



UNIVERSIDADE ESTADUAL DO MARANHÃO
CENTRO DE CIÊNCIAS TECNOLÓGICAS
PÓS-GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO E SISTEMAS

JOSÉ CHARLES MEDEIROS

**UM CONTROLADOR ROBUSTO LQR/LQG E
FILTRO DE KALMAN EM UM MANIPULADOR
ROBÓTICO VIA MODELO GENÉTICO NEURAL**

São Luís

2017

JOSÉ CHARLES MEDEIROS

**UM CONTROLADOR ROBUSTO LQR/LQG E
FILTRO DE KALMAN EM UM MANIPULADOR
ROBÓTICO VIA MODELO GENÉTICO NEURAL**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Computação e Sistemas da Universidade Estadual do Maranhão como parte dos requisitos para a obtenção do título de Mestre em Engenharia de Computação.

Orientador: Prof. Ivanildo Silva Abreu, Ph.D

Ph.D - USP

São Luís

2017

Medeiros, José Charles

Um controlador robusto LQR/LQG e filtro de Kalman em um manipulador robótico via modelo genético neural / José Charles Medeiros - 2017.

115.f

Dissertação (Mestrado) — Curso de Engenharia da Computação e Sitemas, Universidade Estadual do Maranhão, 2017.

Orientador: Prof. Ivanildo Silva Abreu.

1.Computação Evolutiva 2.Filtro de Kalman 3.Controlador robusto LQR/LQG 4.Redes neurais. I.Título.

CDU 681.5

JOSÉ CHARLES MEDEIROS

UM CONTROLADOR ROBUSTO LQR/LQG E FILTRO DE KALMAN EM UM MANIPULADOR ROBÓTICO VIA MODELO GENÉTICO NEURAL

Dissertação apresentada ao Programa do Mestrado Profissional de Engenharia da Computação e Sistemas da Universidade Estadual do Maranhão como parte dos requisitos para a obtenção do título de Mestre em Engenharia da Computação.

Aprovado em 22 de março de 2017

BANCA EXAMINADORA

Prof. Ivanildo Silva Abreu, Ph.D

Ph.D - USP

Prof. Leonardo Henrique Gonsioroski Furtado da Silva

Dr. Engenharia Elétrica - PUC-RJ

Prof. Carlos Henrique Rodrigues de Oliveira

Dr. Engenharia Elétrica - UNICAMP-SP

*Aos meus pais, filho, amigos e irmãos pela
compreensão e apoio que me deram para a ob-
tenção de mais um objetivo importante ao longo
da vida profissional.*

*Ao Professor Ivanildo, pelo companheirismo,
paciência e dedicação no ato de transmitir co-
nhecimento.*

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus, pois é Ele que dá sustentação em toda minha vida. A minha família, filho e irmã na colaboração e paciência ajudando de forma direta para o alcance de meu objetivo.

Ao professor Ivanildo pela orientação, amizade e principalmente, pela paciência, em conduzir de forma eficiente a realização deste trabalho de pesquisa.

Aos professores do Programa de Pós-graduação em Engenharia de Computação e Sistemas pela grandiosa colaboração na transmissão de conhecimento aos seus discentes, contribuindo assim, na formação profissional, pedagógica e ética. Aos funcionários da seção administrativa, que durante esses anos, contribuíram dando celeridade aos diversos tipos de documentos quando solicitados pelos alunos e também ao valoroso incentivo para a conclusão do curso.

*“Os princípios éticos elevados produzem
métodos comerciais eficazes”.*

James Watt (1736-1819)

RESUMO

Neste trabalho o objetivo é descrever o projeto do controlador robusto LQR/LQG e Filtro de Kalman e analisar o desempenho no domínio do tempo e da frequência utilizando um modelo Genético Neural. Para a resolução das estruturas de otimização restrita e irrestrita, usam-se técnicas de Inteligência Artificial, tais como Algoritmos Genéticos e Rede Neural Recorrente de múltipla camadas fundamentado nas interações de respostas no domínio do tempo e da frequência, bem como, as ilustrações da função energia. Através dos pesos da RNR pode-se concluir que a abordagem apresentada é uma boa proposta para a solução do problema.

Palavras-chaves: Computação Evolutiva, Filtro de Kalman, Controlador Robusto LQR/LQG , Rede neuronal.

ABSTRACT

In this work the objective is to describe the project of the robust controller LQR/LQG and Kalman Filter and analyze the performance in the domain of time and frequency using a Genetic Neural model. For the resolution of the restricted and unrestricted optimization structures, Artificial Intelligence techniques such as Genetic Algorithms and multiple layer Recurrent Neural Networks are used, based on the interactions of time and frequency domain responses, as well as the function illustrations energy. Through the weights of RNR one can conclude that the presented approach is a good proposal for the solution of the problem.

Keywords: Evolutive Computation, Kalman filter, Controller Robust LQR / LQG, Neural network.

Lista de Figuras

2.1	Diagrama bloco do sistema LTI	23
2.2	Sistemas de realimentação positivo e negativo	25
3.1	Sistema multivariável	29
3.2	Sistema MIMO	29
3.3	Diagrama bloco padrão de malha fechada	30
3.4	Diagrama bloco de um sistema realimentado MIMO do tipo linear	32
3.5	Sistema $G(s)$ com entrada d e saída y	35
3.6	Configuração de malha fechada padrão	39
4.1	Um neurônio e o esquema da sinapse	47
4.2	Modelo para o funcionamento do neurônio	48
4.3	Estrutura de um axônio	49
4.4	Função logística	50
4.5	Função tangente hiperbólica	50
4.6	Exemplo de uma rede multicamadas	51
4.7	Exemplo de uma rede de neurônios com retroalimentação	52
4.8	O <i>perceptron</i>	52
4.9	Arquitetura gráfica de um <i>perceptron</i> multicamada com duas camadas intermediárias	53
4.10	A rede de <i>Kohonen</i>	54
4.11	Lei de aprendizagem da rede de <i>Kohonen</i>	54
4.12	Rede de associação linear	55
4.13	Rede de função de base radial de complexidade reduzida	56

4.14	Estrutura de uma rede neural de base radial (RNFBR)	57
4.15	Classificação dos métodos de otimização	60
4.16	Exemplo de uma população do AG	64
4.17	Exemplo do operador de mutação	66
4.18	Esquemático do AG	67
5.1	Manipulador robótico a esquerda e sua aplicação industrial a direita	71
5.2	Anatomia do braço robótico fabricado pela RRRobotica	72
5.3	Esquema de notação de um braço mecânico ilustrativo	73
5.4	Manipulador robótico e ambiente	73
6.1	Diagrama dos métodos utilizados no controle ótimo LQR	75
7.1	Arquitetura do RNAR para a EAR	82
8.1	Diagrama de blocos do sistema	86
8.2	Diagrama de blocos do Filtro de <i>Kalman</i>	87
9.1	Análise dos deslocamentos do Manipulador Robótico e ambiente-alvo. . . .	90
9.2	Sensibilidade total de cada indivíduo	91
9.3	Evolução do processo - função objetivo média e melhor função objetivo das gerações	92
9.4	<i>Fitness</i> normalizada de cada geração e de cada indivíduo	94
9.5	Superfície da função energia para $\eta_p = 1000 - 10000$ e $\eta_z = 1 - 10$	95
9.6	Superfície da norma infinito para $\eta_p = 1000 - 10000$ e $\eta_z = 1 - 10$	95
9.7	Comportamento das camadas da RNA	96
9.8	Comportamento das camadas da RNA	96
9.9	Trajetórias dos estados versus os estimados	98
9.10	Erro de estimação para cada estado	98

A.1 Fluxograma do AG	107
I.1 Sistema regulador ótimo	108

Lista de Tabelas

8.1	Parâmetros duais do LQR e do FK	87
9.1	Pop. final AG-sensibilidades e função objetivo dos indivíduos viáveis	93
9.2	Melhor indivíduo da população final-autovalores e traços das matrizes de ponderação	93

Lista de Abreviaturas e Siglas

AG	Algoritmo Genético
BIBO	<i>Bounded-Input Bounded-Output</i>
CE	Computação Evolutiva
DVOR	<i>Doppler Very High Frequency Omnidirectional Radio Range</i>
EAR	Equação Algébrica de Riccati
FK	Filtro de <i>Kalman</i>
GDL	Graus de Liberdade
IA	Inteligência Artificial
ILS	<i>Instrument Landing System</i>
LIT	Linear Invariante no Tempo
LQR	<i>Quadratic Linear Regulator</i>
LQG	Linear Quadrático Gaussiano
MIMO	<i>Multiple input Multiple output</i>
PD	Proporcional Derivativo
PID	Proporcional Integrativo Derivativo
RUR	<i>Rosumovi Univerzálni Roboti</i>
RNR	Rede Neural Recorrente
RNAR	Rede Neural Artificial Recursiva
SISO	<i>Single-Input - Single-Output Landing Sytems</i>
SVD	<i>Singular Value Decomposition</i>
VHF	<i>Very High Frequency</i>

Lista de Símbolos

A	Matriz de ordem $n \times n$
B	Matriz de ordem $n \times 1$
$crom$	Número de cromossomos
$H(t)$	Hamiltoniano
J	Funcional
P_c	Matriz de controlabilidade $n \times m$
P_o	Matriz de observabilidade $n \times n$
C	Vetor linha $1 \times n$
\mathbb{R}^n	Conjunto do vetores reais de dimensão n
\mathbb{R}^m	Conjunto do vetores reais de dimensão m
\vec{x}	Vetor coluna $n \times 1$
Ξ, Θ	Matrizes de covariância
$\delta(t - \tau)$	Delta de <i>Dirac</i>
$\bar{\sigma}(G)$	Valor singular máximo
$\underline{\sigma}(G)$	Valor singular mínimo

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Objetivo geral	19
1.2	Objetivos específicos	19
1.3	Justificativa	20
1.4	Organização da dissertação	21
2	SISTEMAS LINEARES INVARIANTES NO TEMPO	23
3	SISTEMAS MULTIVARIÁVEIS	29
3.1	Sistemas multivariáveis MIMO	32
3.2	Análise multivariável MIMO da resposta em frequência	34
3.3	Obtendo a resposta de frequência de $G(s)$	35
3.4	Direções em sistema multivariáveis	37
3.5	Sensibilidade e cosensibilidade	39
3.6	Decomposição do valor singular	42
3.7	Valores singulares para desempenho	44
4	SISTEMAS INTELIGENTES DE CONTROLE	45
4.1	Sistemas baseados em redes “neurais”	47
4.1.1	Rede de neurônios artificial - multicamadas	50
4.1.2	Rede Neural - rede de neurônios artificial	51
4.1.3	Principais arquiteturas de redes neurais artificiais	52
4.2	Sistemas baseados em algoritmos genéticos (AG)	58
4.2.1	Algoritmos de otimização	59

4.2.2	Comparação dos algoritmos genéticos com outras técnicas de otimização	62
4.2.3	Vantagens e limitações do algoritmo genético	63
4.2.4	Aplicações dos algoritmos genéticos	63
4.2.5	Algoritmo genético (AG) com parâmetros codificados binariamente	64
4.2.6	Algoritmo genético (AG) com parâmetros contínuos	66
5	MANIPULADOR ROBÓTICO	71
6	CONTROLE INTELIGENTE	74
6.1	Modelagem clássica do projeto LQR/LQG	75
6.2	Modelagem inteligente utilizando o Algoritmo Genético	77
7	SOLUÇÃO NEURAL DA EQUAÇÃO ALGÉBRICA DE <i>RICCATI</i>	80
7.0.1	Análise de estabilidade	82
7.0.2	Análise da solvabilidade	82
8	FILTRO DE KALMAN PARA ESTIMAÇÃO DOS ESTADOS	84
9	EXPERIMENTO COMPUTACIONAL DO MANIPULADOR ROBÓTICO	88
9.1	Modelamento matemático do manipulador	88
9.1.1	Equações de estado e de saída do manipulador robótico	89
9.2	Desempenho da população inicial	91
9.3	Desempenho da população final	92
9.4	Análise da RNR na solução da EAR	95
9.5	Análise da filtragem de Kalman	98
10	CONCLUSÃO	99
10.1	Produção científica	100

10.2 Trabalhos futuros	101
REFERÊNCIAS	102
A Apêndice - FLUXOGRAMA DO AG	107
I Anexo - SISTEMAS REGULADORES QUADRÁTICOS ÓTIMOS	108
I.1 O problema do regulador quadrático ótimo	108
II Anexo - ESTABILIDADE DOS SISTEMAS LTI	111
II.1 Estabilidade de entrada e saída de sistemas LTI	111

1 INTRODUÇÃO

A Explosão tecnológica do século XX que teve aceleração com advento dos computadores e sistemas de controle resultaram em um avanço muito grande no campo das ciências. Estes, por sua vez, permeiam a vida de todas as sociedades avançadas de hoje. Os mesmos agiram e agem como catalisadores na promoção do progresso e desenvolvimento, impulsionando assim, a sociedade do século atual. Com o desenvolvimento tecnológico, tornaram-se possíveis o surgimento de equipamentos e máquinas capazes de operar de forma eficiente em um tempo de trabalho reduzido. Surgem então trens de alta velocidade (trens-balas), veículos capazes de explorar outros planetas, sofisticadas aeronaves civis e militares, eficientes linhas de montagem robotizadas, eficientes controles de poluição sejam em fábricas ou cidades, automóveis híbridos complexos, sistemas de controle de navegação aérea (Radares, ILS, DVOR, VHF), entre outros benefícios. Logo, a operação bem sucedida de todos esses sistemas dependem do funcionamento adequado e sem riscos na utilização.

Percebe-se então, que os chamados **sistemas** têm por objetivo fornecer produtos economicamente úteis para sociedade. Essa por sua vez usufruindo de todos os benefícios que vierem a surgir no controle de máquinas e processos industriais e econômicos.

A aplicação dos sistemas de controle para benefícios da sociedade é utilizada desde o período de 300 a 1 a.C., na Grécia [1, 2, 3, 4]. Cita-se como exemplo, o relógio de água de *Ktesibios* que utilizava um regulador com boia e a lâmpada a óleo inventada por *Philon* em aproximadamente 250 a.C. que utilizava um regulador com boia para manter um nível constante de óleo combustível. *Heron* de Alexandria, que viveu no primeiro século d.C., publicou o primeiro livro intitulado *pneumática*, no qual esboçou várias formas de mecanismos de nível de água usando reguladores com boia [2].

No século XVIII surgiu o primeiro trabalho significativo de controle automático construído por *James Watt* [5] para controle da máquina a vapor. Outros trabalhos importantes se devem a *Minorsky*, *Hazen* e *Nyquist*, entre outros. Em 1922, *Minorsky* trabalhou em controladores automáticas para a pilotagem de embarcações e demonstrou

como a **estabilidade** poderia ser determinada a partir de equações diferenciais que descrevem o sistema. Em 1932, Nyquist desenvolveu um procedimento relativamente simples para a determinação da estabilidade de **sistemas de malha fechada** com base na proposta de **malha aberta** a excitações senoidais estacionárias. Em 1934, *Hazen* introduziu o termo **servomecanismo** para sistemas de controle de posição, discutiu o projeto de servomecanismo a relé capazes de acompanhar uma variação da entrada com **acurácia**.

Na Europa moderna, foi inventado o primeiro sistema com realimentação que foi o regulador de temperatura de *Cornelis Drebbel* (1572-1633) da Holanda [2]. *Dennis Papin* (1647-1712) inventou o primeiro regulador de pressão para caldeiras a vapor em 1681.

O grande impulso para teoria e a prática do controle automático ocorreu durante a Segunda Guerra Mundial [6], com surgimento de aeronaves com pilotos automáticos, sistemas de posicionamento de armas, sistemas de controle de antenas de radares e outros sistemas militares baseados na abordagem do controle com realimentação. Técnicas no domínio da frequência e do lugar das raízes, essenciais à teoria de controle clássico, eram bem estudadas nessa época. Sendo para isso, necessários o uso das Transformadas de *Laplace* e do plano complexo de frequência, isto é, a utilização da matemática se tornava essencial.

Durante a década de 1940, método de resposta em frequência (especialmente os métodos com base nos diagramas de *Bode*) tornaram possível o surgimento de sistemas de controle linear de malha fechada que satisfizessem o **desempenho** requerido [7]. Muitos desses sistemas usavam controladores PID (Proporcional Integral Derivativo) no controle de pressão e temperatura.

À medida que os sistemas modernos se tornam mais e mais complexos, a sua descrição no campo da matemática, se tornava mais complexa, pois aumentava o uso de equações para representarem esses sistemas [1]. A partir de 1960, com a disponibilidade dos computadores digitais, possibilitou-se a análise de sistemas complexos diretamente no domínio do tempo. A teoria de controle moderno, com base na análise e na síntese do domínio de tempo com o emprego de variáveis de estado, foi desenvolvida para lidar com a crescente complexidade dos sistemas modernos e seus rigorosos requisitos, como a precisão, a importância e ao custo em aplicações militares, espaciais e industriais.

No intervalo entre 1960 a 1980, o controle ótimo de **sistemas determinísticos**

e **estocásticos** foi amplamente pesquisado, bem como o controle adaptativo e o controle de aprendizagem de sistemas complexos. Entre 1980 a 1990, surgem desenvolvimentos na teoria do controle moderno, o **controle robusto**. Tal teoria baseia-se em estabelecer um sistema **estável** com o mínimo de erro possível estabelecido entre o sistema real e seu modelo [5]. Essa teoria incorpora tanto abordagem de resposta em frequência quanto abordagem de domínio do tempo.

Portanto, um dos tópicos de maior atenção da comunidade de controle nos últimos tempos tem sido o estudo da **robustez** dos sistemas a realimentação [8]. Sem falar a importância também da **estabilidade** em um sistema que é o objetivo principal em qualquer projeto. Sem **estabilidade** de alguma variável significativa pode-se afirmar que o sistema não possui utilidade industrial. Um exemplo de métodos que são utilizados para se obter um controle robusto em sistemas de controle é o método LQG/LTR. Sua principal característica é a robustez e é aplicada em sistemas multivariáveis, de natureza sistemática numa abordagem frequencial. Aplica-se, também, em sistemas lineares invariantes no tempo [8].

Com a crescente complexidade dos sistemas de controle industriais e comerciais, surgem então novos estudos e metodologias para o controle eficiente de qualquer processo dinâmico. A Inteligência Artificial (IA) é a parte da ciência da computação envolvida no projeto de sistemas que exibem características que associamos com a inteligência no comportamento humano [9, 10].

Assim várias técnicas na área de IA são utilizadas para se obter diversos tipos de sistemas “*inteligentes*”, para a síntese de sistemas de controle ótimo do tipo LQR (Regulador Linear Quadrático).

Com o crescente desenvolvimento tecnológico, torna-se necessário o projeto de sistemas de controle complexos e de alta exatidão, os chamados *controles ótimos*, neles a **controlabilidade** de determinados sistemas é feita na mais adequada exatidão com erros praticamente desprezíveis. Portanto, os principais obstáculos no projeto de sistemas de controle de alto desempenho (controle robusto) são os erros de modelagem, variações nos parâmetros do sistemas a ser controlado, ruídos e distúrbios devido as forças de impacto e as forças de estado estacionário, alterando assim, impreterivelmente, o comportamento dos sistemas a serem controlados. O **manipulador robótico**, objeto desse estudo, é um exemplo dentre as várias aplicações práticas de um sistema de controle complexo devendo

ser perfeitamente **controlado** e apresentar **robustez** aos efeitos dos fatores, sejam eles internos ou externos ao sistema.

Toda a análise do controlador robusto LQR/LQG é realizada em um manipulador robótico e os resultados apresentaram um maior desempenho com relação a robustez avaliadas e comprovadas por meio das barreiras de desempenho e estabilidade robustas. Procura-se também realizar uma abordagem teórica das ferramentas empregadas no desenvolvimento do projeto do controlador robusto LQR/LQG que são o Regulador Linear Quadrático (LQR) e o Filtro de *Kalman* (FK), bem como apresentar a solução da Equação Algébrica de *Ricatti* (EAR) utilizando rede neuronal.

1.1 Objetivo geral

Solucionar o problema ótimo utilizando o projeto LQR/LQG, bem como estimar as variáveis de estado utilizando o Filtro de Kalman(FK).

1.2 Objetivos específicos

- Levantar o estado da arte em relação a análise e projeto de sistemas de controle em espaço de estados;
- Modelar o sistema de controle do manipulador robótico;
- Obter o ganho do compensador através do projeto LQR;
- Analisar resultados no domínio da frequência via diagrama de Bode Multivarável;
- Encontrar a solução da Equação Algébrica de Riccati utilizando rede neuronal;
- No desenvolvimento do projeto utilizar o Regulador Linear Quadrático (LQR) e o Filtro de Kalman (FK);
- Utilizar o software Matlab para simular os dados obtidos.

1.3 Justificativa

A teoria do controle moderno trata de sistemas que têm características de auto-organização, adaptação, robustez e otimalidade [1]. Baseado nessa afirmação a **automação** em seu uso moderno surge como uma tecnologia que usa comandos programados para operar um dado processo combinado com a realimentação de informação para determinar se os comandos foram executados apropriadamente. Isso garante, nesses tipos de sistemas, quando *automatizados*, a operar sem assistência ou interferência humana, reduzindo assim, o tempo de trabalho com maior exatidão e precisão. Já os processos *semiautomatizados* são aqueles que incorporam tanto humanos quanto robôs.

Desta forma será utilizado nesse trabalho, um manipulador robótico que tem como principais funções a *rapidez*, *o controle preciso* e *uniformidade* quando utilizado nas linhas de produção e sendo controlado de modo que as forças de impacto, bem como as forças em regime permanente, não danifique os objetos. Desse modo são apresentadas algumas justificativas para a utilização do projeto LQR e Filtro de *Kalman* (FK):

- O uso do controlador LQR justifica-se devido a sua margem de ganho ser infinita e a margem de fase ser de 60° via diagrama de *Bode* na análise frequencial;
- Apresentar um procedimento fundamentado em Inteligência Artificial uma vez que pelo levantamento do estado da arte constatou-se poucos artigos referente a este procedimento;
- O uso da teoria do controle moderno, pois este trabalha com sistemas que têm características de auto-organização, adaptação, robustez e otimalidade [11] ;
- A utilização de milhares de robôs industriais e laboratoriais aplicados nas indústrias. Visando substituir o trabalho humano em tarefas repetitivas, prejudiciais à saúde ou perigosas, e sua utilização em linhas de montagem ou linhas de produção permitem aumentar a produtividade, melhorar a qualidade da produção e reduzir os custos;
- O crescente uso de manipuladores robóticos que tem como principais funções a rapidez, o controle preciso e uniformidade quando utilizado nas linhas de produção.

1.4 Organização da dissertação

A presente dissertação está organizada em Capítulos, Apêndices e Anexos. No Capítulo 1 foi descrito uma síntese histórica dos sistemas de controle, bem como, seus benefícios para a sociedade, tornando assim, uma parte importante no avanço no campo das ciências.

No Capítulo 2 aborda-se os **Sistemas Lineares Invariantes no Tempo** (LIT) para a solução de equações diferenciais lineares com coeficientes constantes, e uma abordagem de estabilidade desses sistemas estão no Anexo B.

No Capítulo 3 apresenta-se os *Sistemas multivariáveis* destacando sua importância dentro do projeto do controle robusto, bem como a realização da análise da resposta frequencial para o sistema MIMO. Aborda-se também, a influência da sensibilidade e co-sensibilidade na equações de entrada, saída e de erro, assim como a decomposição em valores singulares e suas especificações para o desempenho em altas e baixas frequências.

No Capítulo 4 é dada uma pequena abordagem sobre *Sistemas inteligentes de controle* baseados na *Inteligência Artificial* (IA), e as técnicas surgidas a partir dessa abordagem. Dentre essas técnicas, as abordadas neste trabalho, são os sistemas baseados em *redes neurais* e os baseados em *algoritmos genéticos*.

No Capítulo 5 apresenta-se o *Manipulador robótico* que são sistemas multifuncionais, ressaltando assim, sua funcionalidade e aplicabilidade na engenharia de controle moderno, bem como o modelo utilizado nessa dissertação.

No Capítulo 6 aborda-se o *Controle inteligente* destacando as técnicas do controle moderno, o *Regulador Linear Quadrático* (LQR) com sua teoria apresentada no Anexo A, assim como os métodos de controle inteligentes para a resolução do problema do *LQR*, a saber, o algoritmo genético para a solução das matrizes \mathbf{Q} e \mathbf{R} e a rede neural recorrente para a solução da *Equação Algébrica de Riccati* (EAR) representados pelo fluxograma do Apêndice A.

No Capítulo 7 é apresentado as equações dinâmicas para a solução neural da *EAR*, além de ilustrar uma arquitetura da rede neuronal, assim como, a análise de estabilidade e solvabilidade dessa rede.

No Capítulo 8 aborda-se o *Filtro de Kalman* para a estimação dos estados através da alocação de autovalores e autovetores, bem como a utilização do *Algoritmo*

genético para este filtro (FK) a fim de obter as matrizes de covariância Ξ e Θ . Faz-se também a relação de dualidade do *FK* e o *LQR*.

No Capítulo 9 são feitas as análises e discussões do trabalho apresentado, bem como toda a formulação matemática do **Manipulador robótico** utilizado. São realizadas também, análises da população inicial e final, traçando assim, o perfil de cada indivíduo dessas populações através dos valores da **Função objetivo** no processo de busca do indivíduo que possui *QR* ótimo. As análises e conclusões são feitas para a rede neural na solução da *EAR* e da filtragem de *Kalman*, verificando assim, as trajetórias dos estados versus os estados estimados.

Por fim, no Capítulo 10 são apresentados as conclusões e comentários, sobre a metodologia empregada nessa dissertação no desenvolvimento dos controladores *LQR* e *LQG* de uma forma particular, aplicado nesse caso, ao problema do **Manipulador robótico**. Apresenta-se também a produção científica (os trabalhos submetidos, aceitos e publicados) e as sugestões para trabalhos futuros.

2 SISTEMAS LINEARES INVARIANTES NO TEMPO

A fim de se analisar um sistema dinâmico completamente torna-se necessário um modelo matemático preciso que descreve tal sistema. Essa descrição pode ser representada por equações diferenciais conhecidas ou por dados de ensaios experimentais [12]. A capacidade de analisar o sistema e determinar o seu desempenho depende de como as características podem ser expressas matematicamente. Existem técnicas matemáticas para a solução de equações diferenciais lineares com coeficientes constantes. O sistema considerado neste trabalho é baseado nesse tipo de equações diferenciais, e é chamado de *Sistema Linear Invariante no Tempo* (LIT), em que a relação entre a entrada e a saída desse sistema é independente do tempo. Portanto, os sistemas LIT são aqueles que atendem as propriedades de linearidade e invariância no tempo simultaneamente, ou seja:

$$\text{Se } y_1(t - \tau) = T \{x_1(t - \tau)\} \quad \text{e} \quad y_2(t - \tau) = T \{x_2(t - \tau)\}$$

$$\text{Então } T \{\alpha \cdot x_1(t - \tau) + \beta \cdot x_2(t - \tau)\} = \alpha \cdot y_1(t - \tau) + \beta \cdot y_2(t - \tau)$$

A Figura 2.1 apresenta o diagrama bloco desse tipo de sistema onde τ representa um atraso de tempo arbitrário.

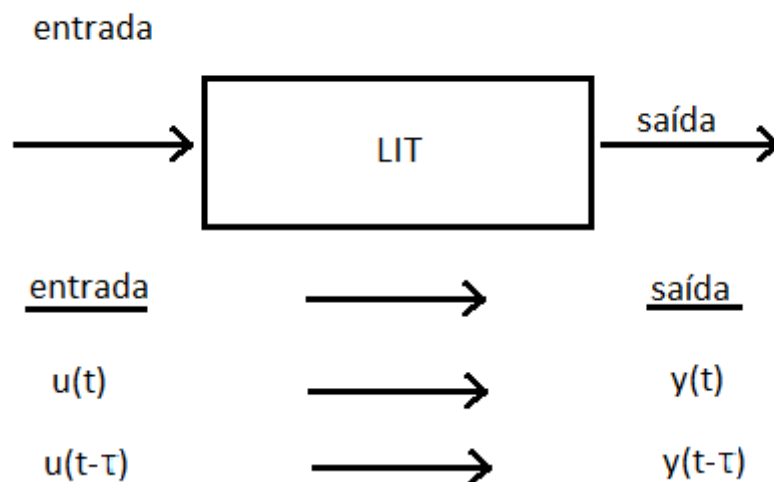


Figura 2.1: Diagrama bloco do sistema LTI

Como exemplo desse tipo de sistema, temos:

- Sistema quadrático

$$y(t) = x^2(t)$$

- Diferenciação

$$y(t) = \frac{d}{dt}x(t)$$

$$\frac{d}{dt}x(t - \tau) = y(t - \tau)$$

- Integração

$$y(t) = \int_{-\infty}^t x(\lambda)d\lambda$$

$$y(t - \tau) = \int_{-\infty}^{t-\tau} x(\lambda)d\lambda = \int_{-\infty}^{t-\tau} x(\lambda - \tau)d\lambda$$

Em [13] um sistema é dito Invariante no Tempo se para cada par de estado de entrada-saída

$$\left. \begin{array}{l} \mathbf{X}(t_0) \\ u(t), \quad t \geq t_0 \end{array} \right\} \implies y(t), \quad t \geq t_0$$

e qualquer T , temos

$$\left. \begin{array}{l} \mathbf{X}(t_0 + T) \\ u(t - T), \quad t \geq t_0 + T \end{array} \right\} \implies y(t - T), \quad t \geq t_0 + T \quad (\text{deslocamento de tempo}).$$

Isto significa que o estado inicial é deslocado para um tempo $t_0 + T$ e a mesma forma de onda de entrada é aplicada para $t_0 + T$ em vez de t_0 , então a forma de onda de saída deverá ser a mesma, exceto as que começam aparecer para um tempo $t_0 + T$. Em outras palavras, se o estado inicial e a entrada são as mesmas, não importa em que momento são aplicados a forma de onda de saída deverá sempre ser a mesma. Portanto, para sistemas invariantes no tempo, podemos assumir, sem perda de generalidade, que $t_0 = 0$. Se o sistema é não invariante no tempo diz-se que o tempo é variável.

A invariância de tempo é definida para sistemas, não para sinais. Sinais são na maioria das vezes variantes no tempo. Se um sinal é invariante no tempo tal como

$u(t) = 1$ para todo t , então o sinal é muito simples ou trivial. As características de sistemas invariantes no tempo podem ser independentes do tempo.

Alguns sistemas físicos podem ser modelados em sistemas invariantes no tempo por exemplo, a combustão de um foguete é um sistema invariante no tempo porque sua massa decresce rapidamente com o tempo. Logo um grande número de sistemas físicos podem se modelados como sistemas invariantes no tempo sobre um período de tempo limitado.

Tipos de entrada - saída

As respostas de estado zero de um sistema linear pode ser descrito por

$$y(t) = \int_{t_0}^t g(t, \tau)u(\tau)d\tau. \quad (2.1)$$

Agora se o sistema é invariante no tempo, então temos

$$g(t, \tau) = g(t + \tau, \tau + T) = g(t - \tau, 0) = g(t - \tau).$$

para qualquer T . Logo (2.1) reduz para

$$y(t) = \int_0^t g(t, \tau)u(\tau)d\tau = \int_0^t g(\tau)u(t - \tau)d\tau \quad (2.2)$$

Onde substituímos t_0 por 0 (zero). A integração (2.2) é chamada de integração de convolução. Ao contrário do caso do tempo variável em que g é uma função de duas variáveis, g é uma função de uma variável simples no caso do tempo invariante. Por definição $g(t) = g(t - 0)$ é uma saída em um tempo t devido a entrada de um impulso aplicado em tempo 0(zero). A condição para um sistema linear invariante no tempo ser **causal** é $g(t) = 0$ para $t < 0$.

Exemplo 2.0.1. Considere um sistema com realimentação única mostrada na Figura 2.2

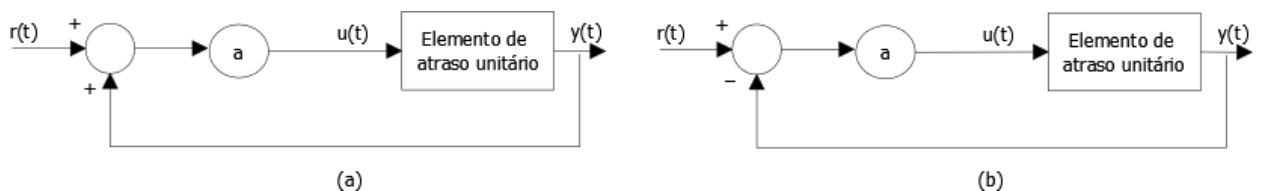


Figura 2.2: Sistemas de realimentação positivo e negativo

Fonte: Adaptado de [14]

(a) consiste de multiplicador com ganho a em um elemento de atraso unitário. Este é um sistema SISO. Se $r(t)$ é a entrada de um sistema de realimentação e se

$r(t) = \delta(t)$, então a saída à resposta do impulso do sistema de realimentação é igual

$$gf(t) = a\delta(t-1) + a^2\delta(t-2) + a^3\delta(t-3) + \dots = \sum_{i=1}^{\infty} a^i \delta(t-i). \quad (2.3)$$

Se $r(t)$ é qualquer entrada com $r(t) \equiv 0$ para $t < 0$; então a saída é dada por

$$y(t) = \int_0^t gf(t-\tau)r(\tau)d\tau = \sum_1^{\infty} a^i \int_0^t \delta(t-\tau-i)r(\tau)d\tau = \sum_1^{\infty} \Big|_{\tau=1-i}^{\infty} = \sum_1^{\infty} a^i r(t-i)$$

porque o sistema de atraso de tempo unitário é distribuído, assim é o sistema de realimentação.

Matrizes das funções de transferências

A transformada de Laplace é uma importante ferramenta no estudo de sistema invariantes no tempo. Se $\hat{y}(s)$ é a transformada de Laplace de $y(t)$, isto é:

$$\hat{y}(s) = \int_0^{\infty} y(t)e^{-st}dt.$$

Para sistemas causal, temos $g(t) = 0$ para $t < 0$ ou $g(t-\tau) = 0$ para $\tau > t$. Assim o limite de integração superior em (2.2) deve ser substituído por ∞ . Substituindo (2.2) e permutando a ordem de integração, obtemos

$$\begin{aligned} \hat{y}(s) &= \int_{t=0}^{\infty} \left(\int_{\tau=0}^{\infty} g(t-\tau)u(\tau)d\tau \right) e^{-s(t-\tau)} \cdot e^{-s\tau} d\tau \\ &= \int_{\tau=0}^{\infty} \left(\int_{t=0}^{\infty} g(t-\tau)e^{-s(t-\tau)} d\tau \right) u(\tau)e^{-s\tau} d\tau \end{aligned}$$

que tornar-se, após a introdução de uma nova variável $\nu = t - \tau$,

$$\hat{y}(s) = \int_{\tau=0}^{\infty} \left(\int_{\nu=-\tau}^{\infty} g(\nu)e^{-s\nu} d\nu \right) u(\tau)e^{-s\tau} d\tau$$

Usando novamente a condição de causalidade para substituir o limite de integração inferior dentro do parêntese para $\nu = -\tau$ a $\nu = 0$, a integração torna-se independente para τ e as duas integrações tornam-se

$$\hat{y} = \int_{\nu=0}^{\infty} g(\nu)e^{-s\nu} d\nu \int_{\tau=0}^{\infty} u(\tau)e^{-s\tau} d\tau$$

ou

$$\hat{y}(s) = \hat{g}(s)\hat{u}(s) \quad (2.4)$$

onde

$$\hat{g}(s) = \int_0^{\infty} g(t)e^{-st}dt$$

é chamada de função de transferência do sistema. Assim a função de transferência é a transformada de Laplace da resposta ao impulso e, reciprocamente, a resposta ao impulso é a transformada inversa de Laplace da função de transferência. A transformada de Laplace transforma a integral de convolução em (2.2) em uma equação algébrica em (2.3). Em análise e projeto, é mais simples usar equações algébricas do que usar convoluções.

Para um sistema de entrada p e saída q , (2.2) poder ser estendida como

$$\begin{bmatrix} \hat{y}_1(s) \\ \hat{y}_2(s) \\ \vdots \\ \hat{y}_q(s) \end{bmatrix} = \begin{bmatrix} \hat{g}_{11}(s) & \hat{g}_{12}(s) & \cdots & \hat{g}_{1p}(s) \\ \hat{g}_{21}(s) & \hat{g}_{22}(s) & \cdots & \hat{g}_{2p}(s) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{g}_{q1}(s) & \hat{g}_{q2}(s) & \cdots & \hat{g}_{qp}(s) \end{bmatrix} \cdot \begin{bmatrix} \hat{u}_1(s) \\ \hat{u}_2(s) \\ \vdots \\ \hat{u}_p(s) \end{bmatrix}$$

ou

$$\hat{\mathbf{y}}(s) = \hat{\mathbf{G}}(s)\hat{\mathbf{u}}(s)$$

onde $\hat{g}_{ij}(s)$ é a **função transferência** de entrada j e saída i . A matriz $\hat{\mathbf{G}}(s)$ de ordem $q \times p$ é chamada de matriz de função de transferência ou simplesmente, matriz de transferência do sistema.

Equação de espaço de estado

Todo sistema linear invariante no tempo agrupado pode ser descrito por um conjunto de equações da forma

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (2.5)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t). \quad (2.6)$$

Para o sistema com p entradas, q saídas e n variáveis de estado $\mathbf{A}, \mathbf{B}, \mathbf{C}$ e \mathbf{D} são respectivamente $n \times n$, $n \times p$, $q \times n$ e $q \times p$ matrizes constantes. Aplicando a transformada de Laplace em (2.5) e (2.6) produz

$$s\hat{\mathbf{x}}(s) - \mathbf{x}(0) = \mathbf{A}\hat{\mathbf{x}}(s) + \mathbf{B}\hat{\mathbf{u}}(s)$$

$$\hat{\mathbf{y}}(s) = \mathbf{C}\hat{\mathbf{x}}(s) + \mathbf{D}\hat{\mathbf{u}}(s)$$

que implica

$$s\hat{\mathbf{x}}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{x}(0) + (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\hat{\mathbf{u}}(s) \quad (2.7)$$

$$\hat{\mathbf{y}}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{x}(0) + \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\hat{\mathbf{u}}(s) + \mathbf{D}\hat{\mathbf{u}}(s). \quad (2.8)$$

Elas são equações algébricas dado $\mathbf{x}(0)$ e $\hat{\mathbf{u}}(s)$, $\hat{\mathbf{x}}(s)$ e $\hat{\mathbf{y}}(s)$ e podem ser calculadas algebricamente em (2.7) e (2.8). Sua transformada inversa produz o tempo de resposta $\mathbf{x}(t)$ e $\mathbf{y}(t)$. As equações mostram de fato que as respostas do sistema linear são decompostas como resposta de estado zero e respostas de entrada zero. Se o estado inicial $\mathbf{x}(0)$ é zero, então (2.8) reduz para

$$\hat{\mathbf{y}}(s) = [\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}] \hat{\mathbf{u}}(s)$$

comparando esta com (2.4) produz

$$\hat{\mathbf{G}}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}. \quad (2.9)$$

Esta relaciona a entrada- saída (ou matriz de transferência) e a representação no espaço de estado.

Portanto, a transformada Laplace não é usada para estudo de sistema lineares variáveis no tempo. A transformada de Laplace de $g(t)$ é uma função de duas variáveis no tempo e $[\mathbf{A}(t)\mathbf{x}(t)] \neq [\mathbf{A}(t)] \cdot [\mathbf{x}(t)]$. Assim a transformada de Laplace não oferece qualquer vantagem, e por isso não é usada em estudo de sistemas variáveis no tempo.

3 SISTEMAS MULTIVARIÁVEIS

Muitos são os desafios futuros no que diz respeito a sistemas robustos complexos em relação a análise e otimização de projetos grandes e complexos. Sendo assim, a estabilidade e robustez desses sistemas tem que ser de suma importância quando se trata de projeto de sistemas de controle robusto **multivariável**. Chama-se multivariável devido o mesmo possuir *múltiplas entradas e múltiplas saídas*.

Nesse projeto é abordado um sistema multivariável chamado MIMO (*Multi-Input - Multi-Output*, Múltiplas entradas-Múltiplas saídas). Nesse sistema a entrada não afeta somente a sua saída correspondente, mas também as outras saídas da planta. As Figuras 3.1 e 3.2 mostram a representação desse sistema.

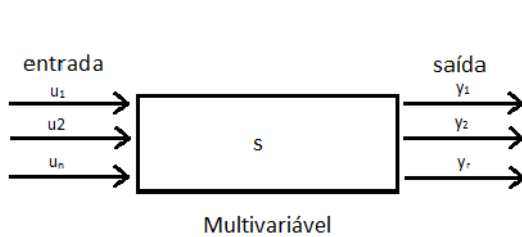


Figura 3.1: Sistema multivariável

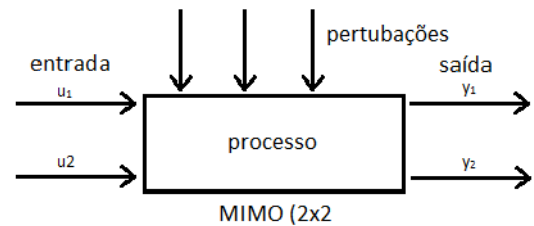


Figura 3.2: Sistema MIMO

Nesse tipo de sistema há uma interação ou acoplamento entre as entradas e as saídas [15]. Desse modo essa característica aumenta a complexidade na interpretação de seu comportamento fazendo com que o projeto e análise de sistemas multivariáveis sejam mais complexos do que os monovariáveis, os chamados SISO (*Single-Input - Single-Output*, Única entrada-Única saída). Uma outra diferença entre os sistemas SISO e MIMO é a presença de direções no MIMO [15]. Sendo essas direções relevantes para vetores e matrizes, mas não para escalares. Entretanto, apesar das diferenças, a maioria dos argumentos utilizados na demonstração do caso MIMO são os mesmos do caso SISO [8].

Na figura 3.3 é ilustrado um diagrama em bloco padrão de um sistema com realimentação no domínio da frequência [8, 16].

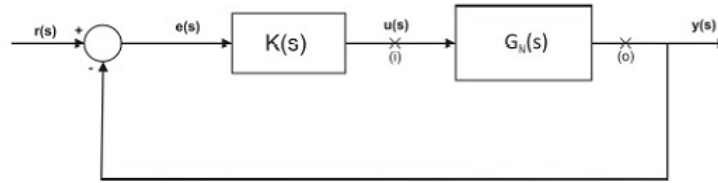


Figura 3.3: Diagrama bloco padrão de malha fechada

No projeto de sistemas de controle, as equações diferenciais lineares formam uma base importante para a representação ou aproximação em sistemas dinâmicos não lineares (mundo real) [17]. Diversos sistemas apresentam dinâmicas que são importantes para altas frequências que muitas das vezes, são negligenciados em um modelo do projeto. Estas dinâmicas não modeladas em alta frequência podem atuar para desestabilizar um sistema de controle que pode ter um comportamento bastante adequado apenas em termos do modelo do sistema.

Além disso, à medida que o sistema não linear muda seu ponto de operação de equilíbrio, o modelo de planta linearizado que descreve seu comportamento também muda com as perturbações introduzidas no sistema. Esta variação de parâmetro é um efeito que ocorre em baixas frequências que também pode atuar para desestabilizar o sistema. Para compensar esta variação, pode-se determinar ganhos de controlador adequados para modelos linearizados em vários pontos de equilíbrio de projeto sobre um conjunto de operação. Estes ganhos projetados podem ser programados em computadores para o desempenho satisfatório do controlador sobre o ambiente inteiro. Para que a programação de ganhos funcione, é essencial que os ganhos do controlador em cada ponto de equilíbrio de projeto garantam estabilidade para condições de operação reais próximas desse ponto de equilíbrio. Assim, é importante projetar controladores que tenham estabilidade e robustez, que é a capacidade de fornecer estabilidade apesar de erros de modelagem devido à dinâmica não modulada para altas frequências e variações de parâmetros da planta.

Perturbações e desempenho robusto

Muitas vezes, é importante ter em conta as perturbações e também o ruído de medição do sensor. Perturbações essas que podem causar desempenhos insatisfatórios em um sistema que foi projetado sem levá-los em conta. Assim, é importante projetar controladores que tenham desempenho robusto, que é a capacidade de garantir um rendimento

aceitável (em termos, por exemplo, de percentual de *overshoot*, tempo de assentamento, etc.) mesmo que o sistema possa estar sujeito a distúrbios.

Projeto robusto clássico

No controle clássico, a robustez pode ser projetada no sistema desde o início fornecendo ganho suficiente e margem de fase para neutralizar os efeitos de modelagem ou distúrbios imprecisos. Em termos de diagrama de magnitude de *Bode*, o ganho de malha deve ser alto em baixas frequências para a robustez de desempenho, mas baixa em altas frequências, onde a dinâmica não modelada pode estar presente, para estabilidade robusta. O conceito de largura de banda é importante nesse sentido, assim como o conceito de função de sensibilidade.

As técnicas de projeto de controles clássicos são geralmente no domínio da frequência, e assim eles proporcionam uma abordagem conveniente para a concepção robusta para sistemas de entrada única / saída única (SISO). No entanto, as margens de ganho individuais, as margens de fase, e as sensibilidades de todas as funções de transferência SISO em um sistema multivariável ou malha fechada têm pouco a ver com sua robustez geral. Deste modo, tem havido problemas no entendimento das noções clássicas de desempenho robusto para sistemas várias-entradas / várias-saídas (MIMO).

Projeto robusto moderno

Técnicas de controle modernos fornecem uma maneira direta de projetar controladores em malha fechada para sistemas MIMO fechando todas as malhas simultaneamente. O desempenho é garantido em termos de minimizar um índice de desempenho quadrático que, com uma formulação de problema sensível, geralmente implica também uma estabilidade em malha fechada.

3.1 Sistemas multivariáveis MIMO

Considere um sistema **MIMO** que possui m – entradas e l – saídas. Então, o modelo básico da função de transferência é

$$y(s) = G(s)u(s) \quad (3.1)$$

onde y é um vetor $l \times 1$, u é um vetor $m \times 1$ e $G(s)$ uma matriz de função de transferência $l \times m$ [15].

Nesse tipo de sistema, uma mudança na primeira entrada u_1 afeta, geralmente, todas as saídas y_1, y_2, \dots, y_l , assim haverá interações entre as entradas e saídas. Uma não interação na planta resultaria se u_1 afeta somente y_1 , u_2 afeta somente y_2 , e assim por diante.

Seja um sistema MIMO N-dimensional controlável e observável quadrático (que tem o mesmo número de entradas e saídas) como mostrado na Figura 3.4 [18]. Onde $\varphi(s)$, $f(s)$ e $\varepsilon(s)$ representam as transformadas de *Laplace* dos sinais de entrada, saída e sinal de erro $\varphi(s)$, $f(s)$ e $\varepsilon(s)$ N-dimensional, respectivamente (considerando como elementos de algum espaço complexo N-dimensional); $W(s) = w_{kr}(s)$ denota a matriz de função de transferência quadrática do sistema de malha aberta de ordem $\times N$ (matriz de transferência em malha aberta) com entradas $w_{kr}(s) (k, r = 1, 2, \dots, N)$ que são funções escalares racionais própria em variáveis complexas s . Os elementos $w_{kk}(s)$ na diagonal principal de $W(s)$ são as funções de transferência dos canais separados, e os elementos não-diagonalizáveis $w_{kr}(s) (k \neq r)$ são as funções de transferência das ligações cruzadas do r-ésimo para k-ésimo canal.

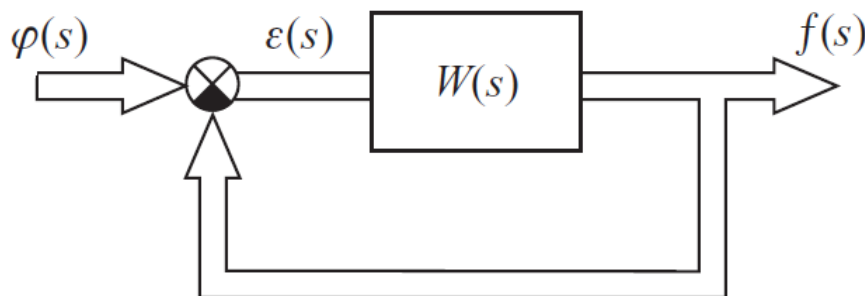


Figura 3.4: Diagrama bloco de um sistema realimentado MIMO do tipo linear

Fonte: [18]

Os vetores de saída $f(s)$ e erro $\varepsilon(s)$, onde

$$\varepsilon(s) = \varphi(s) - f(s) \quad (3.2)$$

estão relacionados com o vetor de entrada $\varphi(s)$ pelas seguintes equações de operação:

$$f(s) = \Phi(s)\varphi(s), \quad \varepsilon(s) = \Phi_\varepsilon(s)\varphi(s), \quad (3.3)$$

onde

$$\Phi(s) = [I + W(s)]^{-1} W(s) = W(s) [I + W(s)]^{-1} \quad (3.4)$$

$$\text{e } \Phi_\varepsilon(s) = [I + W(s)]^{-1} \quad (3.5)$$

são as matrizes da função de transferência do sistema **MIMO** de malha fechada (ainda, brevemente, referidas como matrizes de transferência em malha fechada) em relação aos sinais de saída e de erro, e I é a matriz unitária. As matrizes de transferência $\varepsilon(s)$ e $\varphi(s)$ são usualmente chamadas de *matriz de função de sensibilidade e matriz de função de sensibilidade complementar*¹.

Por simples cálculo, é fácil verificar que $\Phi_\varepsilon(s)$ e $\Phi(s)$ satisfazem a relação:

$$\Phi(s) + \Phi_\varepsilon(s) = I. \quad (3.6)$$

Portanto, conclui-se então que é impossível trazer para zero o erro do sistema se o sinal de entrada for uma soma (mistura) de um sinal de referência e perturbações, neste último pode ser, por exemplo, a medida ou outros ruídos. De fato, se o sistema rastreia (busca) idealmente o sinal de referência de entrada, isto é, se a matriz $\Phi_\varepsilon(s)$ é identicamente igual à matriz zero, então, devido ao princípio de superposição [5, 19], esse sistema também se reproduz idealmente na saída do ruído de entrada (desde que, $\Phi_\varepsilon(s) = 0$, então a matriz $\Phi(s)$ na equação (3.6) é igual à matriz unitária I). Um certo trade-off (compromisso) só pode ser conseguido desde que o sinal de referência de entrada e o ruído de medição não tenham sobreposição (pelo menos, parcialmente) nas faixas de frequência.

¹Os termos função de sensibilidade e função de sensibilidade complementar foram introduzidos por Bode (1945).

3.2 Análise multivariável MIMO da resposta em frequência

A função de transferência $G(s)$, conforme apresentada na Figura 3.4, é uma função da variável s de *Laplace* e pode ser usada para representar um sistema dinâmico [15]. Contudo, se $s = s_0$ então pode-se dizer que $G(s_0)$ é simplesmente uma matriz complexa, que pode ser analisada usando ferramentas padrões da álgebra matricial. Em particular, a escolha $s_0 = j\omega$ é de interesse desde que $G(j\omega)$ represente a resposta a um sinal senoidal de frequência ω .

Um problema que surge imediatamente para os sistemas **MIMO** é o de entender o gráfico de magnitude de Bode para o sistema **SISO**. Não há interesse em fazer várias plotagens individuais de frequência no sistema **SISO** para várias combinações das entradas e saídas no sistema **MIMO** e examinar as margens de ganho e fase. Essas abordagens têm sido testadas e nem sempre proporcionam muita percepção do verdadeiro comportamento do sistema **MIMO**. Isto é devido ao acoplamento que geralmente existe entre todas as entradas e todas as saídas de um sistema **MIMO**.

Para sistemas multivariáveis **MIMO**, o desempenho robusto é avaliado por meio do *Valor Singular* mínimo de ganho de malha. Devendo ser *grande* para baixas frequências, sendo que essas estão suscetíveis a perturbações, enquanto para valor singular máximo do ganho de malha deve ser *pequeno* em altas frequências, nestas apresentam imprecisões na modelagem. Para que se garanta a estabilidade robusta, apesar das variações dos parâmetros no modelo linearizado, o valor singular máximo deve estar abaixo de um limite superior [17].

3.3 Obtendo a resposta de frequência de $G(s)$

O domínio da frequência é ideal para estudar direções em sistemas multivariáveis em qualquer frequência dada [15]. Considere o sistema $G(s)$ na Figura 3.5 com entrada $d(s)$ e saída $y(s)$:

$$y(s) = G(s)d(s) \quad (3.7)$$



Figura 3.5: Sistema $G(s)$ com entrada d e saída y

Levando em consideração a resposta senoidal para sistemas escalares e também a generalização desse resultado para sistemas multivariáveis, cujos elementos, são dados por $g_{ij}(j\omega)$ da matriz $G(j\omega)$. Onde

- $g_{ij}(j\omega)$ representa a resposta senoidal da entrada j para a saída i . Para ser mais específico, aplicar ao canal de entrada j um sinal senoidal escalar dado por

$$d_j(t) = d_{j0}\text{sen}(\omega t + \alpha_j). \quad (3.8)$$

Este sinal de entrada é contínuo, isto é, quando aplicado para $t = \infty$. Em seguida, o sinal saída contínuo correspondente no canal i é também uma senoide com a mesma frequência

$$y_i(t) = y_{i0}\text{sen}(\omega t + \beta_i) \quad (3.9)$$

onde a amplificação (ganho) e fase podem ser obtidos a partir do número complexo $g_{ij}(j\omega)$ do seguinte modo

$$\frac{y_{i0}}{d_{j0}} = |g_{ij}(j\omega)|, \quad \beta_i - \alpha_j = \angle g_{ij}(j\omega) \quad (3.10)$$

Na notação fasorial a representação de forma compacta do tempo de resposta senoidal é descrita por

$$y_i(\omega) = g_{ij}(j\omega)d_j(\omega) \quad (3.11)$$

onde

$$d_j(\omega) = d_{j0}e^{j\alpha}, \quad y_i(\omega) = y_{i0}e^{j\beta}, \quad (3.12)$$

neste caso, o uso de ω (não $j\omega$) como o argumento de $d_j(\omega)$ e $y_i(\omega)$ implica que estes são números complexos representando cada frequência ω a magnitude e fase do sinal senoidal em (3.8) e (3.9).

A resposta geral aos sinais de entrada contínuo da mesma frequência em vários canais de entrada, pelo princípio da superposição para sistemas lineares, é igual a soma das respostas individuais e têm-se da equação (3.11)

$$y_i(\omega) = g_{i1}(j\omega)d_1(\omega) + g_{i2}(j\omega)d_2 + \cdots = \sum_j g_{ij}(j\omega)d_j(\omega) \quad (3.13)$$

ou na forma matricial

$$y(\omega) = G(j\omega)d(\omega) \quad (3.14)$$

onde

$$d(\omega) = \begin{bmatrix} d_1(\omega) \\ d_2(\omega) \\ \vdots \\ d_j(\omega) \\ \vdots \\ d_m(\omega) \end{bmatrix} \quad e \quad y(\omega) = \begin{bmatrix} y_1(\omega) \\ y_2(\omega) \\ \vdots \\ y_i(\omega) \\ \vdots \\ y_l(\omega) \end{bmatrix} \quad (3.15)$$

que representam os vetores dos sinais senoidais de entrada e saída.

3.4 Direções em sistema multivariáveis

Considere o sistema multivariável ilustrado na Figura 3.5 com entrada $d(s)$ e saída $y(s)$. Para o sistema **SISO** $y = Gd$ o ganho em uma dada frequência pode ser calculado simplesmente por [15]

$$\frac{|y(\omega)|}{|d(\omega)|} = \frac{|G(\omega)d(\omega)|}{|d(\omega)|} = \frac{y_0}{d_0} = |G(j\omega)|$$

O ganho depende da frequência ω , desde que o sistema seja linear, e independente da magnitude de entrada $|d(\omega)|$. Como os sinais de entrada e saída são vetores, torna-se necessário somar suas magnitudes utilizando uma soma de normas. No sistema **MIMO** o principal problema é a não existência de um ganho único devido as diferentes direções nesse tipo de sistema.

Utilizando-se a norma Euclidiana como medida da magnitude desses vetores, então para dada frequência ω , a magnitude do sinal do vetor de entrada é dado por

$$\|d(\omega)\| = \sqrt{\sum_j |d_j(\omega)|^2} = \sqrt{d_{10}^2 + d_{20}^2 + \dots} \quad (3.16)$$

e a magnitude do sinal do vetor de saída é dado por

$$\|y(\omega)\| = \sqrt{\sum_l |y_l(\omega)|^2} = \sqrt{y_{10}^2 + y_{20}^2 + \dots} \quad (3.17)$$

O ganho desse sistema $G(s)$ para um sinal de entrada particular $d(\omega)$ é dado pela razão

$$\frac{\|y(\omega)\|}{\|d(\omega)\|} = \frac{\|G(j\omega)d(\omega)\|}{\|d(\omega)\|} = \frac{\sqrt{y_{10}^2 + y_{20}^2 + \dots}}{\sqrt{d_{10}^2 + d_{20}^2 + \dots}} \quad (3.18)$$

este o ganho depende da frequência ω e independe da magnitude do sinal de entrada $\|d(\omega)\|$. No caso para um sistema multivariável **MIMO**, existem graus de liberdade adicionais e o ganho depende também da direção do sinal de entrada d . Portanto obtêm-se ganho com magnitudes diferentes quando se modifica as componentes de d , mesmo que $\|d(\omega)\|$ seja constante.

O valor máximo do ganho em (3.18) à medida que a direção de entrada é variada é o valor *singular máximo* de G ,

$$\bar{\sigma}(G) = \max_{d \neq 0} \frac{\|Gd\|}{\|d\|} = \max_{\|d\|=1} \|Gd\|$$

enquanto que o valor mínimo do ganho é o valor *singular mínimo* de G ,

$$\underline{\sigma}(G) = \min_{d \neq 0} \frac{\|Gd\|}{\|d\|} = \min_{\|d\|=1} \|Gd\|.$$

No controle moderno estes valores singulares máximos e mínimos são muito usados no estudo da robustez e desempenho de sistemas multivariáveis **MIMO** no domínio da frequência.

3.5 Sensibilidade e cosensibilidade

Seja o sistema de realimentação padrão no domínio da frequência mostrado na Figura 3.6 [17]. A planta é $G(s)$ e $K(s)$ é o compensador retroalimentado, podendo ser projetada utilizando técnicas do controle moderno. A saída da planta é $z(t) \in \mathbb{R}^q$, o controle de entrada da planta é $u(t) \in \mathbb{R}^m$ e a entrada de referência é $r(t) \in \mathbb{R}^q$.

Pode se observar que rastreamento perfeito pode não ser alcançado a menos que o número m das entradas de controle $u(t)$ seja maior ou igual ao número q das saídas de desempenho $z(t)$ [20, 21]. Portanto, assume-se que $m = q$ para que a planta $G(s)$ e o compensador $k(s)$ sejam quadráticos. Isso é apenas uma consequência do projeto prático, e não uma restrição sobre os tipos de plantas que podem ser consideradas.

Observa-se que nessa Figura 3.6 foram adicionados alguns itens para caracterizar as incertezas. O sinal $d(t)$ representa uma perturbação que atua direto sobre o sistema, típico no que aparece no controle clássico. Isso poderia representar, por exemplo, rajadas de vento. O ruído ou erros de medição do sensor são representados por $n(t)$. Ambos os sinais são geralmente vetores de dimensão q . Tipicamente, as perturbações ocorrem em baixas frequências, abaixo de algum ω_d , enquanto o ruído de medida $n(t)$ tem seu efeito predominante em altas frequências, acima de algum valor ω_n .

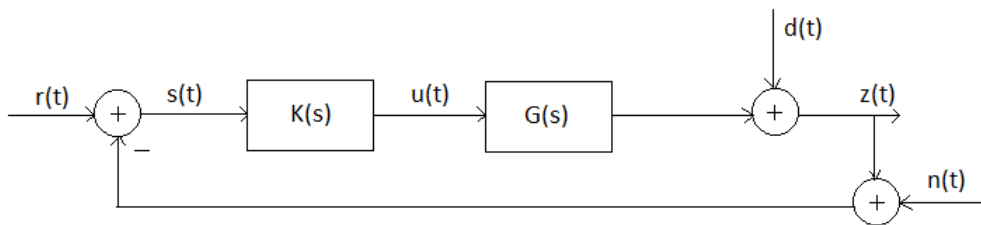


Figura 3.6: Configuração de malha fechada padrão

Fonte: Adaptado de [17]

O erro de rastreamento é dado por:

$$e(t) = r(t) - z(t). \quad (3.19)$$

Devido a presença de ruído na medida, o sinal $s(t)$ pode ser representado pela equação:

$$s(t) = r(t) - z(t) - n(t) = e(t) - n(t). \quad (3.20)$$

Na saída da planta:

$$z(t) = G(s)K(s)s(t) + d(t). \quad (3.21)$$

Analisando o sistema no domínio de frequência para verificação dos efeitos das incertezas sobre o desempenho do sistema e aplicando a transformada de *Laplace* nas equações (3.19), (3.20) e (3.21) têm-se:

$$E(s) = R(s) - Z(s) \quad (3.22)$$

$$S(s) = R(s) - Z(s) - N(s) \quad (3.23)$$

$$Z(s) = G(s)K(s)S(s) + D(s). \quad (3.24)$$

Substituindo a equação (3.23) na equação (3.24) têm-se:

$$Z(s) = G(s)K(s) [R(s) - Z(s) - N(s)] + D(s) \quad (3.25)$$

$$Z(s) = G(s)K(s)R(s) - G(s)K(s)Z(s) - G(s)K(s)N(s) + D(s)$$

$$Z(s) + G(s)K(s)Z(s) = G(s)K(s)R(s) - G(s)K(s)N(s) + D(s). \quad (3.26)$$

Por simplicidade, denota-se $G(s)K(s)$ por GK , e substituindo na equação (3.26)(4.35) têm-se:

$$Z(s) + GKZ(s) = GK R(s) - GKN(s) + D(s)$$

$$Z(s) [I + GK] = GK [R(s) - N(s)] + D(s)$$

$$Z(s) = [I + GK]^{-1} GK [R(s) - N(s)] + [I + GK]^{-1} D(s). \quad (3.27)$$

Substituindo a equação (3.27) na equação (3.22) têm-se:

$$E(s) = R(s) - [(I + GK)^{-1} GK (R(s) - N(s)) + (I + GK)^{-1} D(s)]$$

$$E(s) = R(s) - (I + GK)^{-1} GK R(s) + (I + GK)^{-1} GKN(s) - (I + GK)^{-1} GK D(s).$$

Logo o erro $E(s)$ em função das entradas $R(s)$, $D(s)$ e $N(s)$ pode ser representado pela seguinte equação:

$$E(s) = [I - (I + GK)^{-1} GK] R(s) + (I + GK)^{-1} GKN(s) - (I + GK)^{-1} GK D(s). \quad (3.28)$$

De acordo com o lema da inversão de matrizes [17] a equação (3.28) é dada por:

$$E(s) = (I + GK)^{-1} GK (R(s) - D(s)) + (I + GK)^{-1} GKN(s). \quad (3.29)$$

Sendo GK quadrada e inversível, então pode-se escrever a identidade como:

$$\begin{aligned} (I + GK)^{-1} GK &= [(GK)^{-1} (I + GK)^{-1}]^{-1} = [(GK)^{-1} + I]^{-1} \\ &= [(GK)^{-1} (I + GK)]^{-1} = GK [(I + GK)]^{-1}. \end{aligned} \quad (3.30)$$

Após realizar as devidas substituições, as equações que representam respectivamente a saída $Z(s)$ e o erro $E(s)$ são dadas por:

$$Z(s) = GK (I + GK)^{-1} (R(s) - N(s)) + GK (I + GK)^{-1} D(s) \quad (3.31)$$

$$E(s) = (I + GK)^{-1} (R(s) - D(s)) + GK (I + GK)^{-1} N(s). \quad (3.32)$$

Portanto, para o sistema da Figura 3.6, o ganho de malha é dado por $G(s)K(s)$. As equações que representam a função **sensibilidade** e **sensibilidade complementar** são respectivamente dadas por:

$$S(s) = (I + GK)^{-1} \quad (3.33)$$

$$T(s) = GK (I + GK)^{-1} = (I + GK)^{-1} GK \quad (3.34)$$

desde que

$$S(s) + T(s) = (I + GK) (I + GK)^{-1}. \quad (3.35)$$

Note-se que o inverso da função **sensibilidade** é dada por:

$$L(s) = I + GK. \quad (3.36)$$

A saída e o erro do sistema podem agora ser expresso em termos da função **sensibilidade** e **sensibilidade complementar** representadas respectivamente pelas equações:

$$Z(s) = T(s) (R(s) - N(s)) + S(s)D(s) \quad (3.37)$$

$$E(s) = S(s) (R(s) - D(s)) + T(s)N(s). \quad (3.38)$$

Essas expressões aumentam as noções clássicas de ganho em malha e diferença de retorno à sensibilidade em sistemas multivariáveis. As mesmas são geralmente matrizes de função de transferência quadráticas de dimensão $m \times m$.

Para garantir pequenos erros de rastreamento, deve-se ter $S(j\omega)$ pequena nas frequências ω onde a entrada de referência $r(t)$ e a perturbação $d(t)$ são grandes. Produzindo assim, uma boa rejeição de perturbação. Por outro lado, para uma rejeição satisfatória do ruído do sensor, deve-se ter $T(j\omega)$ pequena nas frequências ω onde $n(t)$ é grande. Infelizmente $S(j\omega)$ e $T(j\omega)$ não podem ser simultaneamente pequenos em qualquer frequência ω .

3.6 Decomposição do valor singular

A decomposição do valor singular é dado pela Definição 3.6.1. Entretanto o interesse maior é na sua interpretação física quando é aplicado à resposta frequência de um sistema **MIMO** $A(s)$ com m entradas e l saídas [15].

Definição 3.6.1. Qualquer matriz A complexa $l \times m$ pode ser composta em uma decomposição do valor singular

$$A = U\Sigma V^H \quad V^H = V^{-1}(\text{transposta conjugada complexa de } V)$$

onde a matriz U $l \times l$ e a matriz V $m \times m$ são matrizes unitárias e a matriz Σ $l \times m$ contém uma matriz diagonal Σ_1 com valores singulares reais, não negativos, σ_i , dispostos em ordem decrescente como em:

$$\Sigma = \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix}; \quad l \geq m \quad (3.39)$$

$$\text{ou } \Sigma = \begin{bmatrix} \Sigma_1 & 0 \end{bmatrix}; \quad l \leq m \quad (3.40)$$

onde,

$$\Sigma_1 \text{diag} \{ \sigma_1, \sigma_2, \dots, \sigma_k \}; \quad k = \min \{ l, m \} \quad (3.41)$$

$$e \quad \bar{\sigma} \triangleq \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k \triangleq \underline{\sigma}. \quad (3.42)$$

As matrizes unitárias U e V formam bases ortonormais de um espaço coluna (saída) e linha (entrada) de A . Aqui os vetores colunas de V denotado v_i chamado à direita ou vetores singulares de entrada e os vetores colunas de U , denotado u_i , chamado à esquerda ou valores de saída.

Note que esta decomposição é não única desde que $A = U'\Sigma V'^H$, onde $U' = VS^{-1}$ e $S = \text{diag} \{ e^{j\theta_i} \}$ θ_i é um número real qualquer, também é um *SVD* de A . Portanto, os valores singulares θ_i são únicos.

Definição 3.6.2. Matriz unitária: Uma matriz complexas U é unitária se

$$U^H = U^{-1}. \quad (3.43)$$

Todos os autovalores da matriz unitária tem valor absoluto igual a 1 e todos os seus valores singulares são, portanto, igual a 1.

Os valores singulares são raízes quadradas dos $k = \min(l, m)$ que são os maiores autovalores tanto AA^H e $A^H A$ têm-se:

$$\sigma_i(A) = \sqrt{\lambda_i(A^H A)} = \sqrt{\lambda_i(AA^H)}. \quad (3.44)$$

Além disso, as colunas de U e V são autovetores unitários de AA^H e $A^H A$, respectivamente. Calculando (3.44) escreve-se:

$$AA^H = (U\Sigma V^H) (U\Sigma V^H)^H = (U\Sigma V^H) (V\Sigma^H U^H) = U\Sigma\Sigma^H U^H \quad (3.45)$$

e equivalentemente desde que U seja unitário e satisfaça $U^H = U^{-1}$.

$$(AA^H)^U = U\Sigma\Sigma^H, \quad (3.46)$$

U é uma matriz de autovetores de AA^H e $\{\sigma_i^2\}$ são seus autovalores. Similarmente, V é uma matriz de autovetores de $A^H A$.

Portanto, no projeto de sistemas modernos multivariáveis, a decomposição dos valores singulares é de suma importância e apresenta algumas vantagens sobre a decomposição de autovalores para analisar ganhos e direcionalidades dessas plantas [15]. Algumas dessas vantagens são:

1. Os valores singulares dão melhores informações sobre os ganhos da planta;
2. As direções da planta obtidas a partir da Decomposição do Valor Singular (*SVD*) são ortogonais;
3. A *SVD* também se aplica diretamente à planta não quadrática.

3.7 Valores singulares para desempenho

O valor máximo singular também é muito útil em termos de **desempenho** e **robustez** no domínio da frequência [15]. O ponto importante é que em **baixas frequências** é requerido que os valores singulares mínimos sejam grandes, enquanto que em **altas frequências** os valores singulares máximas sejam pequenas [17].

Para um sistema de controle realimentado de um grau de liberdade, a resposta de frequência em malha fechada, r , para o erro de controle $e = y - r$ é $e = -Sr$. Para o sistema **SISO** $S(j\omega)$ é avaliado como uma função da frequência que fornece informações úteis sobre a efetividade do controle de realimentação, porque para senoides $|e(\omega)|/|r(\omega)| = |S(j\omega)|$. Para sistemas **MIMO**, uma generalização útil, resulta se considerar a razão $\|e(\omega)\|_2/\|r(\omega)\|_2$ onde $\|\cdot\|_2$ é um vetor de *norma-2*. Esta razão depende da direção de $r(\omega)$ e

$$\underline{\sigma}(S(j\omega)) \leq \frac{\|e(\omega)\|_2}{\|r(\omega)\|_2} \leq \bar{\sigma}(S(j\omega)). \quad (3.47)$$

Em termos de desempenho, é razoável exigir que a razão $\|e(\omega)\|_2/\|r(\omega)\|_2$ permaneça pequena para qualquer direção de $r(\omega)$, incluindo “pior caso” da direção que dá a razão como $\bar{\sigma}(S(j\omega))$. Deixar em cada frequência $\frac{1}{|\omega_p(j\omega)|}$ representar a magnitude máxima, permitida de $\|e\|_2/\|r\|_2$. Isso resulta no seguinte requisito de desempenho:

$$\bar{\sigma}(S(j\omega)) \leq 1/|\omega_p(j\omega)|, \quad \forall \omega \Leftrightarrow \bar{\sigma}(\omega_p S) \leq 1, \quad \forall \omega \Leftrightarrow \|\omega_p S\|_\infty \leq 1 \quad (3.48)$$

onde a norma H_∞ é definida como um pico de máximo valor singular de frequência de resposta

$$\|M(s)\|_\infty \triangleq \max \bar{\sigma}(M(j\omega)). \quad (3.49)$$

Tipicamente, eles são pequenos para baixas frequências e a realimentação é eficaz, e se aproximam de 1 em altas frequências porque qualquer sistema real é estritamente adequado. O valor máximo singular, $\sigma(S(j\omega))$, tem geralmente um pico maior do que 1 em frequência cruzada. Este pico é indesejável, mas é inevitável para sistemas reais.

4 SISTEMAS INTELIGENTES DE CONTROLE

A Inteligência Artificial (IA) nasceu tentando “aprender” como os seres humanos pensam e resolvem problemas, explicitando metodologias e introduzindo-as na programação de computadores [10]. Logo, a ideia era a construção de sistemas que fossem “inteligentes” e capazes de “pensar”.

Simplificadamente, afirma-se que a IA procura utilizar computadores para simular o processo do pensamento humano durante a realização de determinados problemas de forma “inteligente” que atualmente, são chamados de sistemas “inteligentes”. Sendo assim, qualquer aplicação computacional que execute tarefas, consideradas pelos seres humanos como “inteligente” pode ser baseado em Inteligência Artificial. Logo, tem-se como definição a IA como a parte da ciência da computação envolvida no projeto de sistemas que exibem características que associamos com inteligência no comportamento humano [9, 10]. Ou seja, execute um processo de *raciocínio*, *decisão* e *aprendizagem*.

A IA apresenta diversas técnicas utilizadas para a obtenção de diversos tipos de sistemas “inteligentes” que são:

- Sistemas especialistas - inspirado na inferência humana;
- Sistemas baseados na lógica “*fuzzi*” - Processo Linguístico;
- Sistemas baseados em redes “neurais” - inspirados por neurônios biológicos;
- Sistemas baseados em Algoritmos Genéticos (AG) - inspirados na evolução biológica.

Sistemas especialistas são programas computacionais destinados a solucionar problemas em um campo especializado do conhecimento humano.

Lógica nebulosa (*Fuzzy Logic*) tem por objetivo modelar o modo aproximado de raciocínio humano, através de sistemas computacionais capazes de tomar decisões racionais em um ambiente incerto e impreciso.

Redes Neurais Artificiais são modelos computacionais não lineares, inspirados na estrutura e operação do cérebro humano, procurando assim, reproduzir características humanas, tais como: *aprendizado*, *associação*, *generalização* e *abstração*.

Algoritmos Genéticos (GAS: *Genetic Algorithms*) são algoritmos matemáticos inspirados nos mecanismos de evolução natural e recombinação genética.

4.1 Sistemas baseados em redes “neurais”

A utilização de redes de neurônios ou “neurais” é bem antiga. Sabe-se que na década de 40, **McCulloch** e **Pitts** em seu livro intitulado “*A logical calculus of the ideas immanent in nervous activity*” já se interessavam pelo problema de representar funções lógicas, como o “E” ou o “OU” lógico, à partir de unidades de decisão. Definiram então o “neurônio formal” que são resultados obtidos da neurobiologia.

A utilização das técnicas não simbólicas e baseadas em redes “neurais” ou sistemas conexionistas, ou seja, sistemas computacionais capazes de realizar tarefas intelectuais complexas, tais como: a resolução de problemas, o reconhecimento e classificação de padrões, os processos intuitivos e dedutivos. Logo, as redes neurais imitam a estrutura física do cérebro como um modelo base, ou seja, na maneira como o cérebro é organizado na sua arquitetura elementar e como essa rede é capaz de executar essas tarefas computacionais.

Os **neurônios**, ou células nervosas, são os elementos de base do sistema nervoso central. Sendo da ordem de 100 bilhões, eles, possuem características particulares em relação às outras células do corpo humano. Possuem então a capacidade de:

- receber sinais de outros neurônios vizinho;
- integrar os sinais vindos de vários outros neurônios de forma a tirar uma conclusão;
- gerar e conduzir um sinal elétrico dentro do mesmo;
- transmitir um sinal a um outro neurônio capaz de recebê-lo.

A constituição de um neurônio é dado em três partes: corpo celular, dendritos e axônio. Estas são apresentadas na Figura 4.1.

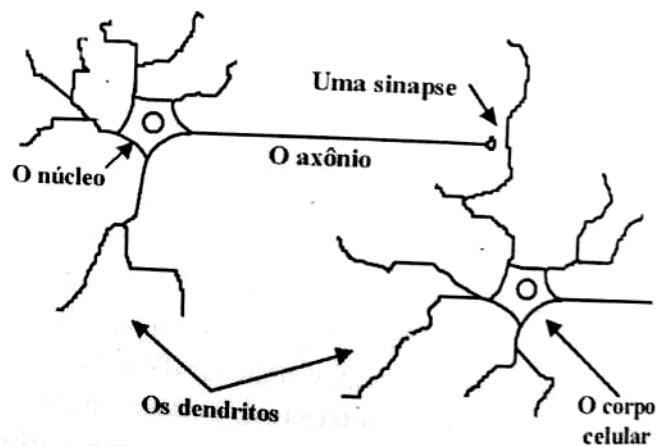


Figura 4.1: Um neurônio e o esquema da sinapse

- Corpo celular
 - contém o núcleo do neurônio sendo responsável pelas transformações bioquímicas necessárias à síntese das enzimas e das outras moléculas que garantem a vida do neurônio.
- Os dentritos
 - são finas extensões tubulares de alguns décimos de microns de diâmetro e de comprimento de algumas dezenas de microns. São os principais receptores do neurônio que captam sinais que lhe são destinados.
- O axônio
 - é uma fibra nervosa, que serve de meio de transporte para os sinais emitidos pelo neurônio.

A formação de um sistema nervoso se dá pela conexão dos neurônios formando um arranjo espacial completo. As conexões entre dois neurônios são feitas em lugares chamados **sinapse**. Essas tem um papel fundamental para a comunicação entre as células nervosas. Essa comunicação é efetuada através de substâncias químicas chamadas de “**neuro transmissores**”.

Os sinais nervosos são transmitidos eletricamente ou quimicamente. A transmissão elétrica prevalece no interior de um neurônio, enquanto que os processos químicos operam entre os diferentes neurônios. A transmissão elétrica é baseada em uma descarga elétrica que começa no corpo da célula e, em seguida, viaja para baixo do axônio para as várias conexões sinápticas [22].

Na Figura 4.2 é apresentado o esquema clássico de um neurônio que efetua a soma dos sinais nervosos transmitidos pelos seus dentritos. Caso a soma ultrapasse um determinado valor, o neurônio responde enviando um sinal que é propagado ao longo do seu axônio. Caso contrário, o mesmo fica na inatividade.

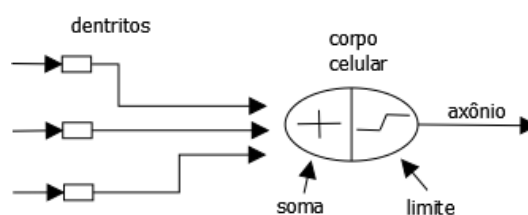


Figura 4.2: Modelo para o funcionamento do neurônio

Fonte: Adaptado de [10]

Como os neurônios processam as correntes elétricas que chegam, dos seus dendritos, e envia essa corrente elétrica resultante aos neurônios conectados ao seu axônio.

No estado de inatividade, o interior de um neurônio, o *protoplasma*, está carregado negativamente contra o líquido neural circuncidante. Tendo em um potencial de repouso no valor aproximado de -70 mV suportado pela ação da membrana celular, esta é impenetrável para os íons Na^+ , causando assim, uma deficiência de íons positivos no protoplasma [22]. Na figura 4.3 está a representação dessa estrutura.

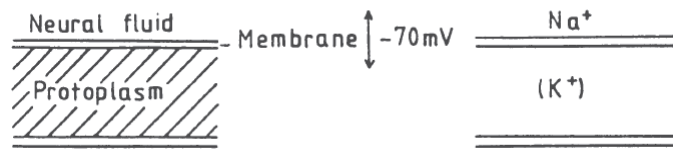


Figura 4.3: Estrutura de um axônio

Fonte: [22]

Em termos matemáticos, um modelo de rede neural é definido como **grafo** direcionado que possui as seguintes propriedades:

1. Variável de estado η_i estar associado com cada nó de i
2. Um peso real estimado ω_{ik} estar associado com cada ligação (ik) entre dois nós i e k
3. Uma direção real estimada ϑ_i estar associada com cada nó i
4. Uma função de transferência $f_i[\eta_k, \omega_{ik}, \vartheta_i, (k \neq i)]$ sendo definida para cada nó i , que determina o estado de nós como uma função de suas direções, dos pesos de suas ligações de entrada e dos estados dos nós conectados por essas ligações.

Na terminologia padrão, os nós são chamados de *neurônios*, as ligações são chamadas de *sinapses* e as direções que são conhecidas como o *limiar de ativação*. A função de transferência normalmente, assume a forma $f(\sum_k \omega_{ik} \eta_k - \vartheta_i)$, onde $f(x)$ é também uma função de passo descontínuo ou sua generalização suavemente crescente conhecida como **função sigmoide** [22]. A função sigmoide é normalmente a mais utilizada nos neurônios não lineares, sendo sua saída proporcional a soma ponderada das suas entradas. Essa função é a que mais se aproxima da **função ativação** de um neurônio real. Tem como principais características:

- Limitada;
- Monotonicamente crescente;
- Contínua e suave(tem derivada em todas os seus pontos).

Como exemplo desse tipo de função temos respectivamente a função logística e a função tangente hiperbólica.

-Função logística:

$$\begin{aligned} f(u) &= \frac{1}{1 + e^{(-au)}} \\ f'(u) &= \frac{ae^{(-au)}}{[1 + e^{(-au)}]^2} \end{aligned} \quad (4.1)$$

onde a parâmetro variável.

-Função tangente hiperbólica:

$$\begin{aligned} f(u) &= \tanh(u) \\ \tanh(u) &= \frac{1 - e^{(-2u)}}{1 + e^{(-2u)}}. \end{aligned} \quad (4.2)$$

Nas Figuras 4.4 e 4.5 são apresentadas os gráficos dessas funções.

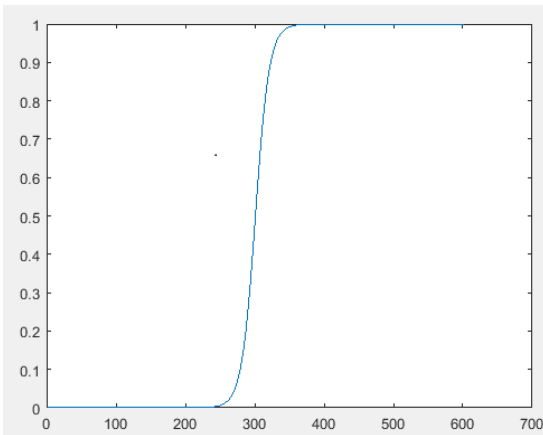


Figura 4.4: Função logística

Fonte: Autor

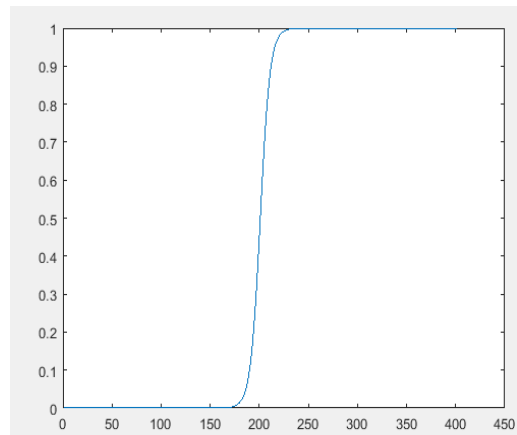


Figura 4.5: Função tangente hiperbólica

Fonte: Autor

4.1.1 Rede de neurônios artificial - multicamadas

A rede de neurônios multicamadas é um sistema artificial [23] composto por células elementares, neurônios, organizadas em camadas sucessivas que são conectadas entre elas. Na Figura 4.6 é mostrado esse tipo de rede.

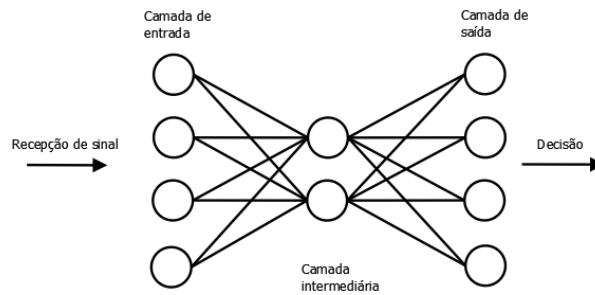


Figura 4.6: Exemplo de uma rede multicamadas

Fonte: Adaptado de [10]

- Camada de entrada - Corresponde a retina;
- Camada de saída - Função de decisão;
- Camada intermediária - Função de associação.

Dessa forma, as redes neurais, implementam uma interconexão massiva de células computacionais chamadas de neurônios artificiais ou unidades de processamento. Logo, uma rede neural é um processador paralelo distribuído de forma massiva que possui a capacidade natural de armazenar conhecimento por experiência adquirida e torná-lo disponível para uso [10].

4.1.2 Rede Neural - rede de neurônios artificial

Os Neurônios artificiais são organizados em camadas e cada uma delas são totalmente conectadas à camada seguinte e a anterior. O sinal que chega à camada de entrada se propaga, camada à camada, até a saída. Todos os neurônios de uma camada calculam suas saídas simultaneamente (paralelismo da rede neural multicamadas [24]). Logo, os parâmetros de caracterização de uma rede multicamadas são: número de camadas, número de neurônios por camada, escolha das conexões, tipo de funções de ativação dos neurônios e peso de cada uma das conexões.

Uma rede neural em malha aberta ou sem retroalimentação (estática) se caracteriza por não possuir uma malha fechada ou ciclo e, ela representa geralmente uma relação algébrica não linear entre suas entradas e saídas. A rede neural fechada ou com retroalimentação (dinâmica) possui pelo menos uma malha fechada ou ciclo. Na Figura 4.7 é mostrado um exemplo de rede com retroalimentação.

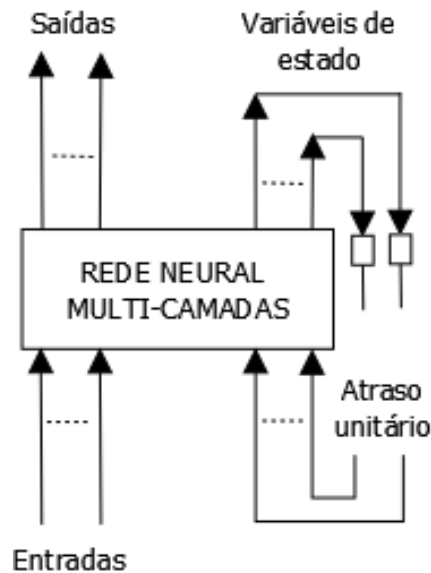


Figura 4.7: Exemplo de uma rede de neurônios com retroalimentação

Fonte: Adaptado de [10]

4.1.3 Principais arquiteturas de redes neurais artificiais

- *Perceptron*: Foi a primeira rede neural que obteve sucesso prático [16]. Inventado por Roseblant (em 1958). Esta rede estática consiste de um ou mais elementos de processamentos (neurônios artificiais), chamados de *Perceptron* mostrado na Figura 4.8

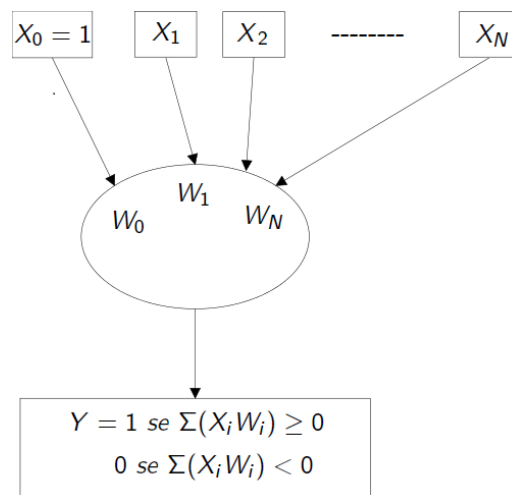


Figura 4.8: O *perceptron*

Fonte: Adaptado de [10]

A saída do perceptron será igual a 1 caso a somatória do produto de cada entrada pelo seu respectivo peso seja maior ou igual a zero. Caso contrário, a saída será igual a zero.

- **Perceptron multicamada:** É uma rede estática que consiste de uma camada de entrada, uma camada de saída e uma ou mais de camadas ocultas (intermediárias) conectadas em *feed-forward* (cada camada se conecta à próxima camada, sem caminho de volta), isto é, a saída de um neurônio não tem dependência dos valores anteriores, pois os seus sinais se propagam num único sentido e suas saídas dependem somente dos sinais que chegam dos outros neurônios (não há laços nesse sistema). Na figura 4.9 é mostrado um exemplo de uma arquitetura gráfica do perceptron multicamada.

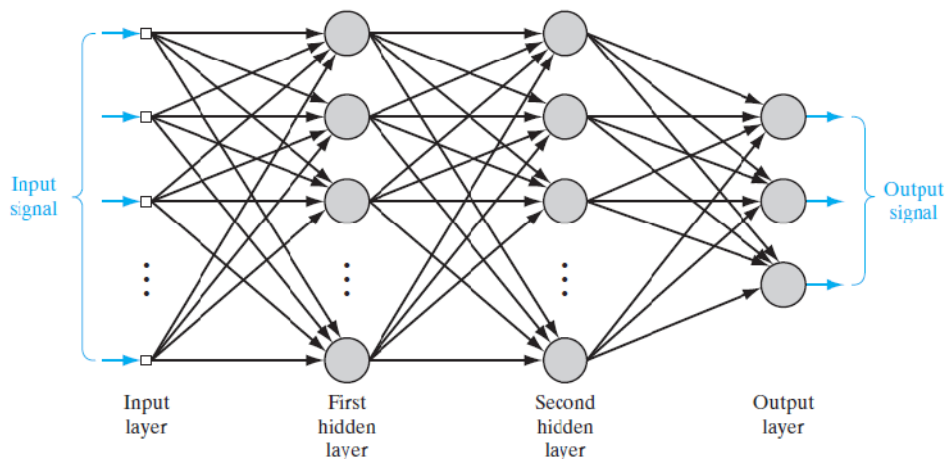


Figura 4.9: Arquitetura gráfica de um *perceptron* multicamada com duas camadas intermediárias

Fonte: [25]

- **Kohonen:** Este tipo de rede é composto de várias unidades de processamento em paralelo. Sendo que cada unidade possui um conjunto de pesos (W_I) e uma função que calcula a distância entre pesos e o vetor de entrada (X) [10]. A unidade de processamento cujos pesos forem mais próximos do vetor de entrada ganha a competição e tem a sua saída ativada ($Z_I = 1$), enquanto as saídas dos outros “neurônios” ficam desativadas ($Z_I = 0$). A Figura 4.10 ilustra um exemplo dessa rede.

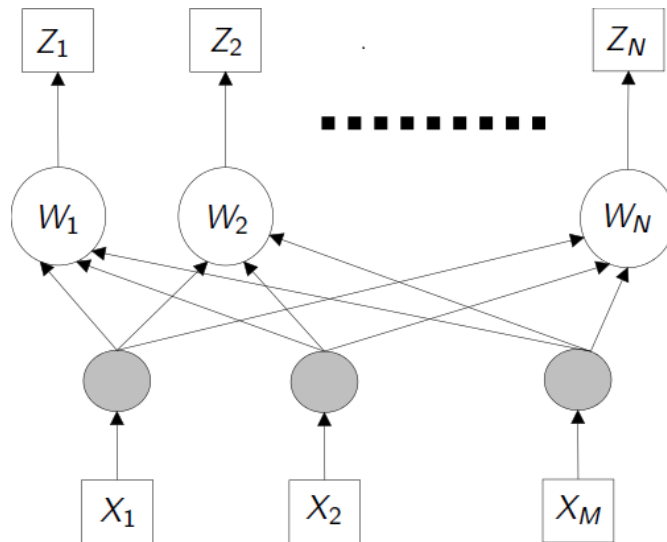


Figura 4.10: A rede de *Kohonen*

Fonte: Adaptado de [10]

Essa rede possui a lei de aprendizagem dada por:

$$W_I^{\text{NOVO}} = W_I^{\text{ANTIGO}} + \alpha \cdot (X - W_I^{\text{ANTIGO}}) \cdot Z_I .$$

Onde α controla a velocidade ou a taxa de aprendizagem possuindo valores entre 0 e 1. A Figura 4.11 mostra um esquema desta lei de aprendizagem [26].

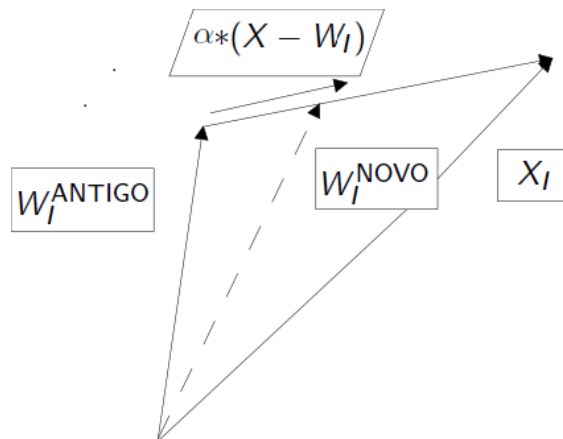


Figura 4.11: Lei de aprendizagem da rede de *Kohonen*

Fonte: Adaptado de [10]

- *Hebb*: Em sua teoria para aprendizagem ao nível celular considera que quando um neurônio emite um pulso em função de uma informação vinda de uma determinada entrada (sinapse), esta entrada “aumenta” a sua capacidade de produzir novos impulsos no futuro (como se ela aumentasse o seu peso). O mesmo postulou que o

aumento da conexão (e portanto da memória) ocorre ao nível da sinapse.

A lei de aprendizagem de *Hebb* ocorre em função de “eventos” (por “coincidência”) e através de recompensas ou punições ao nível celular. Essa lei pode ser usada em uma rede de associação linear mostrada na Figura 4.12 e sendo resumida pela equação,

$$W_I^{\text{NOVO}} = W_I^{\text{ANTIGO}} + Y_I \cdot X_I$$

onde (Y_I, X_I) são exemplos usados para a aprendizagem da rede e os pesos são inicializados com zeros.

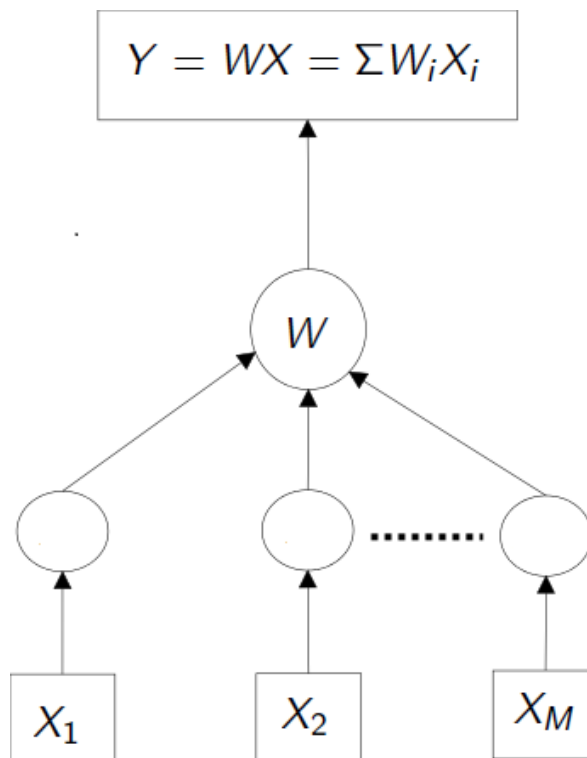


Figura 4.12: Rede de associação linear

Fonte: Adaptado de [10]

- **Rede de funções de base radial (RNFBR):** É uma rede neural de duas camadas em que os nós de saída formam uma combinação linear das funções de cada nó da camada intermediária [10]. A Figura 4.13 mostra a estrutura dessa rede.

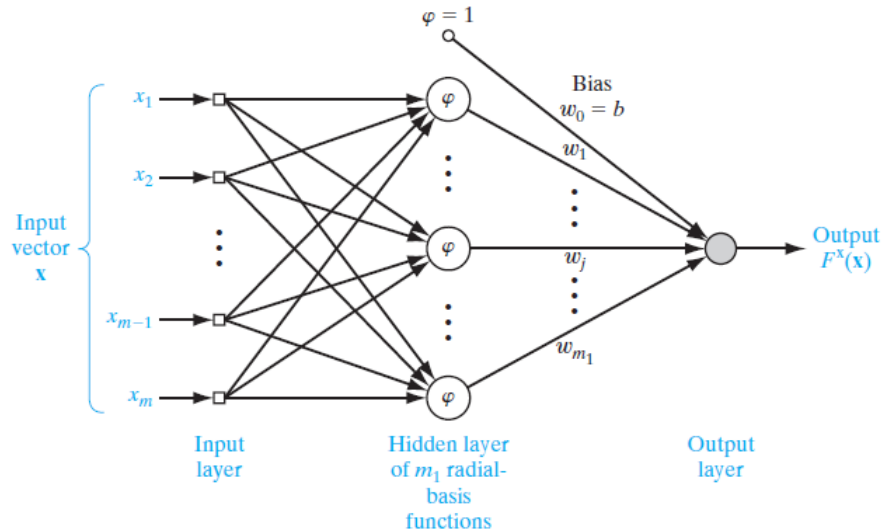


Figura 4.13: Rede de função de base radial de complexidade reduzida

Fonte: [25]

A função de base radial mais comum é a função Gaussiana na forma:

$$b_i(x) = \exp\left[-\frac{[(x - c_i)^T(x - c_i)]}{2\sigma_i^2}\right], \quad i = 1, 2, \dots, J.$$

Onde:

$b_i(x)$: saída do i -ésimo nó da camada intermediária;

x : vetor de entrada;

c_i : referência do i -ésimo nó da camada intermediária, isto é, o centro da função gaussiana para o nó i ;

σ_i^2 : é o parâmetro de normalização para o i -ésimo nó;

J : é o número de nós da camada intermediária.

Na Figura 4.14 é mostrada a estrutura dessa rede baseada nessa função gaussiana.

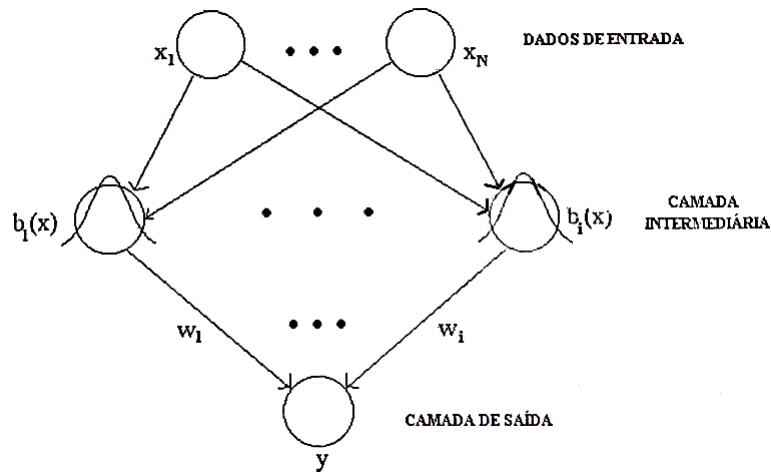


Figura 4.14: Estrutura de uma rede neural de base radial (RNFBR)

Fonte: [10]

A equação do nó de saída é dada por:

$$y = \sum_{i=1}^J w_i b_i(x)$$

onde:

y : saída de rede;

w_i : peso do i -ésimo nó;

$b_i(x)$: saída do i -ésimo nó da camada intermediária.

- **Redes neurais recorrentes (RNR)**: Nesse tipo de rede as saídas dos neurônios são realimentados à rede, resultando assim, em um sistema dinâmico. Um exemplo é a rede *Hopfield*, que no tempo discreto é representado como

$$x_{k+1} = \tanh(Wx_k)$$

onde $x_k \in \mathbb{R}^n$ é o vetor do estado, $W \in \mathbb{R}^{n \times n}$ é uma matriz de pesos sináptica.

A rede de *Hopfield* pode ser usada como uma memória endereçável por conteúdo ou computador analógico para resolver problemas de otimização do tipo combinatório [25].

4.2 Sistemas baseados em algoritmos genéticos (AG)

Os algoritmos genéticos são métodos de busca inspirados nos mecanismos de seleção natural (**Teoria da Evolução**) e da genética. Estes também são muito utilizados popularmente nas pesquisas de **Computação Evolutiva(CE)**. Eles combinam um mecanismo de valorização dos “*melhores*” indivíduos ou dos mais adaptados ao objetivo em questão, com uma estrutura para combinar e “*reproduzir*” aleatoriamente estes indivíduos, criando uma nova população. Baseado no princípio da evolução e seleção de **Charles Darwin** (1659), onde a primeira afirma que com o tempo um pequeno número de organismos semelhantes pode gerar ao longo das gerações muitos descendentes diferentes. Os mesmos podem se diferenciar de tal modo a se igualarem em novos grupos ou “*espécies*” que passam a se reproduzir entre si, fazendo com que as diferenças se perpetuem [27] e a segunda propõe que as plantas e os animais que existem hoje são o resultado de milhões de anos de adaptação às exigências do ambiente. Em algum momento, vários organismos diferentes podem coexistir e competir pelos mesmos recursos em um ecossistema. Os organismos mais capazes são os que adquirem recursos e procriam com êxito e cuja a descendência tende a ser numerosa no futuro e, os menos capazes, por qualquer razão, tendem a ter poucos ou nenhum descendente no futuro. Diz-se os primeiros são mais aptos do que os segundos, e as características distintivas que fizeram com que os primeiros se encaixem, são ditos ser selecionados, por sobre as características destes últimos. Concluindo que toda a população do ecossistema evolui para conter organismos que, em média, são mais aptos do que os de gerações anteriores da população, pois exibem mais dessas características que tendem promover a sobrevivência.

Na teoria da evolução, os dois principais fatores que regem o aparecimento das novas espécies são: o tamanho das populações e a frequência com que aparecem as diferenças entre os indivíduos [10].

- **Tamanho das populações:** os grupos mais numerosos se multiplicam mais rapidamente, aumentando dessa forma, a probabilidade de aparecimento de novos subgrupos, tendo em vista, características distintas dentre os grupos.
- **Frequência de aparecimento das diferenças entre os indivíduos:** quanto maior as frequências de diversificação dos indivíduos mais aptos eles se tornam, obtendo assim, novos “*habitats*” e suscetíveis a mudanças.

A seleção natural, segundo *Darwin*, representa um fator de regulação que mantém o sistema em harmonia, tendo como resultado final uma melhoria sempre crescente da capacidade de adaptação do ser, relativamente às suas condições ambientais [27].

4.2.1 Algoritmos de otimização

O objetivo de uma otimização, no caso do controle de um processo industrial, é melhorar o desempenho de um dado sistema em direção a um ponto ou região ótima de operação [10]. Portanto, existem pontos importantes definidos na otimização:

- método da busca de melhoria do desempenho;
- ponto ótimo.

As técnicas de computação evolutiva (CE) concentram-se nos princípios evolutivos, buscando assim, algoritmos que podem ser usados para buscar soluções ótimas para um determinado problema [21]. Nesses algoritmos de busca, uma série de possíveis soluções para um problema estão disponíveis e a tarefa é encontrar a melhor solução possível em um período de tempo fixo. Para um espaço de busca com apenas um pequeno número de soluções, todas as soluções podem ser encaminhadas em um período de tempo razoável e o ótimo encontrado. No entanto, essa busca exaustiva se torna impraticável à medida que o espaço de busca cresce em tamanho. Os algoritmos de pesquisas tradicionais, *aleatoriamente*, mostram (por exemplo, caminho habitual) ou *heuristicamente* mostram (por exemplo, a descida do gradiente), uma solução em um espaço de pesquisa por vez na esperança de encontrar a solução ótima. Logo, o aspecto que distingue um algoritmo de pesquisa *tradicional* de um algoritmo de pesquisa *evolucionário* é que esse último baseia-se na população. Tendo em vista que através da adaptação de gerações sucessivas de um grande número de indivíduos, o algoritmo evolutivo, realiza uma eficiente busca direcionada. O problema da otimização pode então ser enunciado:

- Encontrar um jogo de parâmetros $x \in \Omega$ que minimiza a função custo $f(x)$, respeitando as restrições: $\Omega = \{x : Q(x) \leq 0, R(x) = 0\}$.

Utiliza-se, neste caso, os algoritmos genéticos, pois sendo este um método, os mesmos ajudam a implementar essa estratégia de otimização.

Portanto, o princípio básico da otimização é a alocação eficiente de recursos limitados. Ela pode ser aplicada a qualquer disciplina científica ou de engenharia. Logo, o seu objetivo também é encontrar um algoritmo, que resolva uma determinada classe de

problemas. Sendo assim, não existe um método específico que resolva todos os problemas de otimização. Considere, por exemplo, uma função

$$f(x) : [x^1, x^u] \longrightarrow [0, 1] \quad (4.3)$$

Onde

$$f(x) = \begin{cases} 1, & \text{se } \|x - a\| < \epsilon, \epsilon > 0 \\ -1, & \text{em qualquer caso} \end{cases}$$

Para a função 4.3, $f(x)$ pode ser mantida decrescente ou fazendo o intervalo $[x^1, x^u]$ grande. Assim, uma tarefa difícil pode ser facilitada. Portanto, pode-se resolver problemas de otimização, combinando a criatividade humana e a capacidade de processamento bruto dos computadores [21].

Do ponto de vista teórico, classifica-se os problemas de otimização em duas classes: os de dimensões finitas e os de dimensões infinitas.

Para os de dimensões finitas, classifica-se os métodos de resolução: Métodos determinísticos e os métodos heurísticos. Conforme estar ilustrado na Figura 4.15.

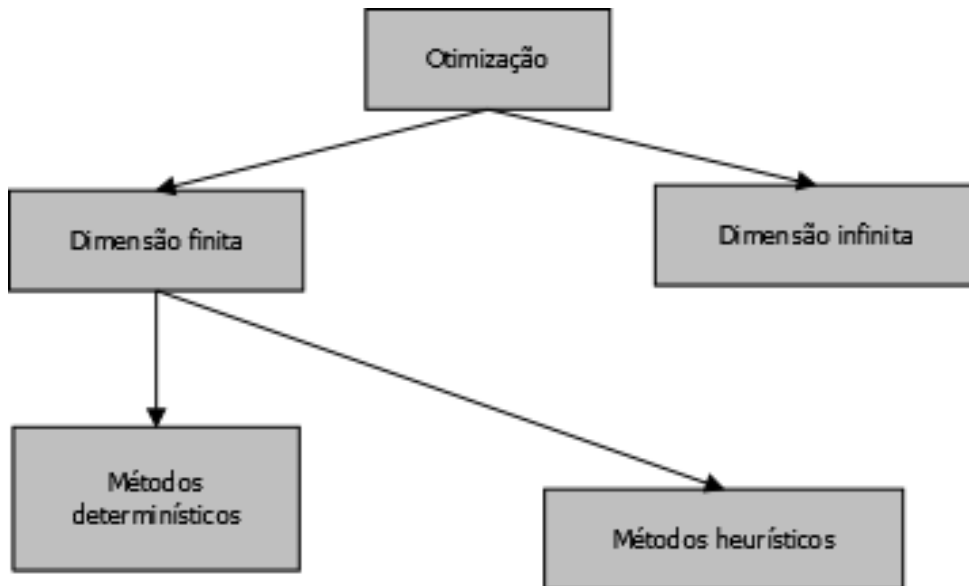


Figura 4.15: Classificação dos métodos de otimização

Fonte: Adaptado de [10]

Uma classificação dos algoritmos de busca ou otimização é dada como:

- **Métodos baseados em cálculos (determinísticos):**
 - Usam informações do gradiente da função a ser otimizada;
 - Tem o problema de serem locais, pois procuram a melhor solução em uma vizinhança, obtendo assim, apenas o ótimo local;
 - Desvantagens de necessitar do cálculo e da existência da derivada da função objetivo e que ela seja contínua, convexa e diferenciável. Devido a essas limitações reduzem a robustez e o espaço de problemas nas aplicações desses métodos.

- **Métodos enumerativos (heurístico):**
 - São métodos seletivos, pois privilegiam certas direções consideradas “locais”;
 - São rápidos, mas não dão garantia de cobrir todo espaço possível de soluções;
 - Discretizam o espaço de soluções e fazem uma busca em todos os pontos possíveis;
 - Se assemelha aos seres humanos ao resolver problemas pequenos;
 - Podem potencialmente obter o máximo e o mínimo global.

- **Métodos probabilísticos (heurístico):**
 - Podem obter o máximo e o mínimo global;
 - São equivalentes em termos de eficiência, aos enumerativos.

Portanto, os algoritmos genéticos são teórica e empiricamente reconhecidos como *robustos* em problemas de busca em espaços complexos [28, 29]. Eles foram inventados e desenvolvidos por *Jonh Holland* em 1975 e, publicado em seu livro “Adaptação em sistemas naturais artificiais”. Propôs inicialmente o AG como um método heurístico, baseado na “sobrevivência do mais forte”. O algoritmo genético foi descoberto como uma ferramenta útil para problemas de busca de otimização [28]. Possuem características importantes. Primeiro, é um algoritmo estocástico e randômico com uma função essencial tendo em vista que tanto na seleção quanto na reprodução é preciso a aleatoriedade. Segundo, os algoritmos genéticos sempre consideram uma população de soluções. Manter na memória mais do que uma única solução em cada iteração oferece muitas vantagens. Eles podem recombinar diferentes soluções para obter os melhores e, portanto, usar o benefício da seleção. Um algoritmo de base populacional também é muito acessível para a paralelização. Por causa da sua robustez que é essencial para o sucesso de qualquer tipo algoritmo.

4.2.2 Comparação dos algoritmos genéticos com outras técnicas de otimização

O princípio do algoritmo genético é simples: imitar a *genética* e a *seleção natural* por um programa de computador. Os parâmetros do problema são codificados mais naturalmente como uma estrutura linear de dados do tipo “DNA-semelhante”, um vetor ou uma *string* (sequência). As vezes, quando o problema é naturalmente de duas ou três dimensões, também é usada uma estrutura de matrizes [21].

Ao comparar os algoritmos genéticos com outros métodos de otimização nota-se que a função de desempenho pode ser aproximadamente algo que deva ser calculado por um computador ou até mesmo algo que não possa ser calculado [21]. Neste último caso, isto pode ser um julgamento humano que não pode ser determinado como um programa rígido, como no caso de um observador, no qual um ser humano seleciona no meio as alternativas geradas por um algoritmo genético. Assim, não há quaisquer restrições matemáticas definidas sobre as propriedades da função de desempenho. Podendo estas serem *discretas*, *multimodal* entre outras.

Os principais critérios usados para classificar os algoritmos de otimização são os seguintes: contínuos/discretos, limitado/não limitados e sequencial/paralelo. Há uma clara diferença entre problemas discretos e contínuos. Portanto, os contínuos às vezes são usados para a resolução de problemas inerentemente discretos e vice-versa. Os paralelos são geralmente usados para acelerar o processamento.

Logo, a diferença entre o *algoritmo genético* e as *técnicas convencionais* de otimização são:

- Os algoritmos genéticos operam com versões codificadas dos parâmetros do problema em vez de parâmetros próprios, isto é, funcionam com a codificação do conjunto de soluções e não com a própria solução;
- Boa parte das técnicas convencionais de otimização pesquisam a partir de um único ponto, mas o algoritmo genético opera sempre em toda uma população de pontos(*strings*);
- Os algoritmos genéticos usam a função de desempenho para analisar ao invés de derivadas. Podendo ser aplicado como resultado em qualquer tipo de problemas de

otimização contínua ou discreta;

- Os algoritmos genéticos usam operadores para transições probabilísticas enquanto os métodos convencionais, para otimização contínua, aplicam operadores para transição determinísticas, isto é, os AG's não usam regras determinísticas.

4.2.3 Vantagens e limitações do algoritmo genético

Dentre as vantagens dos algoritmos genéticos estão:

- Paralelismo;
- Responsabilidade;
- O espaço da solução é amplo;
- O cenário do desempenho é complexo;
- O problema tem função multi-objetivo;
- Controla bem as funções ruidosas;
- As descontinuidades presentes na superfície de resposta têm pouco efeito no desempenho global de otimização.

Dentre as limitações do algoritmo genético estão:

- O problema de identificar a função *fitness* (avaliação);
- Definir a apresentação para o problema;
- O problema de selecionar os vários parâmetros como o tamanho da população, a taxa de mutação, a taxa de cruzamento e o método de seleção mais vantajoso;
- Não é possível usar gradiente;
- Exige um grande número de resposta para as funções de avaliação (*fitness*);
- A sua configuração não é direta.

4.2.4 Aplicações dos algoritmos genéticos

Suas aplicações têm utilidades para problemas complexos, para problemas de aprendizagem de máquinas e na evolução de programas simples. São usados no processamento de sinais em desenvolvimento de imagens e também na música.

Algumas aplicações utilizando os algoritmos genéticos são:

- Sistemas dinâmicos não lineares - previsão e análise de dados;
- Planejamento da trajetória de um robô;

- Evolução de programas, como por exemplo, LISP (programação genética);
- Planejamento estratégico;
- Encontrar a forma das moléculas de proteínas;
- Função para criar imagens;
- Configuração de projeto de semicondutores, projetos de aeronaves, redes de comunicação;
- Aprendizagem de máquinas - desenvolvidos por redes neurais, tanto arquitetura quanto pesos, melhorando o algoritmo de classificação, sistemas classificadores;
- Processamento de sinal - projeto de filtros.

4.2.5 Algoritmo genético (AG) com parâmetros codificados binariamente

Darwin, a partir da sua teoria da evolução e da genética, analisava uma população de indivíduos que sofriam modificação ao longo das gerações. O algoritmo genético, também, tem como base essa análise formada de possíveis soluções do problema a ser resolvido. No início, essas soluções são geradas aleatoriamente e são representadas pelos cromossomos (indivíduos) nos quais são manipulados. O sistema avança na direção da melhor solução através da aplicação dos operadores “genéticos”, tais como *seleção*, *recombinação* e *mutação* [29]. A Figura 4.16 está representando uma população de indivíduos codificados em números binários, criados aleatoriamente.

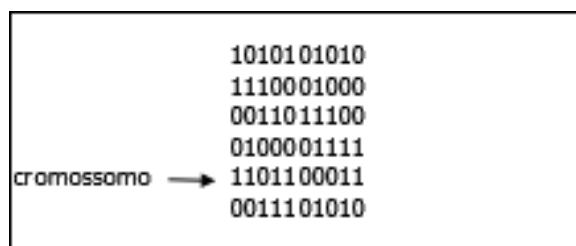


Figura 4.16: Exemplo de uma população do AG

Fonte: [10]

Cada linha dessa população representa um indivíduo (cromossomo), considerado como dois genes, codificados e representados por X_1 e X_2 de uma função objetivo qualquer, conforme destacado na Figura 4.16.

O comprimento desse cromossomo ([1101][1000011]) é a função da quantidade

de parâmetros vezes a quantidade de bits (representação do parâmetro). Esses bits de um parâmetro estão diretamente ligados com a precisão com que se deseja proceder à busca do *valor ótimo* dentro do domínio da variação.

Exemplo, seja uma função $f(x) = X_1 + X_2$, com parâmetros X_1 e X_2 variando em $[X_{\min}, X_{\max}]$. Então o grau de precisão (α) é avaliado por:

$$\alpha = \frac{(X_{\max} - X_{\min})}{2^N}$$

onde N é a quantidade de bits por parâmetro.

As principais etapas para a construção do algoritmo são:

- O cromossomo deve ser construído de acordo com cada problema particular (número de parâmetros, precisão etc);
- Criação de um operador de seleção que aumente a probabilidade de escolha e reprodução desses indivíduos (cromossomos) melhores adaptados à função objetivo;
- Definição de operadores para a recombinação e mutação de cromossomos.

Portanto, um algoritmo genético simples, mas que dá bons resultados em muitos casos práticos [29], é composto de três operadores:

- **Reprodução ou Seleção**

O processo de seleção do AG age, com características semelhantes às da seleção natural, aumentando a probabilidade de reprodução dos melhores indivíduos.

- **Recombinação ou “Crossover”**

Permite combinar dois cromossomos de uma mesma população para formar dois descendentes similares. É aplicado em posições randômicas, e com uma certa probabilidade de ocorrência (probabilidade de recombinação).

- **Mutação**

Altera arbitrariamente um ou mais bits do cromossomo selecionado para aumentar a diversidade da população. Na Figura 4.17 tem-se um exemplo, onde ocorre uma mutação pontual mudando-se 1's por 0's ou vice-versa.

O operador de mutação permite o algoritmo genético explorar novas possibilidades, impedindo assim, que este último fique preso em uma solução “local”. A probabilidade de mutação fica em torno de 1% a 5% dos bits de uma população.

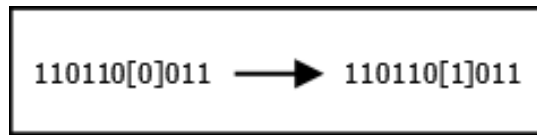


Figura 4.17: Exemplo do operador de mutação

Fonte: Adaptado de [10]

4.2.6 Algoritmo genético (AG) com parâmetros contínuos

O uso de AG com parâmetros codificados binariamente apresentam uma dificuldade quando aplicados em problemas com muitas variáveis. Isto se dá devido os cromossomos ter um grande número de “bits”, prejudicando assim, a eficiência dos operadores de recombinação e mutação e, acarretando uma taxa de convergência lenta do AG.

O algoritmo genético com parâmetros contínuos é bem semelhante ao de parâmetros codificados binariamente. A diferença do primeiro é apenas na forma de apresentar as variáveis e na forma de implementar os operadores de recombinação (*crossover*) e de mutação. Na Figura 4.18 é mostrado um fluxograma dos passos desse algoritmo [30].

- **Codificação e Precisão**

No caso do AG com parâmetros contínuos não é necessário definir o número de bits para representar cada parâmetro em função da precisão requerida, pois eles são representados diretamente como números reais, com a precisão interna do computador que estiver sendo usado. Neste caso, o cromossomo do AG é um vetor com as variáveis do problema:

$$\text{Cromossomo} = [p_1, p_2, p_3, \dots, p_N]$$

$$\text{Função custo} = f [p_1, p_2, p_3, \dots, p_N]$$

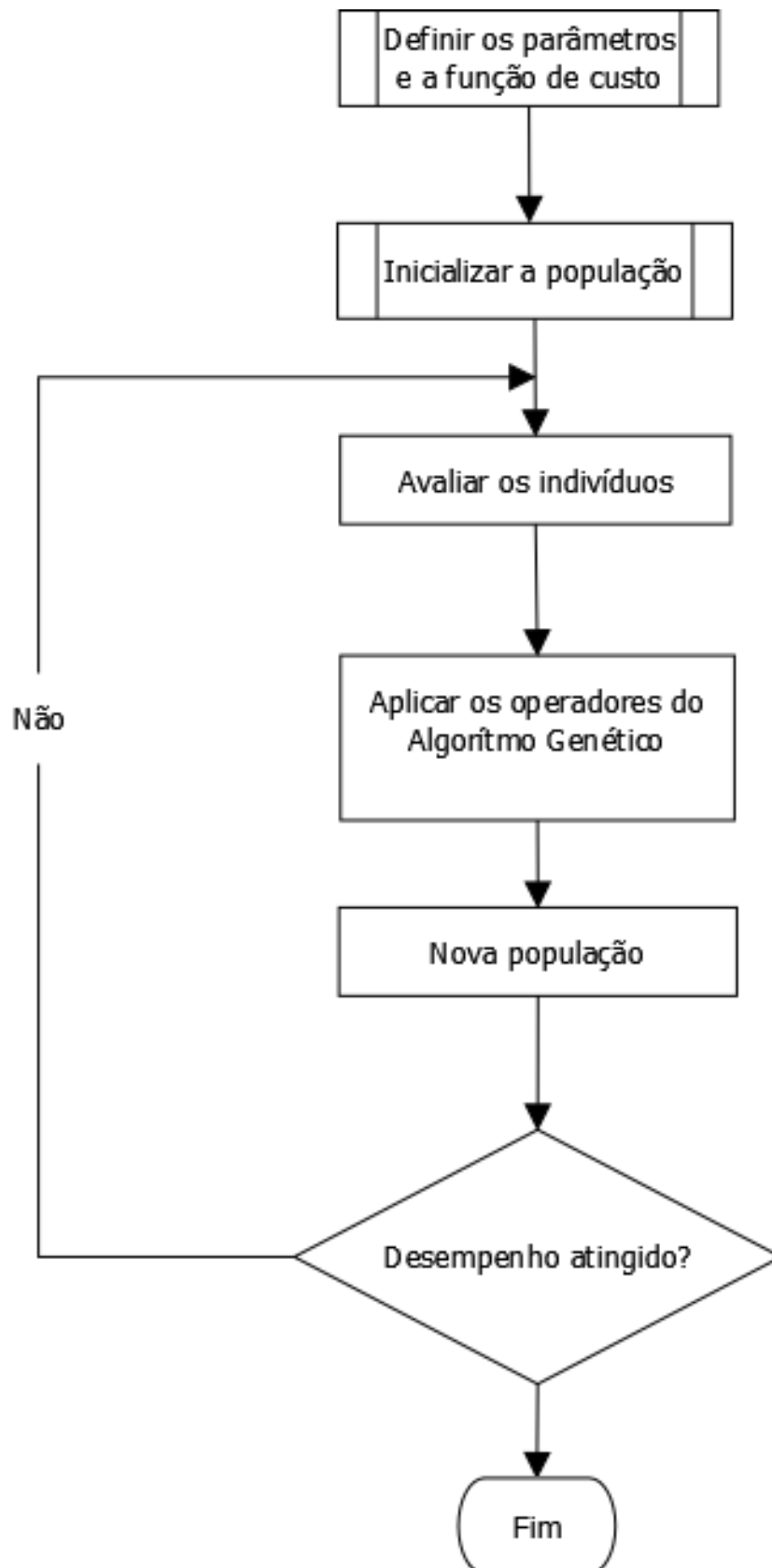


Figura 4.18: Esquemático do AG

Fonte: Adaptado de [10]

- **População inicial**

A população inicial pode ser obtida escolhendo-se um valor para os parâmetros ou variáveis de cada cromossomo randomicamente entre o seu valor mínimo e valor máximo.

$$Cromossomo(i) = [p_1, p_2, p_3, \dots, p_N]$$

Onde p_J = valor randômico entre $p_{J_{\min}}$ e $p_{J_{\max}}$.

Há um compromisso entre o tamanho da população inicial e número de gerações ou interações para se obter a convergência. De modo geral, o AG converge mais rapidamente, em termos de números de interações para a solução, quando a população inicial é maior. Um outro critério que deve ser avaliado para se comparar o desempenho do AG é comparar o número de vezes que a função custo foi avaliada. E quanto maior a população, maior número de avaliações da função custo.

- **Seleção natural**

A população inicial costuma ser maior do que a população nas gerações ou interações seguintes em alguns algoritmos genéticos. A escolha dos cromossomos que irão sobreviver, ou serão selecionados para serem aplicados os operadores de recombinação e mutação, deverá de alguma forma, valorizar aqueles cromossomos que apresentam melhor desempenho quando avaliados pela função custo do problema em questão. Assim sendo, este operador de seleção estará “imitando” a seleção natural e as espécies mais adaptados ao meio tendem a sobreviver. Logo, em um algoritmo de seleção clássico existe uma ordenação dos cromossomos de melhor para o pior, e normalizando assim, o valor de custo de cada cromossomo. E em seguida é realizada a escolha aleatória de um valor que representará a posição hierárquica do cromossomo sendo normalizado o seu custo.

- **O operador de recombinação (*crossover*)**

Este operador tem por objetivo combinar os parâmetros ou variáveis dos “pais”, gerando assim, cromossomos “filhos” que possam somar os pontos fortes de ambos produzindo, portanto, uma melhor solução para o problema em questão.

Há diversos operadores de recombinação. O mais simples consiste em escolher um ou vários pontos aleatórios de corte no cromossomo, e permutar a partir destes pontos as variáveis dos cromossomos “pais” para gerar os “filhos”, de acordo com o esquema abaixo:

$$Pai_1 = [p'_1, p'_2, \times p'_3, \dots, p_N] \Rightarrow Filho_1 = [p_1, p_2, \times p'_3, \dots, p'_N]$$

$$Pai_2 = [p_1, p_2, \times p_3, \dots, p_N] \Rightarrow Filho_2 = [p'_1, p'_2, \times p_3, \dots, p_N]$$

Neste operador nenhuma informação é introduzida nos cromossomos "filhos".

Uma outra possibilidade é realizar uma média ponderada entre os valores dos parâmetros. Tal método é chamado de "*blending*", sendo dado por:

$$P_{NOVO} = \beta \cdot P_{pai_1} + (1 - \beta) \cdot P_{pai_2}$$

onde β = fator de mistura ou parâmetro de ajuste podendo ser um valor randômico entre 0 e 1.

Neste operador nenhuma extrapolação além dos valores dos parâmetros existentes é feita.

Uma outra possibilidade é realizar a combinação linear entre os valores dos parâmetros. Tal método é chamado "*crossover*" linear. Neste, geram-se a partir dos "pais", três cromossomos "filhos", sendo *um* a média dos valores dos "pais", e os *outros* dois valores, extrapolados, de acordo com as equações (4.4), (4.5) e (4.6).

$$P_{NOVO_{-1}} = 0,5 \cdot P_{pai_1} + 0,5 \cdot P_{pai_2} \quad (4.4)$$

$$P_{NOVO_{-2}} = 1,5 \cdot P_{pai_1} - 0,5 \cdot P_{pai_2} \quad (4.5)$$

$$P_{NOVO_{-3}} = -0,5 \cdot P_{pai_1} + 1,5 \cdot P_{pai_2} \quad (4.6)$$

• Operador de mutação

Este operador tem por objetivo introduzir uma diversidade no cromossomo da população do AG, evitando assim, que o mesmo fique preso em mínimos locais.

A implementação se dá de forma simples definindo uma probabilidade de mutação (PM) e definindo o número de parâmetros da população que sofre mutação (N_{PARMUT})

$$(N_{PARMUT}) = PM \cdot N \quad (4.7)$$

onde N é o número de parâmetros da população.

Em seguida, escolhe-se de forma aleatória, dentro da população um parâmetro e aplica-se o operador de mutação dado por

$$P_{J-NOVO} = \text{valor randômico entre } P_{J_{\min}} \text{ e } P_{J_{\max}}. \quad (4.8)$$

Este procedimento será repetido até que se altere o número de parâmetros desejados em função da probabilidade de mutação.

Um outro operador é o “não uniforme”, nele a perturbação permitida nos parâmetros vai diminuindo em função do número de gerações ou de iterações do AG. Aqui também existe uma probabilidade de mutação e um algoritmo que escolhe aleatoriamente qual o parâmetro do cromossomo (indivíduo) que vai sofrer mutação:

Cromossomo = $[p_1, p_2, p_3 \dots, P_N]$, e seja P_J o parâmetro que sofre mutação.

Então:

$$P_{J\text{-NOVO}} = P_J + \Delta [t, (P_{J\text{max}} - P_J)] \text{ se } \phi = 0 \quad (4.9)$$

$$P_{J\text{-NOVO}} = P_J + \Delta [t, (P_J - P_{J\text{min}})] \text{ se } \phi = 1 \quad (4.10)$$

onde ϕ é um número aleatório igual a 0 ou 1, t é índice da interação ou geração atual do AG e a função Δ é:

$$\Delta [t, y] = y \cdot R (1 - (t/T))^B$$

onde T é o número máximo de interações ou gerações do AG, R é um número aleatório entre 0 e 1, B é o fator de sintonia do operador. Portanto, a escolha do operador mais adequado depende do problema em questão.

5 MANIPULADOR ROBÓTICO

São sistemas mecânicos multifuncionais, projetados para moverem materiais, peças, ferramentas ou dispositivos especiais em movimentos variáveis programadas para a realização de uma variedade de tarefas.



Figura 5.1: Manipulador robótico a esquerda e sua aplicação industrial a direita

Fonte: Embraer

O precursor do termo robô [31] por *Karel Capek*, novelista e escritor checo, que em sua famosa peça em 1923, intitulada R.U.R. (*Rosumovi univerzálni Roboti*), usou pela primeira vez a palavra “robot”, que significa a trabalho.

Um **robô** é uma máquina controlada por computador e envolve tecnologia intimamente associada com a automação. A robótica industrial é definida como uma área particular da automação na qual a máquina automática, isto é, o robô, é projetado para substituir a mão da obra humana [1]. Desse modo, os robôs possuem certas características humanas.

Na robótica moderna existem pesquisas e desenvolvimentos de robôs intitulados *humanóides* ou *antromórficos*. Estes criados com a semelhança humana e com capacidade de interagir com o ambiente. Na Figura 5.2 é mostrado um braço robótico fabricado pela *RRRobotica*.

Os robôs são totalmente programáveis, possuindo braços móveis, e são empregados em várias atividades, tais como:

- carregamento e descarregamento de máquinas;
- pintura ou jateamento;
- soldagem a ponto ou outra forma;

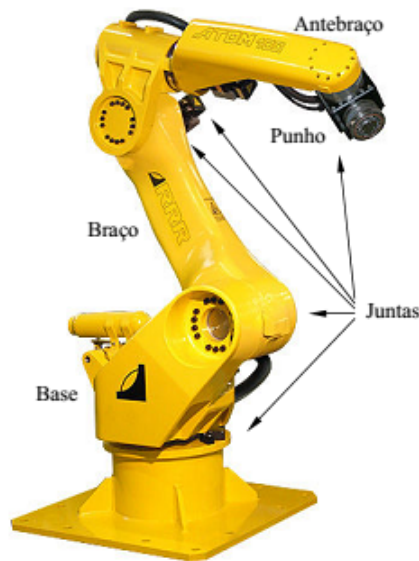


Figura 5.2: Anatomia do braço robótico fabricado pela RRRobotica

Fonte: RRRobotica

- processos de conformação ou usinagem.

Alguns fatores caracterizam também os manipuladores robóticos, tendo em vista, que em grande parte são responsáveis por tornar uma determinada configuração de braço mais adequado a uma dada automação. Esses fatores são:

- anatomia;
- volume de trabalho;
- sistemas de acionamento;
- sistemas de controle;
- desempenho e precisão;
- órgãos terminais;
- sensores;
- programação.

A anatomia de um manipulador robótico, segundo *Grover* (1988), é composto de *braço* e *punho*. O braço composto por elementos denominados *elos* unidos por juntas de movimentos relativo, no qual são acoplados os acionadores para a realização dos movimentos individualmente e dotados de capacidade sensorial sendo instruídos por um sistemas de controle. O punho consiste de várias juntas próximas entre si, unidas por elos compostos, permitindo assim, a orientação do órgão terminal (mão ou ferramenta) nas posições que correspondem à tarefa a ser realizada. A Figura 5.3 mostra esquematicamente um manipulador ou braço robótico.

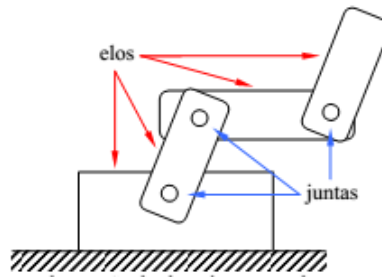


Figura 5.3: Esquema de notação de um braço mecânico ilustrativo

Fonte: [32]

As juntas [6] podem ser rotativas, prismáticas, cilíndricas, esféricas, de parafusos e planares. De acordo com os movimentos permitidos estas podem mover-se em uma, duas ou três direções, ou **graus de liberdade** (GL).

Nessa dissertação utiliza-se um modelo de manipulador robótico que possui um sistema de quarta ordem com representação no espaço de estados. O objetivo é realizar o controle de modo que as forças de impacto e as de regime permanente não venha a danificar os objetos que serão manipulados. Ao mesmo tempo este deve fornecer uma força suficiente para a execução de determinada tarefa. Na Figura 5.4 é apresentado o sistema em estudo.

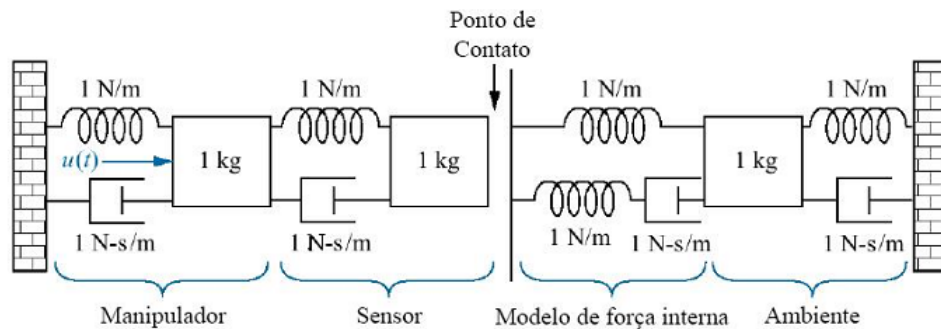


Figura 5.4: Manipulador robótico e ambiente

Fonte: Adaptado de [13]

O sensor desse manipulador deve ser posicionado sobre a superfície de contato sempre de forma controlada, evitando assim, danos a seus componentes. Logo a saída do sistema é o deslocamento do sensor e, a sua entrada uma excitação $u(t)$. Neste sistema são incluídos dois regimes, um sem contato e outro com contato do sensor à superfície de contato. No entanto, para descrever a dinâmica do sistema deve ser encontrada uma função considerando ele como um todo. Mas para testar a metodologia, será analisado o regime sem contato com o meio ambiente.

6 CONTROLE INTELIGENTE

Visando a busca de um ganho ótimo, nesse projeto de pesquisa é utilizado a técnica de controle moderno o *Regulador Linear Quadrático (LQR)*. A utilização do projeto (LQR) tem-se mostrado uma alternativa em diversas aplicações em sistemas modernos, como exemplo, temos no uso de projetos de geração de energia, sejam elas, eólicas, térmicas e nucleares, nos controles de processos químicos e nos controles de servomecanismos em robótica. Isto devido às suas excelentes características de desempenho bem como a sua simples exemplificação.

São apresentadas nesse projeto dois métodos de Inteligência Computacional para resolver o problema do controlador *LQR*. O desenvolvimento destes métodos são resultados de pesquisas utilizando *Algoritmos Genéticos (AG)* na seleção nas matrizes de ponderação Q e R com fins de minimizar o índice de desempenho quadrático, garantindo assim, a estabilidade e robustez ao sistema e na solução da *Equação Algébrica de Riccati (EAR)* por meio de uma *Rede Neural Recorrente (RNR)* [33, 34, 35]. Na Figura 6.1 temos um diagrama que representa os métodos utilizados no controle ótimo na busca das matrizes de ponderação e na solução da *ARE*.

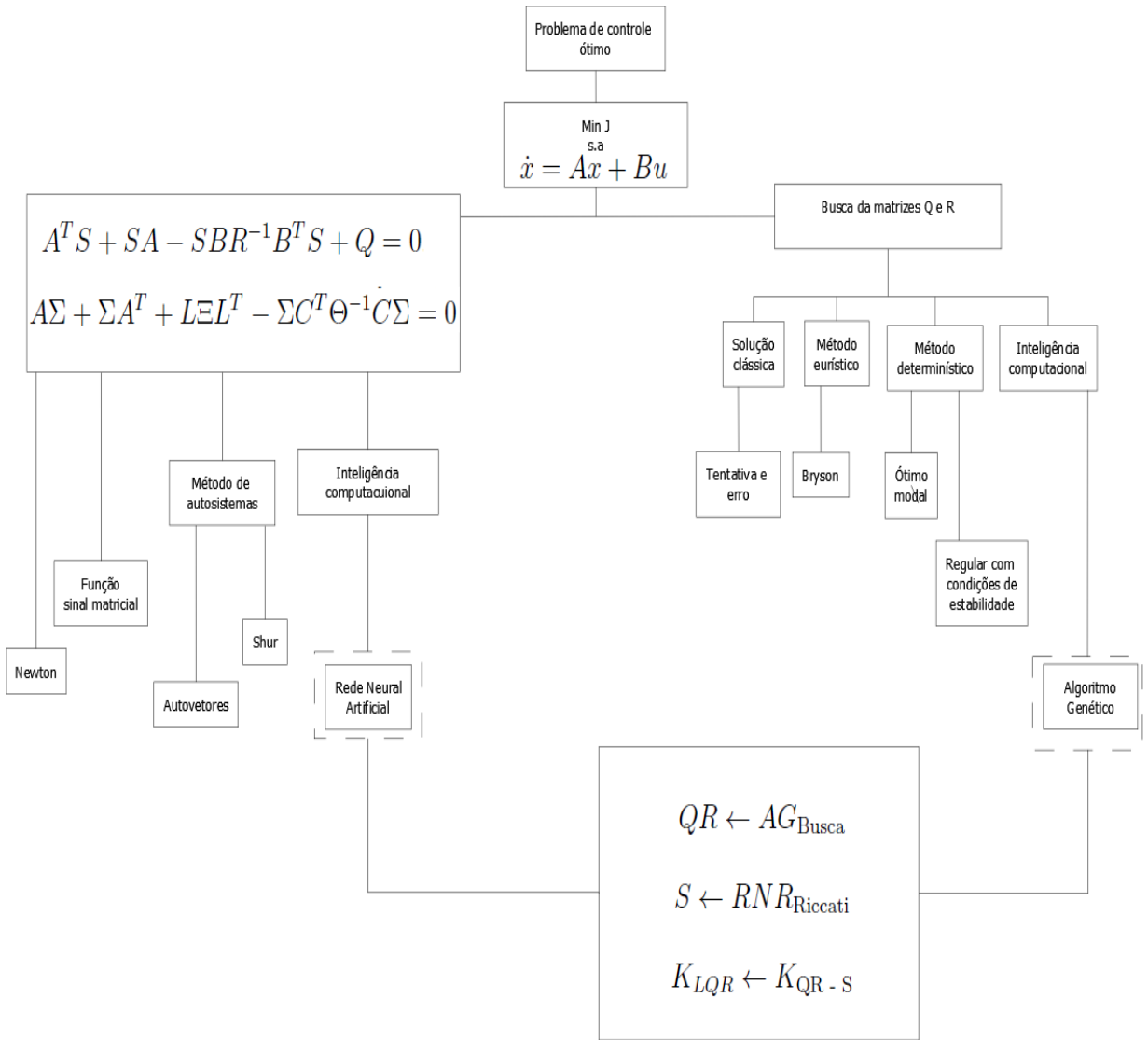


Figura 6.1: Diagrama dos métodos utilizados no controle ótimo LQR

Fonte: Adaptado de [36]

6.1 Modelagem clássica do projeto LQR/LQG

O modelo clássico de controle do projeto do *Regulador Linear Quadrático* (LQR) busca o vetor de controle \mathbf{u} que minimiza o funcional J com restrições para o modelo do sistema. Dentre as vantagens desse tipo de projeto, a margem de estabilidade garantida, ou seja, margens de fase de 60° e margens de ganho infinito, garantindo assim, a qualidade de robustez de estabilidade. Dessa forma o problema é dado pelas Equações (6.2) e (6.3)

$$\min J = \frac{1}{2} \int_{t_0}^T [x^T \mathbf{Q}x + u^T \mathbf{R}u] dt \tag{6.1}$$

$$s.a \quad \dot{x} = \mathbf{A}x + \mathbf{B}u \quad e \quad y = \mathbf{C}x . \tag{6.2}$$

Logo, o *LQR* tem a equação de estado linear variante no tempo, Equação (6.3), e o índice de desempenho quadrático, Equação (6.4), dados respectivamente, por

$$\dot{x} = \mathbf{A}(t) + \mathbf{B}(t) \quad (6.3)$$

$$J(t_0) = \frac{1}{2}x^T(\mathbf{T})\mathbf{S}(\mathbf{T})x(\mathbf{T}) + \frac{1}{2}\int_{t_0}^{\mathbf{T}} [x^T\mathbf{Q}x + u^T\mathbf{R}(t)u] dt \quad (6.4)$$

onde, $x \in \mathbb{R}^n$ e $u \in \mathbb{R}^m$, o par (\mathbf{A}, \mathbf{B}) é controlável, (\mathbf{A}, \mathbf{C}) é observável, $\mathbf{Q} \geq 0$ (semidefinida positiva) $\mathbf{R} > 0$ (definida positiva).

O Hamiltoniano é dado por

$$\mathbf{H}(t) = \frac{1}{2}(x^T\mathbf{Q}x + \mathbf{U}^T\mathbf{R}u) + \lambda^T(\mathbf{A}x + \mathbf{B}u). \quad (6.5)$$

Para um controlador ótimo temos

$$\dot{x} = \frac{\partial \mathbf{H}}{\partial \lambda} = \mathbf{A}x + \mathbf{B}u, \quad (6.6)$$

$$-\dot{\lambda} = \frac{\partial \mathbf{H}}{\partial x} = \mathbf{Q}x + \mathbf{A}^T\lambda, \quad (6.7)$$

$$0 = \frac{\partial \mathbf{H}}{\partial u} = \mathbf{R}u + \mathbf{B}^T\lambda. \quad (6.8)$$

Da Equação (6.8), tem-se

$$u(t) = -\mathbf{R}^{-1}\mathbf{B}^T\lambda. \quad (6.9)$$

Substituindo a Equação (6.9) em (6.6), tem-se:

$$\dot{x} = \mathbf{A}x - \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\lambda. \quad (6.10)$$

Na forma matricial teremos:

$$\begin{bmatrix} \dot{x} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} \quad (6.11)$$

em que

$$\mathbf{H} = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \quad (6.12)$$

a Equação (6.12) é a matriz Hamiltoniana.

Reduzindo-se essa matriz acima por métodos encontrados em [25] encontra-se a equação contínua de Riccati dada por

$$\mathbf{A}^T\mathbf{S} + \mathbf{S}\mathbf{A} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S} + \mathbf{Q} = 0, \quad t \leq \mathbf{T}. \quad (6.13)$$

Todo o método de cálculo variacional [37] são utilizados para resolver o problema da otimização dadas pela Equação (6.2) dado $u = \mathbf{K}x$ onde $\mathbf{K} = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S}$ que é o ganho do controlador.

6.2 Modelagem inteligente utilizando o Algoritmo Genético

A abordagem inteligente é formulada para realizar a busca das matrizes de ponderação Q e R de estado de controle por meio de um Algoritmo Genético (AG). Os parâmetros matriciais do projeto LQR são utilizados para realizar a sintonia do ganho ótimo do compensador. O modelo matemático não-linear para exibir essas matrizes é:

$$\min \sum_{i=1}^n p_i s_i(\mathbf{Q}, \mathbf{R}) s_i(\mathbf{Q}, \mathbf{R}) \leq 1, \quad i = 1, \dots, n, \quad \lambda_{ei} \leq \lambda_{ci}(\mathbf{Q}, \mathbf{R}) \leq \lambda_{di}, \quad i = 1, \dots, n \quad (6.14)$$

sendo p_i os pesos das i -ésimas ponderações das sensibilidades $s_i = S_i/\epsilon_i$ normalizada com relação à i -ésima especificação de projeto $\epsilon_i > 0$. Esse modelo aborda desde a representação genética das matrizes de ponderação, passando pelas operações cromossômicas, até chegar às avaliações de **fitness** (desempenho).

Introduzidos por *John Holland* [28] e popularizados por um de seu alunos, *David Goldberg* [31], os Algoritmos Genéticos (AG) são esquemas de busca e otimização baseados em probabilidades [7]. Estes são insensíveis à forma da função objetivo, sendo igualmente eficientes em funções descontínuas, ruidosas, multidimensionais e multimodais [29].

A ideia principal dos AG é a partir de uma população de soluções para um dado problema de otimização, gerada aleatoriamente ou segundo alguma heurística [38], gerar populações consecutivas de acordo com os passos enunciados [6, 29, 39]:

- Avaliar a qualidade de cada uma das soluções da população atual definindo probabilidades para essas soluções. Sendo que as melhores terão maior probabilidade de serem selecionadas para a próxima população e as ruins terão menor probabilidade de serem selecionadas para gerar novos cromossomos.

- Enquanto o tamanho fixo das populações não tiver sido alcançado, deve-se:

1. continuar selecionando duas soluções da população corrente de acordo com as probabilidades pré-definidas;
2. submetê-las aos operadores genéticos de recombinação (*Crossover*) e mutação, gerando assim, duas novas soluções que farão parte da nova população e darão ao AG opções de novos caminhos dentro do espaço de soluções.

Para a montagem de um modelo genético artificial para matrizes de ponderação Q e R a fim de reduzir a carga computacional considera-se inicialmente a ordem das matrizes $A_{n \times n}$, $B_{n \times m}$ e $C_{n \times p}$ do sistema dinâmico da Equação (6.2). A dimensão das matrizes apresentadas e as especificações do projeto LQR pode-se montar uma matriz $Q_{n \times n}$ e uma matriz $R_{m \times m}$ simétricas, positivas e semipositivas definidas, respectivamente. Logo a representação clássica matricial será:

$$Q = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} & \cdots & q_{1n} \\ q_{21} & q_{22} & q_{23} & q_{24} & \cdots & q_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \\ q_{n1} & q_{n2} & q_{n3} & q_{n4} & \cdots & q_{nn} \end{bmatrix} \quad (6.15)$$

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ r_{21} & r_{22} & \cdots & r_{2m} \\ \vdots & \vdots & \vdots & \\ r_{m1} & r_{m2} & \cdots & r_{mm} \end{bmatrix} \quad (6.16)$$

O cromossomo que irá representar as matrizes Q e R será:

$$QR_k = \bigcup_{i,j=1}^n q_{i,j} \wedge \bigcup_{\iota,l=1}^m r_{\iota,l} \quad (6.17)$$

$k = 1, \dots, crom$

onde:

- n - números de linhas ou colunas de A ;
- m - número de colunas de B ;
- $q_{i,j}$ - gene de Q ;
- $r_{\iota,l}$ - gene de R ;
- $crom$ - número de cromossomos.

A população inicial para este AG é dada por

$$QR_{crom \times g} = [QR_1; QR_2; QR_3; \dots; QR_{crom}] \quad (6.18)$$

onde a notação $crom \times g$ indica a quantidade de cromossomos e genes da população inicial.

O **Fitness** (desempenho) de cada cromossomo do indivíduo que compõe a população $QR_{crom \times g}$ é avaliado e cada indivíduo QR_z é pontuado, isto é, $QR_{crom \times g}(QR_1)$. Todo esse processo é realizado recuperando-se a forma original das matrizes Q e R executadas por cálculos relativos ao LQR . Assim são encontrados autovalores e autovetores

associados ao sistema e a matriz $K_{m \times n}$ é gerada. Logo K_z é o ganho gerado por um indivíduo, A_z é a matriz de malha fechada para este ganho K_z e S_z são as sensibilidades associadas com cada autovalor.

A estratégia de pontuação de cada indivíduo se dá da seguinte forma: para cada autovalor, dentro da margem especificada na estrutura de otimização não linear pelo projetista, atribui-se um ponto (1) ao indivíduo QR_1 e um centésimo (0,01) para cada autovetor associado que possua sensibilidade normalizada $S_z < 1$. O indivíduo que obtiver maior pontuação será considerado o ótimo. Em caso de empate dos indivíduos será considerado o melhor aquele que possuir menor soma das sensibilidades normalizadas.

Sendo a reprodução um dos principais elementos de um AG, pois é responsável pela **seleção** probabilística dos indivíduos que irão continuar no processo de desenvolvimento. Logo, a escolha desses indivíduos é baseada na função *Fitness* (desempenho), ou seja, aquele que tiver maior pontuação terá maior chance de sobreviver. Essa seleção é feita pela roleta cuja montagem é relativamente simples e se dá pela soma parcial da *Fitness* de cada elemento QR_k formando intervalos, conforme mostra a Equação (6.19):

$$Fitness(i) = \sum_{j=1}^{crom} Fitness(j) \quad i = 1, \dots, crom \quad (6.19)$$

logo em seguida gera-se um número aleatório entre zero “0” e o somatório das *Fitness* individuais. Se este número estiver dentro do intervalo, o elemento escolhido de $QR_{crom \times g}$ será o QR_k .

Portanto o perfil da população inicial e final pode ser definido pelos valores da **função objetivo** a fim de encontrar o indivíduo QR_k ótimo.

7 SOLUÇÃO NEURAL DA EQUAÇÃO ALGÉBRICA DE *RICCATI*

As equações de *Riccati*, dentre as equações matriciais não lineares, são as mais estudadas por matemáticos e engenheiros. Podem ser qualquer classes de matrizes: quadráticas, algébricas ou diferencial, surgidas no estudo de sistemas dinâmicos contínuos ou discretos no tempo [40]. Essas desempenham um papel fundamental na solução de problemas de controle linear quadrático gaussiano, estimação de estado e de parâmetros de sistemas, modelagens de séries temporais multivariáveis, e em muitos outros ramos da matemática aplicada.

Atualmente muitas pesquisas têm sido reportadas para resolver sistemas de equações lineares em problemas relacionados com Redes Neurais Artificiais (RNA) .

Nesse trabalho, o objetivo é resolver problemas de controle ótimo *LQR* para sistemas lineares contínuos no tempo utilizando-se a Equação Algébrica de *Riccati* (EAR). Para isso, usa-se o problema de otimização combinatória. Dadas as matrizes A , B , Q e R , o objetivo é encontrar uma solução matricial simétrica e definida positiva P . Logo toda a estrutura é formulada para minimizar uma função energia monotônica não decrescente e limitada inferiormente, $\varepsilon(g, h)$ [35].

Considere o problema de otimização:

$$\min E(\mathbf{P}, \mathbf{L}) = \sum_{i=1}^n \sum_{j=1}^n \{e_{ij}[g_{ij}(\mathbf{P})] + e_{ij}[h_{ij}(\mathbf{P}, \mathbf{L})]\} \quad (7.1)$$

onde $e_{ij} : \mathbb{R} \rightarrow \mathbb{R}$ é uma função matricial convexa e,

$$G(P) = [g_{ij}(P)] = PSP - A^T P - PA - Q; \quad (7.2)$$

$$H(P, L) = [h_{ij}(P, L)] = LL^T - P; \quad (7.3)$$

$$G(P) = g_{ij}(P) = \sum_{k=1}^n \sum_{l=1}^n p_{ik} s_{kl} p_{lj} - \sum_{k=1}^n (a_{ki} p_{kj} + p_{ik} a_{kj}) - q_{ij}; \quad (7.4)$$

$$h_{ij}(P, L) = \sum_{k=1}^{\min\{i,j\}} l_{ik} l_{jk} - p_{ij}. \quad (7.5)$$

Nota-se que e_{ij} é definida somente em função de g_{ij} ou h_{ij} .

As equações dinâmicas da Rede Neural Recorrente (RNR) para a resolução da EAR na forma matricial para $i, j = 1, 2, \dots, n$ é dada por:

$$\frac{dp_{ij}(t)}{dt} = -\eta_p \left\{ \sum_{k=1}^n \sum_{l=1}^n [p_{ik}(t)s_{kl}u_{lj}(t) + u_{lk}(t)s_{kl}p_{lj}(t)] - \sum_{k=1}^n [a_{ik}u_{kj}(t) + u_{ik}(t)a_{jk}] - y_{ij}(t) \right\} \quad (7.6)$$

$$\frac{dz_{ij}(t)}{dt} = -\eta_z \sum_{k=j}^n y_{ik}(t)z_{kj}(t), i \geq j \quad (7.7)$$

$$u_{ij}(t) = f_{ij} \left[\sum_{k=1}^n \sum_{l=1}^n p_{ik}(t)s_{kl}p_{lj}(t) - \sum_{k=1}^n [a_{ki}p_{kj}(t) + p_{ik}(t)a_{kj}] + q_{ij} \right] \quad (7.8)$$

$$y_{ij}(t) = f_{ij} \left[\sum_{k=1}^{\min\{i,j\}} z_{ik}(t)z_{jk}(t) - p_{ij}(t) \right]. \quad (7.9)$$

Escrito na forma matricial,

$$\frac{dP(t)}{dt} = -\eta_p [P(t)SU(t) + U(t)SP(t) - AU(t) - U(t)A^T - Y(t)] \quad (7.10)$$

$$\frac{dZ(t)}{dt} = -\eta_z Y(t)Z(t) \quad (7.11)$$

$$U(t) = F [P(t)SP(t) - A^T P(t) - P(t)A + Q] \quad (7.12)$$

$$Y(t) = F [Z(t)Z^T(t) - P(t)] \quad (7.13)$$

onde $U(t) = [u_{ij}(t)]$, $P(t) = [p_{ij}(t)]$ e $Y(t) = [y_{ij}(t)]$ são matrizes quadradas dos estado de ativação de ordem $n \times n$, $Z(t) = [z_{ij}(t)]$ é uma matriz triangular inferior dos estados de ativação de ordem $n \times n$, e esta representa o fator de **Cholesky**. $F = [f_{ij}]$ é uma matriz não decrescente de funções de ativações; $P(t)$ é a solução da EAR. Os parâmetros positivos de ajuste da rede neuronal são os pesos η_p e η_z . A matriz S representa na Equação (7.12) a influência da matriz de ponderação R , sendo dada por $S = BR^{-1}B^T$, sendo que os ganhos do controlador ótimo são ajustados tendo como base as variações das matrizes Q e R .

A Figura 7.1 ilustra a arquitetura da rede neuronal para resolver a EAR, sendo formada por quatro camadas conectadas bidirecionalmente. A camada de saída $P(t)$ é a solução da EAR, $Z(t)$ representa o fator de *Cholesky* de P . Foi utilizado o método de *Runge-Kutta* de quarta ordem para computar os valores de P e Z . A função de desempenho para mensurar o processo de treinamento da RNAR é a função *Erro Médio Quadrático* (EMQ) que representa a função energia.

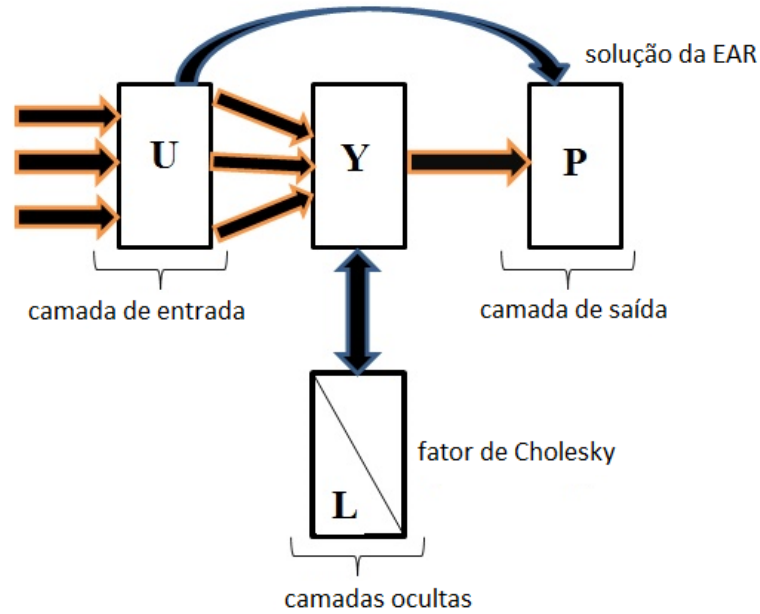


Figura 7.1: Arquitetura do RNAR para a EAR

7.0.1 Análise de estabilidade

A rede recorrente proposta é praticamente um sistema dinâmico não-linear contínuo. A estabilidade dos sistemas dinâmicos não-lineares pode ser analisada usando o método direto de *Lyapunov* dados pelos Teoremas 7.0.1 e 7.0.2 [35].

Teorema 7.0.1. *Se cada função de ativação é contínua, pelo menos diferenciável por partes, monótona não decrescente (isto é, $df_{ij}(\xi)/d\xi \geq 0$ para $-\infty < \xi < +\infty$), e $f_{ij}(\xi) = 0$ se $\xi = 0$ ($i, j = 1, 2, \dots, n$), então a rede neural recorrente é considerada assintoticamente estável (isto é, $\forall \mathbf{P}(0), \mathbf{Z}(0), \exists \mathbf{Z}, \mathbf{P}$ de modo que o $\lim_{t \rightarrow \infty} \mathbf{Z}(t) = \bar{\mathbf{Z}}$ e $\lim_{t \rightarrow \infty} \mathbf{P}(t) = \bar{\mathbf{P}}$).*

7.0.2 Análise da solvabilidade

Consiste em verificar se o problema de controle automático admite ou não solução. A análise de estabilidade indica que a transição de estado de ativação forma um fluxo de gradiente que minimiza E . O seguinte teorema fornece uma condição necessária e suficiente para que o fluxo de gradiente converge para a solução final positiva da equação algébrica de *Riccati*, \mathbf{P} , de praticamente qualquer estado inicial [35].

Teorema 7.0.2. *Suponha que cada função de ativação é contínua, pelo menos diferenciável por partes, monótona não decrescente (isto é, $df_{ij}(\xi)/d\xi \geq 0$ para $-\infty < \xi < +\infty$), e $f_{ij}(\xi) = 0$ se $\xi = 0$. A matriz de estado estacionário da rede neural recorrente é sempre*

a solução simétrica e definida positiva para a equação de Riccati de qualquer estado inicial simétrico $\mathbf{P}(0)$ e quaisquer estados iniciais não nulos $\mathbf{Z}(0)$ (isto é, $\forall \mathbf{P}(0) = \mathbf{P}(0)^T$, $\forall \mathbf{Z}(0) \neq 0$, $\bar{\mathbf{P}} = \mathbf{P}$) se e somente se o posto $(\bar{\mathbf{P}}\mathbf{S} - \mathbf{A}) = n$ e todos o elementos diagonais em \mathbf{Z} são não nulos (ou seja, $\forall i, \bar{z}_i \neq 0$ para $i = 1, 2, \dots, n$).

8 FILTRO DE KALMAN PARA ESTIMAÇÃO DOS ESTADOS

A alocação de autovalores foi a primeira aplicação do método de espaço de estados em sistemas lineares [41]. Os primeiros pesquisadores a perceber o problema de alocação de polos e sua importância foram *J. Bertram* em 1959 e o matemático húngaro *Rudolf Emil Kalman*. Os trabalhos de *Kalman* certamente constituem contribuições muito importantes e também precursoras para a concepção e a solução do problema de alocação de autovalores otimizados ou não [42, 43] e inspiram o desenvolvimento dos primeiros resultados analíticos por *Rissanem* [44] que escreveu a primeira publicação sobre o assunto. Uma das grandes contribuições de *Kalman* foi a invenção de um filtro que hoje leva o seu nome, **Filtro de *Kalman***, utilizado amplamente em projetos de engenharia de controle.

O Filtro de *Kalman* é um método de estimação estocástico utilizado para obter estimativas ótimas das variáveis de estados de um sistema dinâmico de modo que o erro seja minimizado estatisticamente.

Nessa dissertação foi adotado o Filtro de *Kalman* (FK) com objetivo de alocar autovalores e autovetores no problema de estimação do estado estocástico. O diferencial neste trabalho é o emprego das técnicas de computação evolutiva, com Algoritmos Genéticos, para determinar um observador estocástico a fim de estimar os estados com uma dinâmica desejada. Algumas restrições de auto estruturas, autovetores e de autovalores são usados de modo que se possa formular o problema de estimação de estados estocástico. Logo, o desejado é encontrar o ganho $\mathbf{L} = \Sigma(E, \Theta)C^T\Theta^{-1}$ do FK para justamente satisfazer essas restrições.

O Algoritmo Genético (AG) desenvolvido para o Filtro de *Kalman* (FK) tem a função de fazer a busca das matrizes de covariâncias Ξ e Θ respectivamente da perturbação do estado e do ruído de medida do filtro. Para tanto, tem-se interesse em desenvolver um algoritmo numérico, que possa ser utilizado em qualquer problema de estimação de estado estocástico, com o objetivo de determinar o ganho $\mathbf{L}_{(E,\Theta)}$ de forma a alocar autovalores e autovetores.

Devido a dualidade entre o *LQR* e o Filtro de *Kalman* (FK) pode-se utilizar, no desenvolvimento do algoritmo, relações análogas nos dois casos simplesmente fazendo a correspondência entre os parâmetros indicados na Tabela 8.1.

Considere o sistema dinâmico representado por

$$\dot{x}(t) = \mathbf{A}x(t) + \mathbf{G}\xi(t), \quad (8.1)$$

onde $x(t) \in \mathbb{R}^n$, $\xi(t) \in \mathbb{R}^m$ é um processo estocástico chamado de ruído no estado (ou no processo), que se admite ser branco, gaussiano e tal que

$$E[\xi(t)] = 0 \quad (8.2)$$

$$E[\xi(t)\xi(t+\tau)^T] = \Xi\delta(t-\tau) \quad (8.3)$$

$\Xi = \Xi^T > 0$ é a matriz de covariância do ruído no estado ou densidade espectral do ruído do estado, que indica a potência do ruído e $\delta(t-\tau)$ é o **Delta de Dirac** que traduz se o valor do ξ no instante t não está relacionado com seu valor em outro instante $t \neq 1$.

Admita-se que os sensores meçam não o estado diretamente, mas apenas variáveis de saída sujeitos à ação de um ruído de medida aditivo, isto é,

$$y(t) = \mathbf{C}x(t) + \nu(t) \quad y(t) \in \mathbb{R}^m \quad (8.4)$$

$\nu(t)$ é suposto branco, gaussiano, independente de $\xi(t)$ e tal que

$$E[\nu(t)] = 0 \quad (8.5)$$

$$E[\nu(t)\nu(t+\tau)^T] = \Theta\delta(t-\tau) \quad (8.6)$$

$$E[\xi(t)\nu(t+\tau)^T] = 0, \forall t, \tau. \quad (8.7)$$

Onde $\Theta = \Theta^T > 0$ é a matriz de covariância (ou densidade espectral) do ruído de medida que indica a *potência* do ruído. O par (\mathbf{A}, \mathbf{C}) da Equação (6.2) é suposto observável e o sistema dinâmico descrito pode ser representado na forma de diagrama de blocos, conforme mostra a Figura 8.1

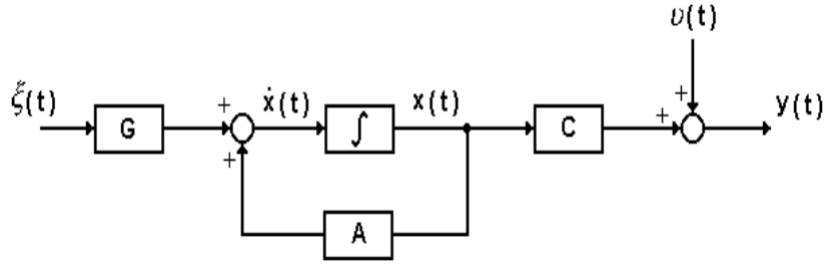


Figura 8.1: Diagrama de blocos do sistema

O problema a ser resolvido consiste em obter uma estimativa $\hat{x}(t)$ do estado $x(t)$ a partir da observação da saída $\{y(\tau), \tau \leq t\}$.

O Filtro de *Kalman* (FK) é um sistema dinâmico, com a estrutura representada na Figura 8.2, em que a matriz de ganho L é dada por

$$L = \Sigma C^T \Theta^{-1} \quad (8.8)$$

onde Σ é a única solução simétrica definida positiva da Equação Algébrica de *Riccati* (EAR) [9], apresentada na Equação (8.9)

$$A\Sigma + \Sigma A^T + L\Xi L^T - \Sigma C^T \Theta^{-1} C \Sigma = 0. \quad (8.9)$$

A estimativa achada pelo filtro é ótima no sentido de que a variância do erro da estimativa $\tilde{x}(t) = x(t) - \hat{x}(t)$ é mínima, isto é,

$$\min \sum_{i=1}^n E \{ [x_i(t) - \hat{x}_i(t)]^2 \}. \quad (8.10)$$

Sendo assim, a dinâmica do Filtro de *Kalman* (FK) é dada por

$$\dot{\hat{x}}(t) = A\hat{x}(t) + L[y(t) - C\hat{x}(t)]. \quad (8.11)$$

Filtro de *Kalman* é tal que

$$\mathbf{Re}[\lambda_i(A - LC)] < 0 \quad (i = 1, 2, 3, \dots, n) \quad (8.12)$$

o que significa que a *malha objetivo* é nominalmente estável.

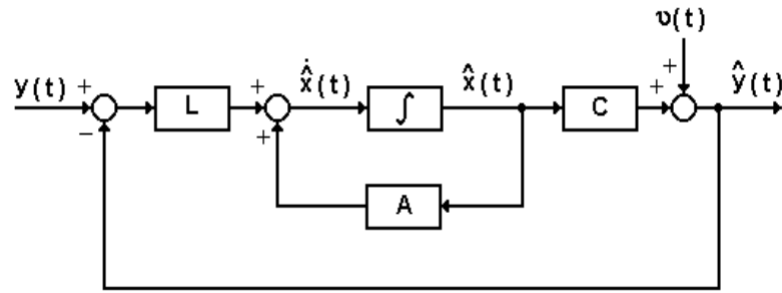


Figura 8.2: Diagrama de blocos do Filtro de *Kalman*

Sendo o Filtro de *Kalman* e o *LQR* duais [45], significando que é possível obter relações análogas nos dois casos, simplesmente fazendo-se a correspondência entre os parâmetros indicados na Tabela 8.1.

LQR	FK
A	A^T
B	C^T
Q	$L \Xi L^T$
R	Θ
K	Σ
G	H^T

Tabela 8.1: Parâmetros duais do *LQR* e do *FK*

Fonte: [8]

Foi verificado, então, que uma estimativa ótima do estado pode ser obtida, a partir de medidas afetadas por ruído de combinações lineares das variáveis de estado, usando um filtro. Esse filtro é constituído de um modelo do sistema com um sinal de realimentação proporcional da diferença entre a medida real e esperada.

9 EXPERIMENTO COMPUTACIONAL DO MANIPULADOR ROBÓTICO

9.1 Modelamento matemático do manipulador

O modelo matemático desse manipulador robótico é dado por:

$$\ddot{x}_r + 2\dot{x}_r + 2x_r - x_s - \dot{x}_s = u(t); \quad -\dot{x}_r - x_r + \ddot{x}_s + \dot{x}_s + x_s = 0 \quad (9.1)$$

Este manipulador deve ser *controlável* e *observável* de acordo com as Definições 9.1.1 e 9.1.2.

Definição 9.1.1. Um sistema é completamente controlável se existe um controle $u(t)$ sem restrições que pode transferir qualquer estado inicial $x(t_0)$ para qualquer outra posição desejada $x(t)$ em um tempo finito, $t_0 \leq t \leq T$ [1].

Para o sistema

$$\dot{x} = \mathbf{A}x + \mathbf{B}u. \quad (9.2)$$

Pode-se determinar se o sistema é **controlável** pela seguinte condição algébrica:

$$\text{rank}[B \ AB \ A^2B \ \dots \ A^{n-1}B] = n \quad (9.3)$$

onde *rank* é o posto da matriz A de ordem $n \times n$ e a matriz B é da ordem $n \times 1$. Para sistemas com múltiplas entradas, B pode ser da ordem $n \times m$, em que m é o número de entradas.

Para um sistema de entrada única e saída única, a **matriz de controlabilidade** (P_c) é descrita em função de A e B como

$$P_c = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} \quad (9.4)$$

a qual é uma matriz $n \times m$. Portanto, se o determinante de P_c é diferente de zero, o sistema é controlável.

Definição 9.1.2. Um sistema é completamente observável se e somente se existe um tempo finito T tal que o estado inicial $x(0)$ pode ser determinado a partir do histórico de observações de $y(t)$ dado o controle $u(t)$, $0 \leq t \leq T$ [1].

Considere o sistema com entrada única e saída única

$$\dot{x} = \mathbf{A}x + \mathbf{B}u \quad e \quad y = \mathbf{C}x \quad (9.5)$$

em que \mathbf{C} é um vetor linha $1 \times n$ e \mathbf{x} é um vetor coluna $n \times 1$. Portanto, este sistema é completamente observável quando o determinante da **matriz de observabilidade** (P_o) é diferente de zero, sendo

$$P_o = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (9.6)$$

que é uma matriz $n \times n$.

Para o manipulador em estudo, as variáveis de estado são $x_1 = x_r$; $x_2 = \dot{x}_r$; $x_3 = x_s$; $x_4 = \dot{x}_s$. Sendo assim, procura-se um compensador $K(s)$ que atue junto com o sistema em malha fechada; que mantenha a estabilidade e ainda tenha Graus de Liberdade (GL) em número suficiente e de acordo com o número de movimentos (uma, duas, três ou mais dimensões). Assim, as condições de desempenho e estabilidade serão satisfeitas.

9.1.1 Equações de estado e de saída do manipulador robótico

Para o manipulador robótico em estudo, as equações no espaço de estado são encontradas utilizando a equação fundamental da dinâmica de *Newton* aplicada ao sistema.

A etapa de modelagem é mostrada na Figura 9.1, nela x_r é o deslocamento do manipulador robótico, x_s é o deslocamento do sensor, e x_e é o deslocamento do ambiente. As equações dinâmicas do sistema do manipulador sem contato com o ambiente alvo é dada por:

$$\begin{cases} \ddot{x}_r + 2\dot{x}_r + 2x_r - x_s - \dot{x}_s = u(t); \\ -\dot{x}_r - x_r + \ddot{x}_s + \dot{x}_s + x_s = 0 \end{cases}$$

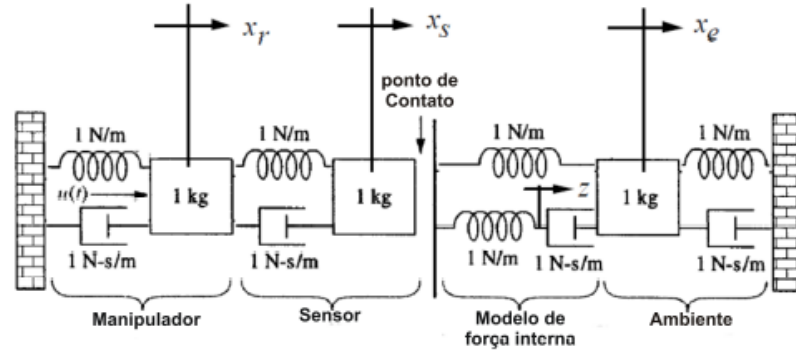


Figura 9.1: Análise dos deslocamentos do Manipulador Robótico e ambiente-alvo.

Fonte: Adaptado de [13]

Assume-se as variáveis de estado como:

$$\begin{cases} x_1 = x_r; \\ x_2 = \dot{x}_r; \\ x_3 = x_s; \\ x_4 = \dot{x}_s. \end{cases}$$

As equações de estado são representadas tomando como base equações diferenciais dinâmicas do sistema e a definição das variáveis de estado:

$$\begin{cases} \dot{x}_1 = \dot{x}_r = x_2 = 0x_1 + 0x_2 + 0x_3 + 0x_4 + 0u(t); \\ \dot{x}_2 = \ddot{x}_r = -2x_r - 2\dot{x}_r + 1x_s + 1\dot{x}_s + 1u(t) = -2x_1 - 2x_2 + 1x_3 + 1x_4 + 1u(t); \\ \dot{x}_3 = \dot{x}_s = x_4 = 0x_1 + 0x_2 + 0x_3 + 1x_4 + 0u(t); \\ \dot{x}_4 = \ddot{x}_s = x_r + \dot{x}_r + x_s - \dot{x}_s + 0u(t) = x_1 + x_2 - x_3 - x_4 + 0u(t). \end{cases}$$

Supondo que a saída seja x_s (deslocamento do sensor), a equação de saída fica:

$$y = x_s = x_3 = 0x_1 + 0x_2 + 1x_3 + 0x_4 + 0u(t)$$

Na forma de matriz vetorial, a equação de estado, $\dot{x} = \mathbf{A}x + \mathbf{B}u$, torna-se:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -2 & -2 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & -1 & -1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u(t)$$

A equação de saída $y = \mathbf{C}x + \mathbf{D}u$ é a seguinte:

$$y = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u(t)$$

9.2 Desempenho da população inicial

O perfil da população inicial pode ser definido pelos valores da função objetivo apresentados na Figura 9.2. Percebe-se claramente uma diversidade genética, sendo baixa entre o vigésimo quinto e o trigésimo primeiro indivíduo e relativamente satisfatória para os outros indivíduos. Essas conclusões a respeito do processo de inicialização da população inicial, implicam em uma população satisfatória no processo de busca do indivíduo QR ótimo.

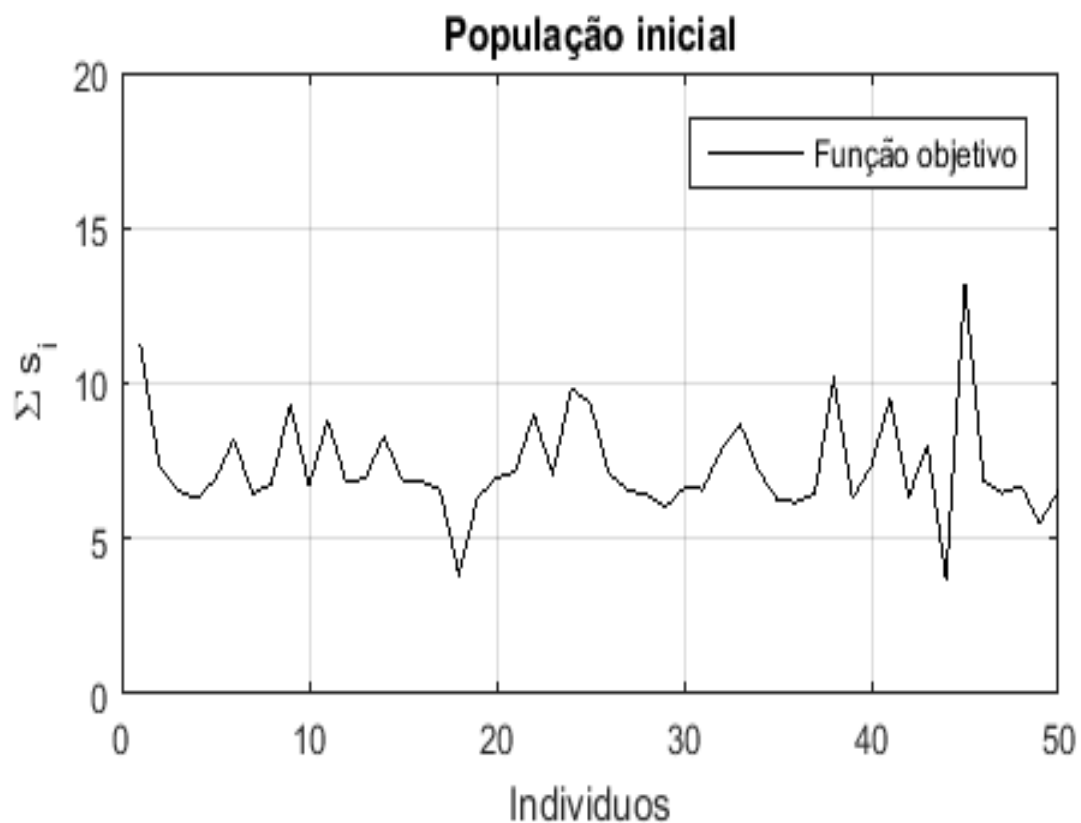


Figura 9.2: Sensibilidade total de cada indivíduo

Fonte: Autor

9.3 Desempenho da população final

Os valores da função objetivo para a população final, em comparação com a média da função objetivo para cada indivíduo da população, durante o processo de busca, são apresentados na Figura 9.3. Os valores médios de cada indivíduo da população final apresentaram melhorias praticamente em todos os indivíduos.

A busca realizada pelo AG encontrou cinco indivíduos viáveis que estão presentes na Tabela 9.1. Conseqüentemente, como esperado, os resultados computacionais para resposta ao sinal impulso mostraram uma ligeira diferença entre todas as soluções viáveis. Entre estes, destaca-se o indivíduo quadragésimo primeiro que gerou as matrizes de ponderação ideais:

$$Q_{LQR} = \begin{bmatrix} 41.436 & 5.494 & 4.069 & 5.120 \\ 5.94 & 44.103 & 9.148 & 3.102 \\ 4.069 & 9.148 & 29.682 & 6.951 \\ 5.123 & 3.102 & 9.951 & 22.067 \end{bmatrix} \quad R_{LQR} = \begin{bmatrix} 15.225 & 2.517 \\ 2.517 & 11.568 \end{bmatrix} \quad (9.7)$$

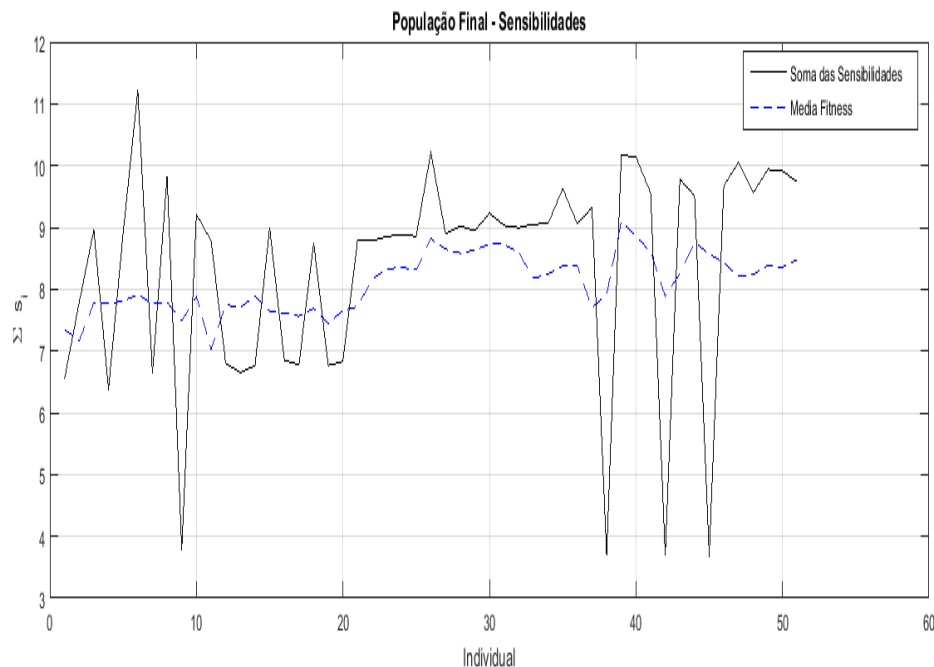


Figura 9.3: Evolução do processo - função objetivo média e melhor função objetivo das gerações

Fonte: Autor

Ger.	Indiv.	s_1	s_2	s_3	s_4	Σs_i
1	20	0,975	0,815	0,694	0,589	3,073
3	13	0,919	0,831	0,677	0,570	2,997
5	17	0,967	0,845	0,684	0,586	3,082
7	14	0,996	0,883	0,686	0,586	3,151
9	48	0,986	0,986	0,683	0,586	3,241

Tabela 9.1: Pop. final AG-sensibilidades e função objetivo dos indivíduos viáveis

Fonte: Autor

Os testes para avaliação entre os indivíduos estão presentes na Tabela 9.2. Nesta, todos os indivíduos viáveis estão próximos das restrições de sensibilidade e da função objetivo. Além disso, a condição de positividade das matrizes são mostradas em termos dos traços e autovalores. As matrizes de ponderação ótimas Q e R do melhor indivíduo factível, são dadas por pela equação (9.7). Na Tabela 9.2, nela estão os traços e os autovalores, confirmam a positividade das matrizes de ponderação. Essa positividade pode ser usada para orientar o processo de busca do AG, de modo que evite a formação indesejável de indivíduos.

Ger.	Indiv.	s_1	s_2	s_3	s_4	Σs_i
1	20	0,975	0,815	0,694	0,589	3,079
3	13	0,919	0,831	0,677	0,570	2,997
5	17	0,967	0,845	0,684	0,586	3,082
7	14	0,996	0,883	0,686	0,586	3,151
9	48	0,986	0,986	0,683	0,586	3,241

Tabela 9.2: Melhor indivíduo da população final-autovalores e traços das matrizes de ponderação

Fonte: Autor

O processo de busca do AG com a configuração de seus parâmetros produziu soluções muito próximas uma das outras, ou seja, o conjunto solução matricial QR apresentada pelo AG, tem apenas pequenas discrepâncias. As sensibilidades da população final e a média da função de fitness de cada geração é apresentada na Figura 9.4.

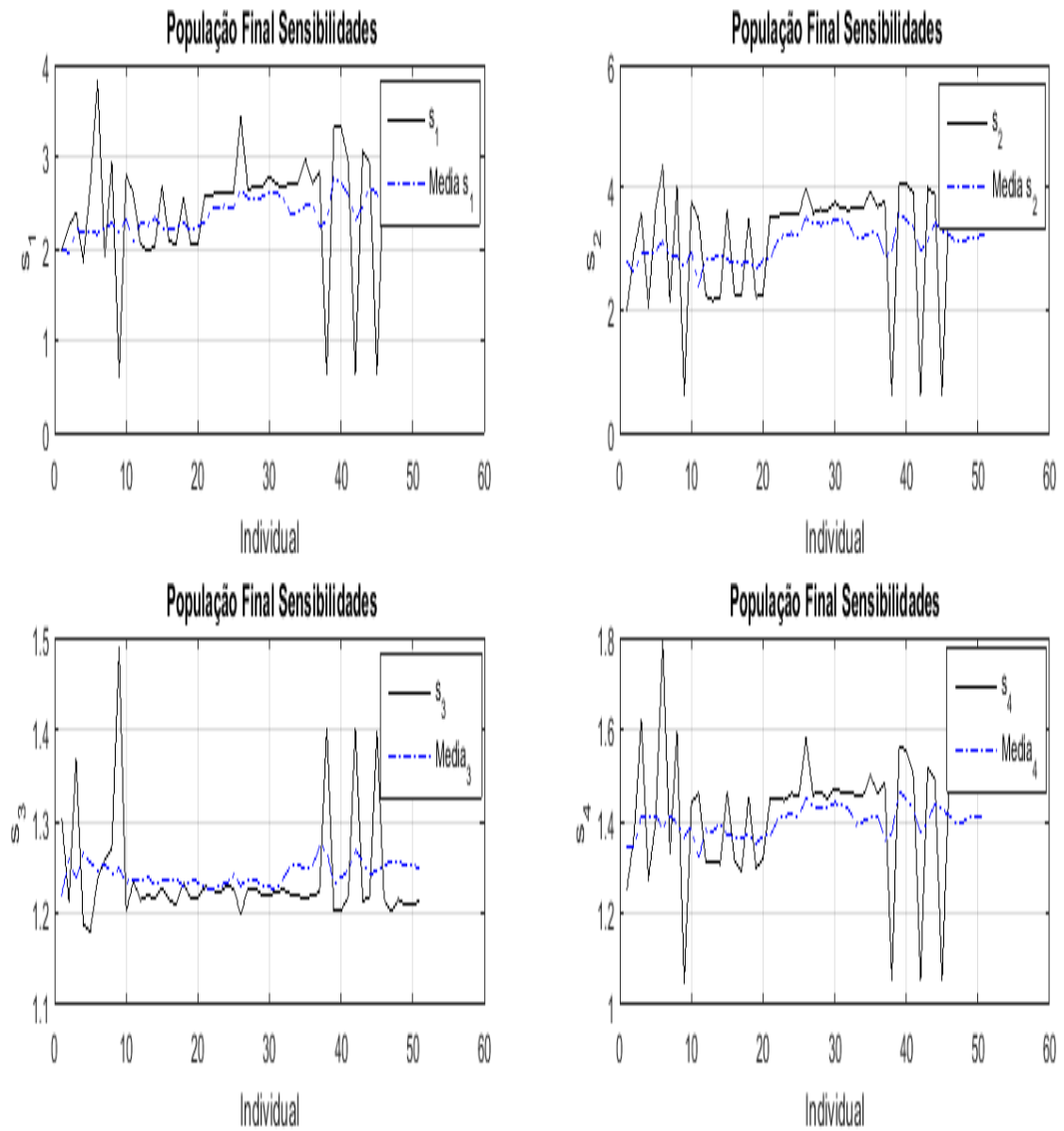


Figura 9.4: *Fitness* normalizada de cada geração e de cada indivíduo

Fonte: Autor

9.4 Análise da RNR na solução da EAR

Neste tópico está compreendido os métodos de ajuste de convergência e análise da *RNAR* para resolver a equação de *Riccati*. Os parâmetros de ajustes, do modelo neuronal, são suportados pela norma infinito e pela superfícies da função energia na solução da equação de *Riccati*. A norma infinito e a superfície da função energia, como funções dos parâmetros (pesos) de dimensionamento, são avaliadas para definir uma configuração, melhorando assim, a estabilidade de convergência e características de solvabilidade da *RNAR*. A norma infinito é uma forma usual de expressar a magnitude da matriz P , solução da *EAR* e é através dela que foi estabelecida uma faixa para os valores de η_p e η_z para se encontrar uma magnitude adequada. Na Figura 9.5 é apresentada a superfície da função energia da matriz de entrada U e na Figura 9.6 é apresentado a superfície da norma infinito, sendo que os intervalos de modificação para os parâmetros, são respectivamente, $[1000 \ 10000]$ e $[1 \ 10]$. Os pesos escolhidos dos parâmetros η_p e η_z são 1000 e 1, respectivamente. A Figura 9.7 mostra que as camadas da RNA têm comportamento assintótico após certo período, concordando com a estabilidade da rede. No entanto, na Figura 9.8 há instabilidade, tendo em vista que os valores da norma infinito estão fora dos atribuídos nos limites de η_p e η_z .

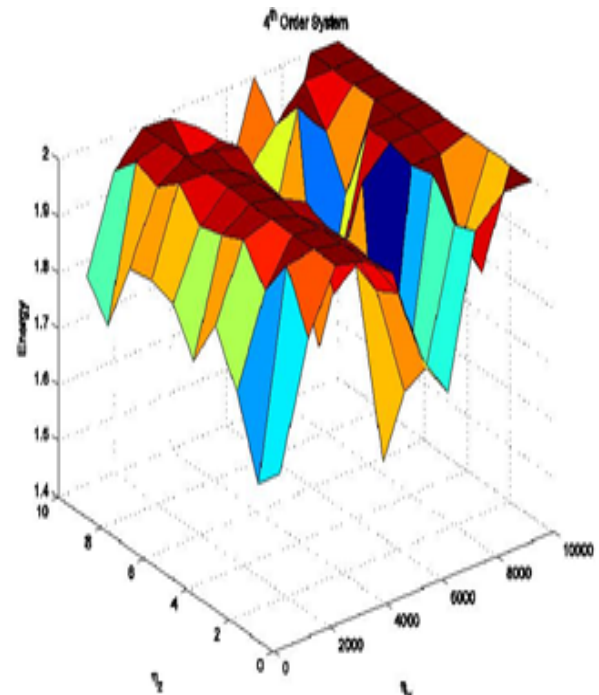
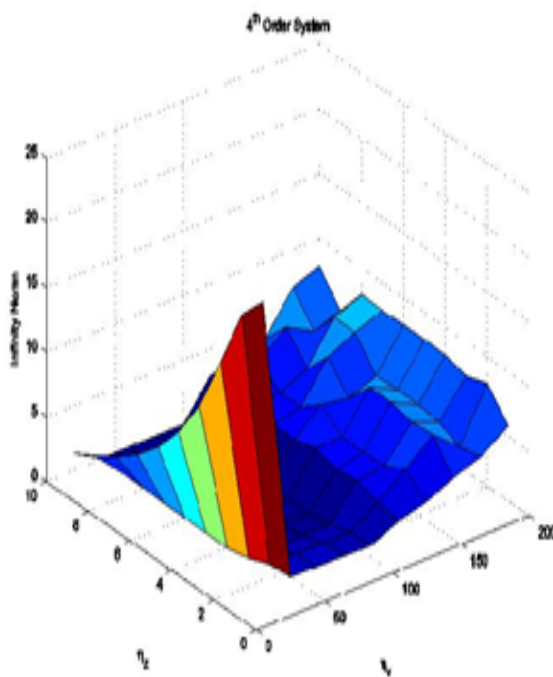


Figura 9.5: Superfície da função energia para $\eta_p = 1000 - 10000$ e $\eta_z = 1 - 10$

Figura 9.6: Superfície da norma infinito para $\eta_p = 1000 - 10000$ e $\eta_z = 1 - 10$

Fonte: Autor

Fonte: Autor

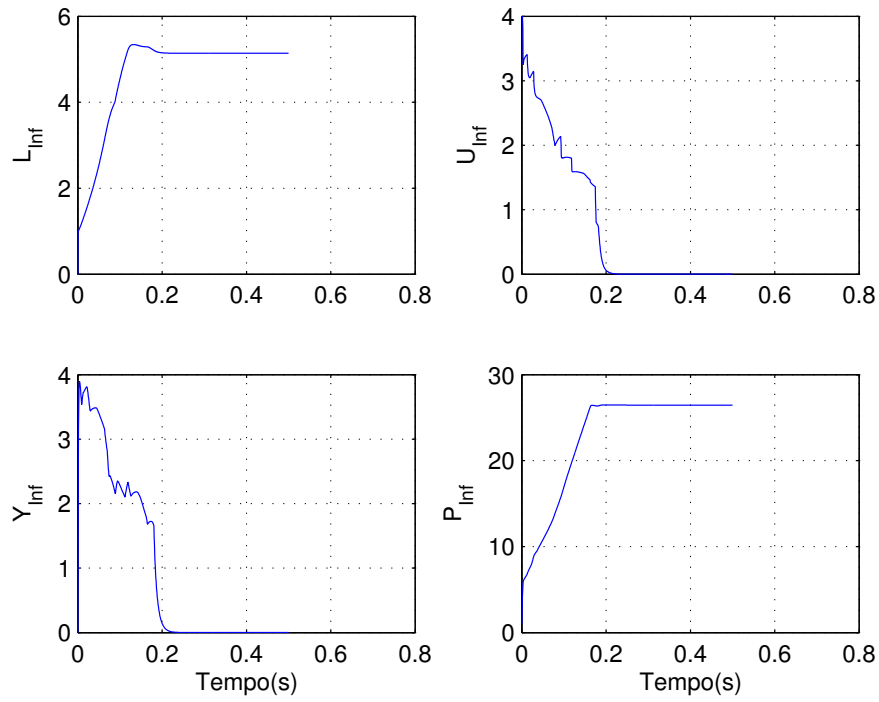


Figura 9.7: Comportamento das camadas da RNA

Fonte: Autor

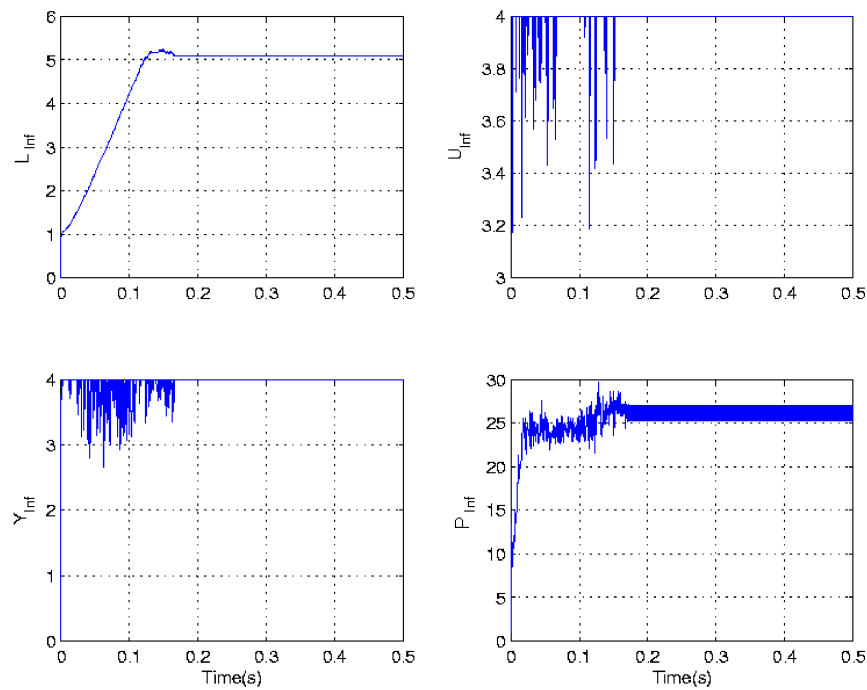


Figura 9.8: Comportamento das camadas da RNA

Fonte: Autor

A solução neuronal da equação de *Riccati*, fundamentado no modelo neuronal que resolve a estrutura de otimização que minimiza a função energia, camada de saída da RNR, Figura 7.1, é dada por

$$P_{RNR} = \begin{bmatrix} 17,884 & 2,228 & 2,533 & 1,838 \\ 2,228 & 19,229 & 2,760 & -2,245 \\ 2,533 & 2,760 & 13,596 & -4,100 \\ 1,838 & -2,245 & -4,100 & 10,859 \end{bmatrix} \quad (9.8)$$

9.5 Análise da filtragem de Kalman

No modelo em estudo, a Figura 9.9, exibe os resultados da Filtragem de *Kalman*. Nela verifica-se as trajetórias dos estados versus os estados estimados. Nota-se que, apesar da presença de ruídos, a ação reguladora é detectada em todos os estados. Para uma análise mais precisa, a Figura 9.10 ilustra o erro de estimação em cada estado. Observou-se que o Estado 4 apresenta o menor erro de estimação, enquanto que o maior erro de estimação foi verificado para o Estado 3.

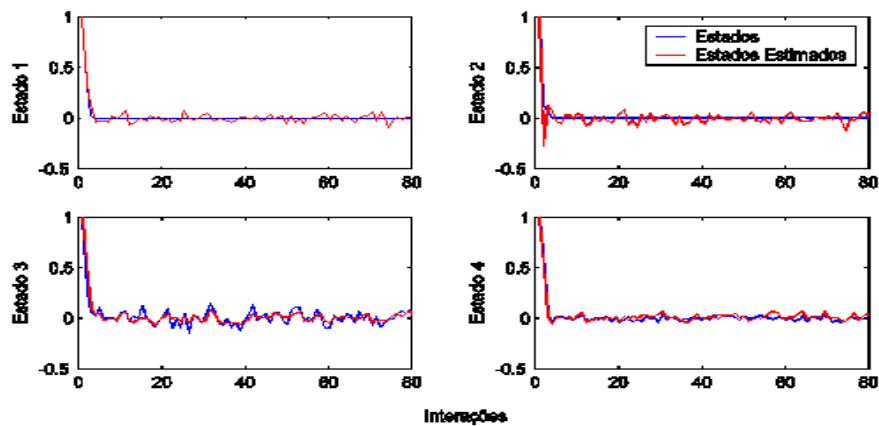


Figura 9.9: Trajetórias dos estados versus os estimados

Fonte: Autor

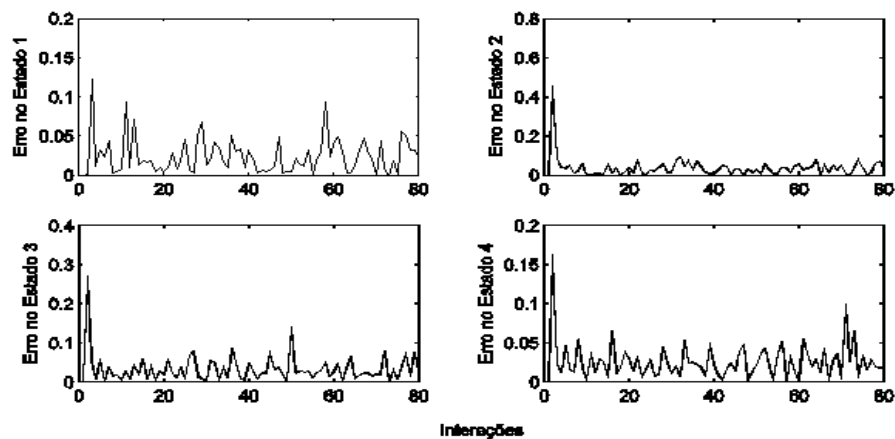


Figura 9.10: Erro de estimação para cada estado

Fonte: Autor

10 CONCLUSÃO

Neste trabalho foi utilizado um modelo hierárquico de algoritmos utilizando Rede Neural Recorrente (RNR) para a resolução da Equação Algébrica de *Riccati*, encontrando assim, as matrizes de ponderação Q e R para ajuste dos ganhos de ponderação. Toda a metodologia foi baseada no desenvolvimento dos controladores LQR e LQG utilizados no desenvolvimento de projetos de controle modernos. Tais controladores foram implementados utilizando uma hierarquia de controle baseados na **computação evolutiva**. Para o projeto em estudo, manipulador robótico, toda a sua implementação foi baseada com o uso de *algoritmo genético* e as *redes neurais artificiais* para realizar os ajustes de ganho de malha, garantindo assim, a robustez e a estabilidade do sistema.

A utilização do controle LQR , usado como uma aplicação particular, de um problema de otimização, caso do manipulador robótico, teve como caracterização a busca de vetores de entrada u que minimize a função de desempenho, J , da Equação (6.1). Este índice de desempenho representa um mapeamento dos espaços dos vetores de estados e de controle ponderados pelas matrizes Q e R , respectivamente. O método do LQG para a resolução do problema está baseado em um estimador de estado estocástico via Filtro de *Kalman* (FK), considerando-o como um esquema prático para a solução de problemas de controle, a partir de uma estimação feita com a ausência de sensores para a medição das variáveis de estado.

Baseados nessas implementações, isto é, controlador LQR e observador (FK), percebe-se claramente através do diagrama no domínio do tempo e da frequência para a estabilidade e desempenho robusto que o ganho de malha está alocado conforme as especificações do projeto. Portanto, os métodos utilizados para o ajuste são vistos como uma boa alternativa para a aplicação prática em tempo real.

10.1 Produção científica

- COBEM-2015 - doi://10.20906/CPS/COB-2015-0799 - ROBUSTNESS BARRIERS IN LQG/LTR CONTROLLER VIA HYBRID MODEL GENETIC-NEURAL IN ROBOTIC MANIPULATOR (19/10/2015);
- Journal of the Brazilian Society of Mechanical Sciences and Engineering - BMSE-D-16-01287 - ROBUSTNESS BARRIERS IN LQG/LTR CONTROLLER VIA HYBRID MODEL GENETIC-NEURAL IN ROBOTIC MANIPULATOR (05/12/2016);
- CONEM-2016 - CON-2016-0993 - UMA HIERARQUIA DE CONTROLADORES ÓTIMOS LQG/LTR E BARREIRA DE ROBUSTEZ VIA MODELO NEUROGENÉTICO EM DEXTEROUS HAND MASTER (16/07/2016);

10.2 Trabalhos futuros

Como perspectivas futuras propõem-se portanto:

- O estudo da complexidade do algoritmo desenvolvido para o Manipulador robótico e proposta de otimização;
- Verificar o desempenho do método de recuperação de malha na planta do equipamento Glide Slope pertencente ao *Instrument Landing System* (Sistema de Pouso por Instrumento);
- Implementar um manipulador robótico utilizando o Projeto de Controladores H_∞ e Algoritmo Genético a nível de comparação aos resultados encontrados neste trabalho.

REFERÊNCIAS

- [1] DORF, Richard C. *Sistemas de controle modernos*. Rio de Janeiro: LTC, 2014.
- [2] MAYR O. *The origins of feedback control*. Cambridge, Mass: MIT Press, 1970.
- [3] MAYR O. *The origins of feedback control: scientific american, v. 223, No. 4, October 1970, p. 110-118*. Cambridge, Mass: MIT Press, 1970.
- [4] MAYR O. *Feedback mechanisms in the historical collections of the national museum of history and technology*. Washigton, D.C: Smithsonian Institution Press, 1971.
- [5] OGATA, Katsuhiko. *Engenharia de controle moderno*. São Paulo: Pearson Prentice Hall, 2010.
- [6] DE JONG, K. *Adaptative system design: a genetic approach. IEEE transactions on systems, man, and cybernectics*. 1980. p. 566-574.
- [7] MICHALEWICZ, Z.; J.B. Krawczyk, M.; C.Z Janikow. *Genetic algorithms and optimal control*. In: PROCEEDING OF THE 29th CONFERENCE ON DECISION AND CONTROL, 1990.
- [8] CRUZ, José Jaime da. *Controle robusto multivariável*. São Paulo: EDUSP, 1996.
- [9] BARR, A.; FEIGENBAUM, E. *The Handbook of Artificial Intelligence, volume 1*. William Kaufmann, Inc., 1981.
- [10] CAMPOS, Mario Massa de; SAITO, Kaku. *Sistema inteligentes em controle e automação de processos*. Rio de Janeiro: Ciência Moderna, 2004.
- [11] DAVALO, E.; NAIN, P. *Des Réseaux de Neurones*. France: Eyrolles, 1992.
- [12] D'AZZO, Jonh J.; HOUPIS, Constantine H. *Feedback control system analysis and synthesis*. New York: McGrall-Hill, 1960.
- [13] CHIU, D.K; LEE, S. *Design and experimentation of a jump impact controller*. IEEE, 1997.

- [14] CHEN, Chi-Tsong. *Linear system theory and design*. New York: Oxford University press, Inc, 1999.
- [15] SKOGESTAD, S.; POSTLETHWAITE, I. *Multivariable feedback control: Analysis and design, v. 2*. Wiley New York, 2007.
- [16] HECHT-NIELSEN, R. *Neurocomputing*. Addison-Wesley publishing company, 1990.
- [17] LEWIS, F.L; VRABIE, D.L; SYRMOS, V.L. *Optmal control*. USA: John Wiley & Sons Inc., 2012.
- [18] GASPARYAN, Oleg. *Linear and nonlinear multivariable feedback control: A classical approach*. England: John Wiley Inc, 2008.
- [19] Kuo, B. J. *Automatic Control Systems*. New Jersey: 7th edn, Prentice Hall, 1995.
- [20] KAWAKERNAAK, H.; SIVAN, R. *Linear Optimal Control Systems*. New York: Wiley Interscience, 1972.
- [21] SIVANADAN, S.N.; DEEPA, S.N. *Introduction to genetic algorithms*. India: Springer-Verlag Berlin Heidelberg, 2008.
- [22] MÜLLER, B.; REINHARDT, J. *Neural networks: An introduction*. New York: Springer-Verlag Berlin Heidelberg, 1995.
- [23] ROSENBLATT, F. *Principles of Neurodynamics*. New York: Spartan Books, 1962.
- [24] MINSKY, M.; PAPERT, S. *Perceptrons*. Massachusetts-USA: MIT Press, 1969.
- [25] HAYKIN, Simon. *Neural networks and learning machines*. Canada: Prentice-Hall, 2009.
- [26] KOHOMEN, T. *Self-organization and associative memory*. Berlin: Springer-Verlag, 1988.
- [27] JACOB, F. *A lógica da vida: Uma história de hereditariedade*. Paris-FRA, Editions Gallimard, 1983.

- [28] HOLLAND, J.H. *Adaptation in natural and artificial systems*. Ann Arbor-Michigan: University Michigan Press, 1975.
- [29] GOLDBERG, David Edward. *Genetic algorithms in search, optimization, and machine learning*. Inc, USA: Addison-Wesley Publishing, 1989.
- [30] HAUPT, R.L.; HAUPT, S.E. *The continuous parameter genetic algorithm, practical genetic algorithm*. USA: Jon Wiley & Sons, Inc., 1198.
- [31] GROOVER, M. P.; WEISS, M.; NAGEL, R. N.; ODREY, N. G. *Industrial robotics: technology, programming, and applications*. São Paulo: McGraw-Hill Higher Education, 1986.
- [32] CARRARA, Valdemir. **Introdução à robótica industrial**.
Disponível em: < <http://urlib.net/8JMKD3MGP3W34P/3K5JPL8> >.
Acesso em: 13 abr. 2016.
- [33] ABREU, Ivanildo.; FONSECA, N. *Controle Inteligente LQR Neuro-Genético para Alocação de Autoestrutura em Sistemas Dinâmicos Multivariáveis: Tese de Doutorado*, 2008.
- [34] ABREU, Ivanildo.; FONSECA, N. *Neural-Genetic Synthesis for State-Space Controllers Based on Linear Quadratic Regulator Design for Eigenstructure Assignment: systems, man, and Cybernetics, part B; Cybernetics*. IEEE Transaction, 2010.
- [35] WANG, Jun.; WU, Guang. *A multilayer recurrent neural network for solving continuous-time algebraic Riccati equations*. Hong Kong, 1998.
- [36] DATTA, Biswa Nath. *Numerical methods for linear control systems*. St. Louis: Elsevier Science & Technology books, 2003.
- [37] ATHANS, Michael; FALB, L. Peter. *Optimal control: an introduction to the theory and its applications*, 2007.
- [38] GREFENSTETTE, J.J. *Optimization of control parameters for genetic algorithms*. IEEE transactions on systems, man, and cybernetics v.16(1) p. 122-128, 1980.

- [39] PARK, D.; A. Kandel; G. Langholz. *Genetic based new fuzzy reasoning models with application to fuzzy control*. *IEEE transactions on systems, man, and cybernetics*. 1994. v. 24(1). p. 39-47.
- [40] LAUB, A. J. *A schur method for solving algebraic riccati equations*. *IEEE Transactions on Automatic Control*. 1979. v.24(6) p. 913-921.
- [41] KAILATH, Thomas. *Linear systems*. First.ed. Inc.. Englewood Cliffs, N.J: Prentice-Hall, 1980.
- [42] KALMAN, R.E. *Contribution to the theory of optimal control*. 1960a. Bol.5. p. 102-119.
- [43] KALMAN, R.E. *On the general theory of control*. In: PROCEEDINGS OF FIRST IFAC CONGRESS. Butter Worth, Londor. 1960b. v.1 p. 481-491.
- [44] RISSANEN, J. *Control systems synthesis by analogue computer based on the generalized linear feedback concept*. In:.. *Proceedings of thr symposium on analog computation applied to the study of chemical processes*. International seminar. Brussels, Belgium, 1960. v.1 p. 1-13.
- [45] BLACKMORE, Perry A.; ROBERT R. Bitmead. *Duality between the discrete-time Kalman an LQ control law*. *IEEE Trasaction on Automatic control*, 1995.
- [46] SUMATHI, S; HAMSAPRIYA, T; SUREKHA, P. *Evolutionary intelligence: An introduction to theory and aplications with Matlab*. India: Springer-Verlag Berlin Heidelberg, 2008.
- [47] CRUZ, José Jaime da. *Contribuição ao estudo da robustez da estabilidade de reguladores não lineares multivariáveis*. Tese(Doutorado)- INPE, são José dos Campos-SP, 1988.
- [48] CASTRUCCI, Plínio de Lauro. *Controle automático*. Rio de Janeiro: LTC, 2011.
- [49] DOYLE, J.C; STEIN, G. *Robustness with observers*. In *IEEE Transactions on Automatic Control*, 1979.
- [50] FU, K.S. *Robotics:: Control, Sensing, Vision and Inteligence*. New York: McGrall-Hill, 1987.

- [51] GROOVER, M. P.; WEISS, M.; NAGEL, R. N.; ODREY, N. G. *Robótica: tecnologia e programação*. São Paulo: McGraw-Hill, 1989.
- [52] JOHNSON, M.A.; M.J. Grimble. *Recent trends in linear optimal quadratic multivariable control systems design*. IEEE- Review, 1987. v.134. p. 53-71.
- [53] MCCULLOCH, W.S; PITS, W. *A logical calculus of the ideas immanent in nervous activity*. Bulletin of Mathematica Biophysics Vol.52. No. 1/2. p.99-115. Chicago, 1990.
- [54] NETO, J.V.; ABREU, I. *A genetic algorithm convergence and models for eigenstructure assignment via linear quadratic regulator*. Revista IEEE, 2007.
- [55] NETO, João Viana da Fonseca et al. *Modelos e Convergência de um Algoritmo Genético para Alocação de Auto-estrutura via RLQ* .
- [56] RÊGO, Patricia Helena Moraes. *Convergência de Algoritmo Genético Hierárquico para Recuperação da Malha LQR por Controladores LQG/LTR: Dissertação de Mestrado*, 2007.
- [57] SERGI, F. *Design and characterization of a compact rotary series elastic actuator for knee assistance during overground walking*. Roma, 2012.
- [58] SILVA, Fábio Nogueira da. *Métodos Neurais para a solução da Equação Algébrica de Riccati e o LQR: Dissertação de Mestrado*, 2008.
- [59] XUE, Dingyü; CHEN, YangQuan; ATHERTON, Derek P. *Linear feedback control: Analysis and design with Matlab*. Philadelphia: Siam, 2007.
- [60] W.J. Cook. *The international space station takes shape: US News and world report, december 7, 1998, p. 56-59*

A Apêndice - FLUXOGRAMA DO AG

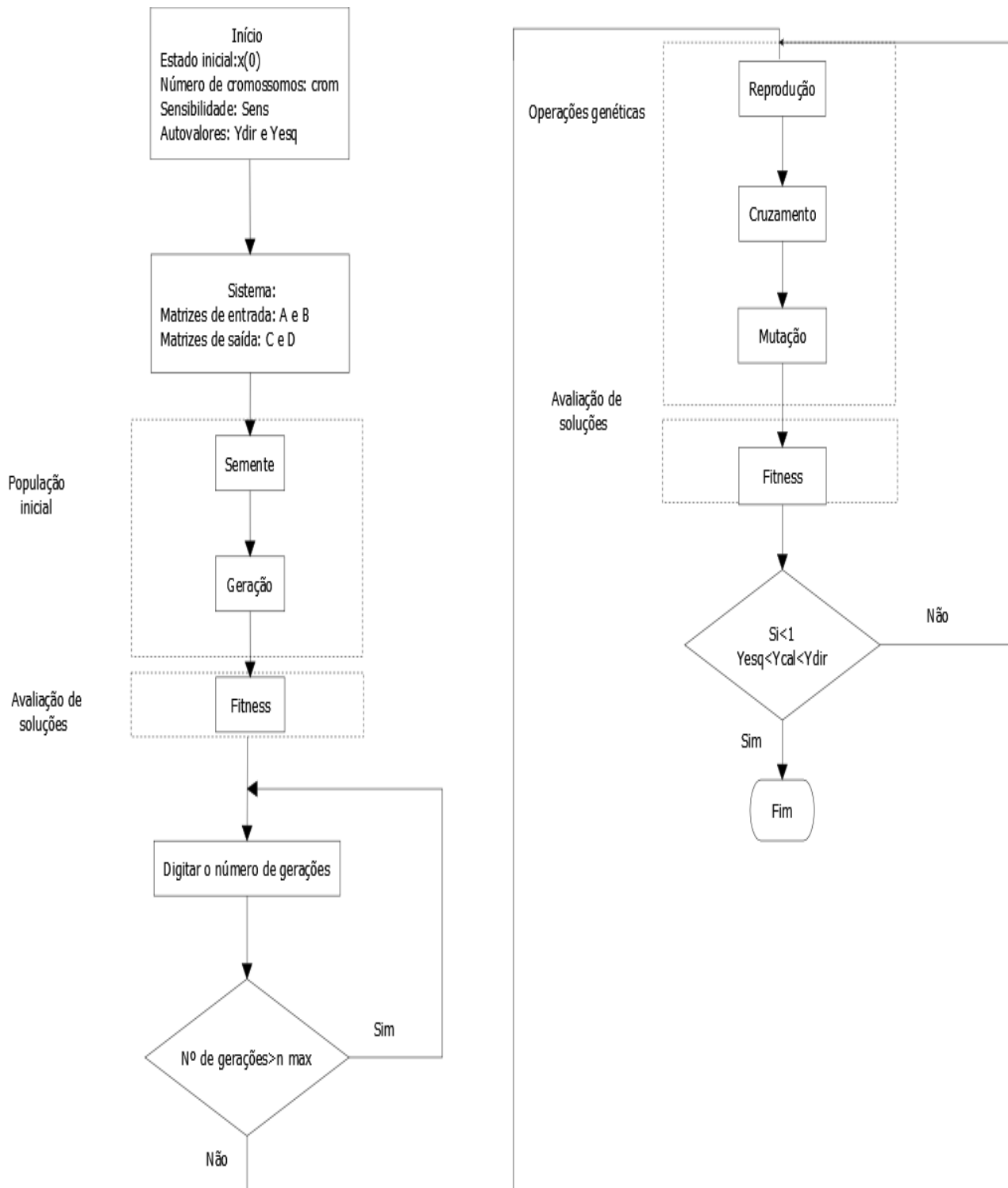


Figura A.1: Fluxograma do AG

Fonte: Adaptado de [46]

I Anexo - SISTEMAS REGULADORES QUADRÁTICOS ÓTIMOS

I.1 O problema do regulador quadrático ótimo

Dada a equação do sistema:

$$\dot{x} = \mathbf{A}x + \mathbf{B}u \quad (\text{I.1})$$

Permite determinar a matriz \mathbf{K} do vetor de controle ótimo

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t) \quad (\text{I.2})$$

para minimizar o índice de desempenho

$$J = \int_0^{\infty} (\mathbf{x}^* \mathbf{Q} \mathbf{x} + \mathbf{u}^* \mathbf{R} \mathbf{u}) dt \quad (\text{I.3})$$

onde \mathbf{Q} é uma matriz hermitiana definida positiva (ou semidefinida positiva) ou real simétrica e \mathbf{R} é uma matriz hermitiana definida positiva ou real simétrica. Note que o segundo do lado direito da Equação (I.3) representa o consumo de energia dos sinais de controle. As matrizes \mathbf{Q} e \mathbf{R} determinam a importância relativa do erro e o consumo dessa energia. Supondo que o vetor de controle $\mathbf{u}(t)$ não seja limitado. A lei de controle linear dada pela Equação (I.2) é a lei de controle ótimo. Portanto, se os elementos não conhecidos da matriz \mathbf{K} forem determinados para minimizar o índice de desempenho, então $\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t)$ será ótimo para qualquer estado inicial $\mathbf{x}(0)$. o diagrama de blocos mostrando a configuração ótima está indicado na Figura I.1.

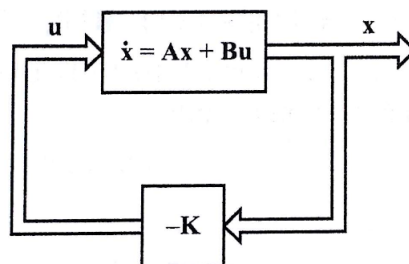


Figura I.1: Sistema regulador ótimo

substituindo a Equação (I.2) na (I.1), têm-se

$$\dot{x} = \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{K}\mathbf{x} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}$$

Substituindo a Equação (I.2) na Equação (I.3), têm-se

$$J = \int_0^{\infty} (\mathbf{x}^* \mathbf{Q} \mathbf{x} + \mathbf{x}^* \mathbf{K}^* \mathbf{R} \mathbf{K} \mathbf{x}) dt = \int_0^{\infty} \mathbf{x}^* (\mathbf{Q} + \mathbf{K}^* \mathbf{R} \mathbf{K}) \mathbf{x} dt$$

Fazendo

$$\mathbf{x}^* (\mathbf{Q} + \mathbf{K}^* \mathbf{R} \mathbf{K}) \mathbf{x} = -\frac{d}{dt} (\mathbf{x}^* \mathbf{P} \mathbf{x})$$

Onde \mathbf{P} é uma matriz hermitiana definida positiva ou simétrica real. Assim obtêm-se

$$\mathbf{x}^* (\mathbf{Q} + \mathbf{K}^* \mathbf{R} \mathbf{K}) \mathbf{x} = -\dot{\mathbf{x}}^* \mathbf{P} \mathbf{x} - \mathbf{x}^* \mathbf{P} \dot{\mathbf{x}} = -\mathbf{x}^* [(\mathbf{A} - \mathbf{B}\mathbf{K})^* \mathbf{P} + \mathbf{P} (\mathbf{A} - \mathbf{B}\mathbf{K})] \mathbf{x} \quad (\text{I.4})$$

Comparando ambos os lados da Equação (I.4) e notando-se que que essa deve ser verdadeira, qualquer que seja \mathbf{x} , têm-se necessariamente

$$(\mathbf{A} - \mathbf{B}\mathbf{K})^* \mathbf{P} + \mathbf{P} (\mathbf{A} - \mathbf{B}\mathbf{K}) = -(\mathbf{Q} + \mathbf{K}^* \mathbf{R} \mathbf{K}) \quad (\text{I.5})$$

Se $\mathbf{A} - \mathbf{B}\mathbf{K}$ for uma matriz estável, existirá uma matriz definida positiva \mathbf{P} que satisfaça a Equação (I.5).

Portanto, o procedimento consiste em determinar os elementos de \mathbf{P} a partir da Equação (I.5) e verificar se ela é definida positiva. Se o sistema for estável, sempre existirá uma matriz \mathbf{P} definida positiva que satisfaça a equação.

O índice de desempenho J pode ser calculado como

$$J = \int_0^{\infty} \mathbf{x}^* (\mathbf{Q} + \mathbf{K}^* \mathbf{R} \mathbf{K}) \mathbf{x} dt = -\mathbf{x}^* \mathbf{P} \mathbf{x} \Big|_0^{\infty} = -\mathbf{x}^*(\infty) \mathbf{P} \mathbf{x}(\infty) + \mathbf{x}^*(0) \mathbf{P} \mathbf{x}(0) \quad (\text{I.6})$$

Supondo que todos os autovalores de $\mathbf{A} - \mathbf{B}\mathbf{K}$ tenham partes reais negativas, têm-se $\mathbf{x}(\infty) \rightarrow 0$. Obtêm-se

$$J = \mathbf{x}^*(0) \mathbf{P} \mathbf{x}(0) \quad (\text{I.7})$$

Assim, o índice de desempenho J pode ser obtido em termos da condição inicial $\mathbf{x}(0)$ e \mathbf{P} .

Para a obter a solução do problema de controle quadrático ótimo, procede-se da seguinte maneira: Supondo que \mathbf{R} seja uma matriz hermitiana definida positiva ou real simétrica, pode-se escrever

$$\mathbf{R} = \mathbf{T}^* \mathbf{T}$$

onde \mathbf{T} é uma matriz não singular. Então, a Equação (I.5) pode ser escrita como

$$(\mathbf{A}^* - \mathbf{K}^* \mathbf{B}^*) \mathbf{P} + \mathbf{P} (\mathbf{A} - \mathbf{BK}) + \mathbf{Q} + \mathbf{K}^* \mathbf{T}^* \mathbf{TK} = 0$$

que pode ser reescrita como

$$\mathbf{A}^* \mathbf{P} + \mathbf{PA} + [\mathbf{TK} - (\mathbf{T}^*)^{-1} \mathbf{B}^* \mathbf{P}]^* [\mathbf{TK} - (\mathbf{T}^*)^{-1} \mathbf{B}^* \mathbf{P}] - \mathbf{PBR}^{-1} + \mathbf{Q} = 0$$

A minimização de J em relação a \mathbf{K} requer a minimização de

$$\mathbf{x}^* [\mathbf{TK} - (\mathbf{T}^*)^{-1} \mathbf{B}^* \mathbf{P}]^* [\mathbf{TK} - (\mathbf{T}^*)^{-1} \mathbf{B}^* \mathbf{P}] \mathbf{x} \quad (\text{I.8})$$

em relação a \mathbf{K} . Como a Equação (I.8) é não negativa, o mínimo ocorre quando ela é zero ou quando

$$\mathbf{TK} = (\mathbf{T}^*)^{-1} \mathbf{B}^* \mathbf{P}$$

portanto,

$$\mathbf{K} - \mathbf{T}^{-1} (\mathbf{T}^*)^{-1} \mathbf{B}^* \mathbf{P} = \mathbf{R}^{-1} \mathbf{B}^* \mathbf{P} \quad (\text{I.9})$$

A Equação (I.9) fornece a matriz ótima \mathbf{K} . Assim, a lei do controle ótimo do problema de controle quadrático ótimo, quando o índice de desempenho é dado pela a Equação (I.3), é linear e é dada por

$$\mathbf{u}(t) = -\mathbf{Kx}(t) = -\mathbf{R}^{-1} \mathbf{B}^* \mathbf{Px}(t)$$

A matriz \mathbf{P} na Equação (I.9) deve satisfazer a Equação (I.5) ou a seguinte equação reduzida

$$\mathbf{A}^* \mathbf{P} + \mathbf{PA} - \mathbf{PBR}^{-1} \mathbf{B}^* \mathbf{P} + \mathbf{Q} = 0 \quad (\text{I.10})$$

A Equação (II.1) é denominada equação matricial reduzida de Riccati. As etapas seguidas para um projeto podem ser expressas como

1. Resolução da Equação (II.1), equação reduzida de Riccati, para a matriz \mathbf{P} ;
2. Substituir essa matriz \mathbf{P} na Equação (I.9). A matriz \mathbf{K} resultante é a matriz ótima.

Se o índice de desempenho for dado em termos do vetor de saída em vez do vetor de estado, isto é,

$$J = \int_0^{\infty} (\mathbf{y}^* \mathbf{Qy} + \mathbf{u}^* \mathbf{Ru}) dt$$

então a expressão do índice pode ser modificada utilizando-se a equação de saída

$$\mathbf{y} = \mathbf{Cx}$$

para

$$J = \int_0^{\infty} (\mathbf{x}^* \mathbf{C}^* \mathbf{Qx} + \mathbf{u}^* \mathbf{Ru}) dt$$

Assim, as etapas do projeto, podem ser aplicadas para obter a matriz ótima \mathbf{K} .

II Anexo - ESTABILIDADE DOS SISTEMAS LTI

II.1 Estabilidade de entrada e saída de sistemas LTI

Considere um sistema linear invariante no tempo (LTI) **SISO** descrito por:

$$y(t) = \int_0^t g(t, \tau)u(\tau)d\tau = \int_0^t g(\tau)u(t - \tau)d\tau \quad (\text{II.1})$$

onde $g(t)$ é a resposta ao impulso ou saída excitada por um impulso aplicado a entrada em $t = 0$. Recorde-se que a equação dada por (II.1) descreve um sistema linear, invariante no tempo e causal. Além disso, o sistema deve ser inicialmente reduzido em $t = 0$.

Uma entrada $u(t)$ é dita limitada se $u(t)$ não cresce infinitamente positivo ou negativo ou, equivalentemente, existe uma constante u_m tal que

$$|u(t) \leq u_m < \infty|, \quad \forall t \geq 0$$

Um Sistema é dito **BIBO** estável (entrada-limitada, saída estável-limitada) se todas as entradas limitadas excita uma saída limitada. Esta estabilidade é definida para as respostas do estado zero e é aplicado somente se o sistemas são inicialmente reduzidos.

Teorema II.1.1. *Um sistema **SISO** descrito por (II.1) é **BIBO** estável se somente se $g(t)$ é absolutamente integrável em $[0, \infty]$ ou*

$$\int_0^{\infty} |g(t)| dt \leq M < \infty$$

para uma constante M

Prova Primeiro mostra-se que se $g(t)$ é absolutamente integrável, então toda entrada limitada excita uma saída limitada. Se $u(t)$ é uma entrada arbitrária com $|u(t)| \leq u_m < \infty, \forall t \geq 0$. Então

$$\begin{aligned} |y(t)| &= \left| \int_0^t g(\tau)u(t - \tau)d\tau \right| \leq \int_0^t |g(\tau)||u(t - \tau)| d\tau \\ &\leq u_m \int_0^{\infty} |g(\tau)| d\tau \leq u_m M \end{aligned}$$

Portanto a saída é limitada. Em seguida, é mostrado intuitivamente que se $g(t)$ é não absolutamente integrável, então o sistema **BIBO** é não estável. Se $g(t)$ é não absolutamente integrável, então existe um t_1 tal que

$$\int_0^{t_1} |g(\tau)| d\tau = \infty$$

Escolhendo

$$u(t_1 - \tau) = \begin{cases} 1 & \text{se } g(\tau) \geq 0 \\ -1 & \text{se } g(\tau) < 0 \end{cases}$$

Claramente u é limitado. No entanto, a saída excitada por esta entrada é igual

$$y(t_1) = \int_0^{t_1} g(\tau)u(t_1 - \tau)d\tau = \int_0^{t_1} |g(\tau)| d\tau = \infty$$

que é não limitado. Portanto o sistema é **BIBO NÃO ESTÁVEL**.