17 e 45.

# Apêndice A

## Modelagem Matemática da Aeronave F-16

### A.1 Descrição do Modelo

A lei de controle lateral-direcional é baseada na dinâmica linearizada, dinâmica de rolagem e espiral da aeronave. As equações de estado lateral-direcionais linearizadas são derivadas em termos de pequenas perturbações no referencial do eixo do corpo da aeronave (STENGEL, 2015) e (ALBOSTAN; GÖKAŞAN, 2018).

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx\end{aligned}\tag{A.1}$$

sendo  $x = [r \ v \ p \ ]^T$ ,  $u = [\delta_a \ \delta_r]^T$  e  $C = I^{4 \times 4}$ .

Os atuadores podem ser incluídos no modelo linearizado, de seguinte forma.

$$\begin{aligned}\hat{A} &= \begin{bmatrix} A & B \\ 0^{2 \times 4} & -A^{act} \end{bmatrix} \\ \hat{B} &= \begin{bmatrix} 0^{4 \times 2} \\ A^{act} \end{bmatrix} \\ \hat{C} &= I^{6 \times 6} \\ A_{act} &= \text{diag} [ 20.2 \ 20.2 ]\end{aligned}\tag{A.2}$$

sendo  $x = [r \ v \ p \ \delta_a \ \delta_r]^T$  e  $u_c = [\delta_a^c \ \delta_r^c]^T$ .

Por uma questão de simplificação, as representações de segunda ordem dos modelos

de sensores são usadas para definir as saídas reais.

$$\dot{\hat{x}} = \tilde{A}\tilde{x} + \tilde{B}u_s \quad u_s = \hat{x} \quad \tilde{y} = \tilde{C}\tilde{x} + \tilde{D}u_s \quad (\text{A.3})$$

sendo

$$\tilde{A} = \text{blockdiag}(A_{rate}, A_{ad}, A_{rate}, A_{att}, A_{def}, A_{def}),$$

$$\tilde{B} = \text{blockdiag}\left(\begin{bmatrix} 0 & 1 \end{bmatrix}^T, \dots, \begin{bmatrix} 0 & 1 \end{bmatrix}^T\right)$$

$$\tilde{C} = \text{blockdiag}(C_{rate}, C_{ad}, C_{rate}, C_{att}, C_{def}, C_{def}),$$

$$\tilde{D} = \text{diag}(D_{rate}, D_{ad}, D_{rate}, D_{att}, D_{def}, D_{def}),$$

$$A_{rate} = \begin{bmatrix} 0 & 1 \\ -1105.3 & -45.5 \end{bmatrix}, \quad B_{rate} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$$

$$C_{att} = \begin{bmatrix} 799.74 & -19.39 \end{bmatrix}, \quad D_{att} = 0.1099$$

$$A_{ad} = \begin{bmatrix} 0 & 1 \\ -5100.2 & -143.9 \end{bmatrix}, \quad B_{ad} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$$

$$C_{ad} = \begin{bmatrix} 3924.1 & -92.5 \end{bmatrix}, \quad D_{ad} = 0.2306$$

$$A_{def} = \begin{bmatrix} 0 & 1 \\ -193440 & -620 \end{bmatrix}, \quad B_{def} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$$

$$C_{def} = \begin{bmatrix} 193440 & 0 \end{bmatrix} \quad e \quad D_{def} = 0$$

A equação total incluindo o modelo dos sensores podem ser denotados da seguinte forma

$$\begin{aligned} \dot{\bar{x}} &= \bar{A}\bar{x} + \bar{B}\bar{u}_c \\ \bar{y} &= \bar{C}\bar{x} \end{aligned} \quad (\text{A.4})$$

$$\text{sendo } \bar{x} = \begin{bmatrix} \tilde{x} & \hat{x} \end{bmatrix}^T, \quad \bar{A} = \begin{bmatrix} \tilde{A}^{12 \times 12} & \tilde{B}^{12 \times 6} \\ 0^{6 \times 12} & \hat{A}^{6 \times 6} \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} 0^{12 \times 2} \\ \hat{B}^{6 \times 2} \end{bmatrix} \quad e \quad \bar{C} = \begin{bmatrix} \tilde{C}^{6 \times 12} & \tilde{D}^{6 \times 6} \end{bmatrix}.$$

As equações derivadas até agora são definidas no referencial do eixo do corpo, a fim de representar as equações no quadro de referência do eixo de estabilidade que precisamos implementar a matriz de transformação.

$$\begin{aligned}\dot{\bar{x}}_s &= \bar{A}_s \bar{x}_s + \bar{B}_s \bar{u}_c \\ \bar{y}_s &= \bar{C}_s \bar{x}_s\end{aligned}\tag{A.5}$$

sendo

$$\bar{x}_s = \begin{bmatrix} \tilde{x} & \hat{x}_s \end{bmatrix}^T, \quad \hat{x}_s = \begin{bmatrix} r_s & \beta & p_s & \delta_a & \delta_r \end{bmatrix}^T$$

$$\bar{A}_s = \bar{T} \bar{A} (\bar{T})^{-1} = \begin{bmatrix} \tilde{A} & \tilde{B} (\hat{T})^{-1} \\ \mathbf{0}^{6 \times 12} & \hat{T} \hat{A} (\hat{T})^{-1} \end{bmatrix}$$

$$\bar{B}_s = \bar{T} \bar{B} = \begin{bmatrix} \mathbf{0}^{12 \times 2} \\ \hat{T} \hat{B} \end{bmatrix}$$

$$\bar{C}_s = \hat{T} \bar{C} (\bar{T})^{-1} = \begin{bmatrix} \hat{T} \tilde{C} & \mathbf{0} \end{bmatrix}, \quad \bar{T} = \begin{bmatrix} I^{12 \times 12} & \mathbf{0}^{12 \times 6} \\ \mathbf{0}^{6 \times 12} & \hat{T}^{6 \times 6} \end{bmatrix}$$

$$\hat{T} = \begin{bmatrix} T^{3 \times 3} & \mathbf{0}^{3 \times 3} \\ \mathbf{0}^{3 \times 3} & I^{3 \times 3} \end{bmatrix} \quad e \quad T = \begin{bmatrix} \cos(\alpha_N) & 0 & -\text{sen}(\alpha_N) \\ 0 & V_N^{-1} & 0 \\ \text{sen}(\alpha_N) & 0 & \cos(\alpha_N) \end{bmatrix}$$

$(\cdot)_N$  valor na condição de compensação.

Um sistema de controle de velocidade de rotação do eixo de estabilidade e ângulo de inclinação pode ser projetado pelo método de *feedforward* e de saída. Isso resulta nas seguintes equações de malha fechada.

$$\begin{aligned}u_c &= K_f r - K \bar{y}_s \\ \dot{\bar{x}}_s &= (\bar{A}_s - \bar{B}_s K \bar{C}_s) \bar{x}_s + \bar{B}_s K_f r\end{aligned}\tag{A.6}$$

sendo  $r = \begin{bmatrix} p_s^{ref} & \beta^{ref} \end{bmatrix}^T$  e  $K_f \in \mathfrak{R}^{2 \times 2}$ .

Como este é um sistema de múltiplas entradas, uma matriz de ganho de realimentação  $K$  é assintoticamente estável, pode ser calculada usando a técnica de alocação de autoestrutura.

$$\begin{aligned}\dot{\hat{x}}_s &= \hat{A}_s \hat{x}_s + \hat{B}_s u_c \\ \hat{y}_s &= \hat{C}_s \hat{x}_s\end{aligned}\tag{A.7}$$

sendo  $\hat{A}_s = \hat{T}\hat{A}\hat{T}^{-1}$ ,  $\hat{B}_s = \hat{T}\hat{B}$  e  $\hat{C}_s = I^{6 \times 6}$ .

As equações dinâmicas lateral-direcional são linearizadas entre as condições de 10kft e 200 kts e transformado em uma referência do eixo de estabilidade.

$$\hat{A}_s = \begin{bmatrix} -0.383 & 4.88 & 0.172 & 0 & 1.487 & -1.53 \\ -0.994 & -0.147 & 0.0024 & 0.094 & 0.0074 & 0 \\ 1.0017 & -13.84 & -1.476 & 0 & -12.01 & 0 \\ 0 & 0 & 1.0134 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -20.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -20.2 \end{bmatrix} \quad (\text{A.8})$$

$$\hat{B}_s = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 20.2 & 0 \\ 0 & 20.2 \end{bmatrix}$$

$$\hat{T} = \begin{bmatrix} \begin{pmatrix} 0.9867 & 0 & -0.1623 \\ 0 & 0.0097 & 0 \\ 0.1623 & 0 & 0.9867 \end{pmatrix} & 0^{3 \times 3} \\ 0^{3 \times 3} & I^{3 \times 3} \end{bmatrix}$$