



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-7472-TDI/712

**USO DE ALGORITMOS GENÉTICOS NA CONSTRUÇÃO DE
CONTROLADOR NEBULOSO PARA O CONTROLE DE ATITUDE
DE UM SATÉLITE ARTIFICIAL DURANTE A FASE DE
APONTAMENTO**

Cláudio Correa

Dissertação de Mestrado em Computação Aplicada, orientada pela Dra. Sandra
Aparecida Sandri e pelo Dr. Luiz Carlos Gadelha de Souza, aprovada em
08 de março de 1999.

INPE
São José dos Campos
1999

Publicado por:

Coordenação de Ensino, Documentação e
Programas Especiais - CEP

Instituto Nacional de Pesquisas Espaciais - INPE

Caixa Postal 515

12201-970 - São José dos Campos - SP - Brasil

Fone: (012) 345.6911

Fax: (012) 345.6919

E-Mail: marciana@sid.inpe.br

- Solicita-se intercâmbio
- Exchange welcome
- Si sollecita intercambio
- Echange souhaité
- Mann bittet un Austausch
- Solicitamos intercambio
- Просим обмена
- 歡迎交換
- 出版物的交換不難

Publicação Exte. na - É permitida sua reprodução para interessados.

MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-7472-TDI/712

**USO DE ALGORITMOS GENÉTICOS NA CONSTRUÇÃO DE
CONTROLADOR NEBULOSO PARA O CONTROLE DE ATITUDE
DE UM SATÉLITE ARTIFICIAL DURANTE A FASE DE
APONTAMENTO**

Cláudio Correa

Dissertação de Mestrado em Computação Aplicada, orientada pela Dra. Sandra
Aparecida Sandri e pelo Dr. Luiz Carlos Gadelha de Souza, aprovada em
08 de março de 1999.

INPE
São José dos Campos
1999

681.3.019 : 629.7.062.2

CORREA, C.

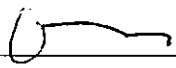
Uso de algoritmos genéticos na construção de controlador nebuloso para o controle de atitude de um satélite artificial durante a fase de apontamento / C. Correa – São José dos Campos: INPE, 1999.

124p. – (INPE-7472-TDI/712).

1.Controle de atitude. 2.Sistemas nebulosos. 3.Controladores. 4.Algoritmos genéticos. 5.Satélites artificiais.
I.Título.


Aprovado pela Banca Examinadora em
cumprimento a requisito exigido para a
obtenção do Título de **Mestre em**
Computação Aplicada.

Dr. Atair Rios Neto



Presidente

Dr^a Sandra Aparecida Sandri



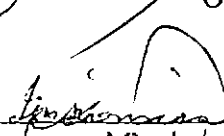
Orientadora

Dr. Luiz Carlos Gadelha de Souza



Orientador

Dr. Ijar Milagre da Fonseca



Membro da Banca

Dr. Antônio Augusto Rodrigues Coelho



Membro da Banca
Convidado

Candidato: Cláudio Correa

AGRADECIMENTOS

Agradeço aos meus orientadores, Dra. **Sandra Aparecida Sandri** e Dr. **Luiz Carlos Gadelha de Souza**, pela confiança e amparo no desenvolvimento deste trabalho.

Ao amigo Adenilson Roberto da Silva, por ceder os algoritmos de simulação do satélite e pelo auxílio incansável no entendimento destas rotinas.

À banca examinadora, que se propôs a analisar este trabalho e por contribuir com suas sugestões.

Aos professores dos cursos de Computação Aplicada e Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle, que contribuíram transmitindo seus conhecimentos e experiências.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico, pelo auxílio financeiro e ao Instituto Nacional de Pesquisas Espaciais pela oportunidade e apoio.

Aos amigos e colegas frequentadores do Laboratório de Computação do SPG, pelo companheirismo e apoio nesta jornada.

Aos funcionários do SIR, pela paciência, atenção e assistência.

RESUMO

Este trabalho apresenta a aplicação de algoritmos genéticos no projeto de controladores para o controle de atitude de um satélite artificial, durante a fase de apontamento para o Sol. Primeiramente, é apresentado um controlador PD para o modelo do satélite, cujos ganhos foram aprendidos por um algoritmo genético. Em seguida, são apresentados dois controladores nebulosos, cujos parâmetros foram obtidos utilizando-se um algoritmo genético com a mesma função de avaliação usada na obtenção do controlador PD. A função de avaliação utilizada é uma medida global do desempenho de cada solução em relação ao simulador existente para este modelo de satélite. O trabalho traz uma análise comparativa do desempenho dos controladores, em relação a um controlador PD desenvolvido pelo método da alocação dos pólos.

THE USE OF GENETIC ALGORITHMS ON A FUZZY CONTROLLER FOR A SATELLITE ATTITUDE CONTROL DURING THE POINTING PHASE

ABSTRACT

This work presents the application of genetic algorithms in the design of controllers for the attitude control of an artificial satellite, in the sun pointing phase. First, a PD controller is presented, whose gains were learned by a genetic algorithm. Then, two fuzzy controllers are presented, whose parameters were obtained using a genetic algorithm with the same fitness function employed for deriving of the PD controller. The fitness function used here is a global measure of the performance of each solution, in relation to an already existing simulator for this satellite model. This work also brings a comparative analysis of these controllers, in relation to a PD controller developed by the poles placement method.

SUMÁRIO

Pág.

LISTA DE FIGURAS

LISTA DE TABELAS

CAPÍTULO 1	INTRODUÇÃO	19
CAPÍTULO 2	ALGORITMOS GENÉTICOS	25
2.1	ESTÁGIOS DE UM ALGORITMO GENÉTICO	28
2.1.1	Operadores Genéticos	30
2.1.1.1	Esquemas de Seleção	30
2.1.1.2	Cruzamento	31
2.1.1.3	Mutação	32
2.1.2	Parâmetros Genéticos	33
CAPÍTULO 3	CONTROLADORES NEBULOSOS	35
3.1	DEFINIÇÕES GERAIS	36
3.2	OPERAÇÕES EM CONJUNTOS NEBULOSOS	38
3.2.1	Operadores de Intersecção, União e Complemento	38
3.2.2	Operadores de Implicação	40
3.3	PROJETO DE CONTROLADORES NEBULOSOS	42
3.3.1	Interface de “Fuzificação”	44
3.3.2	Base de Conhecimento	44
3.3.3	Procedimento de Inferência	45
3.3.3.1	Controle Nebuloso Clássico	46
3.3.3.1.1	Controladores Nebulosos Clássicos de Mamdani e de Larsen	47
3.3.3.2	Controle Nebuloso por Interpolação	48
3.3.3.2.1	Modelo de Tsukamoto	49
3.3.3.2.2	Modelo de Takagi-Sugeno	50
3.3.4	Interface de “Defuzificação”	51

CAPÍTULO 4	MODELO DO SATÉLITE	55
4.1	CARACTERÍSTICAS DO SATÉLITE	55
4.2	CONTROLADOR PD ORIGINAL	58
CAPÍTULO 5	PLATAFORMA COMPUTACIONAL	61
5.1	INTERFACE GRÁFICA	63
5.1.1	Módulos da Interface	64
5.1.2	Restrições da Interface Gráfica	68
5.2	TESTES OPERACIONAIS DA PLATAFORMA	69
5.2.1	Considerações para o Desenvolvimento da Função de Avaliação	69
5.2.2	Testes Iniciais de Funções de Avaliação	71
CAPÍTULO 6	PROJETO DE CONTROLADORES UTILIZANDO UM AG	79
6.1	CONDIÇÕES GERAIS DOS EXPERIMENTOS	79
6.2	FUNÇÃO DE AVALIAÇÃO	82
6.3	PROJETO DE CONTROLADOR PD	83
6.4	PROJETO DE CONTROLADORES NEBULOSOS	86
6.4.1	Parâmetros, Sintonização e Monitoração de Controladores Nebulosos	86
6.4.2	Controladores Nebulosos Projetados pelo AG	88
6.4.2.1	Controlador de Mamdani com 3 termos nebulosos de saída	90
6.4.2.2	Controlador de Mamdani com 5 termos nebulosos de saída	94
6.5	ANÁLISE DOS RESULTADOS	98
6.5.1	Avaliação da Robustez dos Controladores	99
6.5.2	Análise da Função de Avaliação	103
6.5.3	Avaliação da Variável de Saída (Torque) dos Controladores	105
CAPÍTULO 7	CONCLUSÕES E FUTUROS DESENVOLVIMENTOS	107

REFERÊNCIAS BIBLIOGRÁFICAS	114
----------------------------------	-----

LISTA DE FIGURAS

	Pág.
2.1 Estrutura básica de um Algoritmo Genético.	30
2.2 Ilustração de seleção pelo método da roleta.	31
2.3 Representação da operação de <i>crossover</i>	32
2.4 Representação do processo de mutação na posição $k = 4$	33
3.1 Conjunto “crisp” adolescente e conjunto nebuloso adolescente.	36
3.2 Cardinalidade, altura, suporte, núcleo e corte de nível	38
3.3 Ilustração das principais t – <i>normas</i>	39
3.4 Ilustração das principais t – <i>conormas</i>	40
3.5 Ilustração das implicações $I(h, b)$	41
3.6 Termos linguísticos que mapeiam a variável Velocidade	42
3.7 Estrutura de um Controlador Nebuloso.	43
3.8 Modelo clássico de Mamdani.	48
3.9 Modelo clássico de Larsen.	48
3.10 Modelo de interpolação de Tsukamoto.	50
3.11 Método de interpolação de Takagi-Sugeno.	51
3.12 Desfuzificação pelo Método do Centro de Gravidade	52
4.1 Ilustração do sistema de coordenadas do satélite	57
4.2 Torques perturbadores agindo sobre o modelo do satélite.	58
4.3 Erros de posição angular no apontamento do satélite com o controlador PD original.	60
4.4 Erros de velocidade angular desenvolvidos nos eixos X , Y e Z utili- zando controlador PD original.	60
5.1 Janela principal da interface gráfica.	64
5.2 Interfaces para informações do satélite, perturbações, órbita e lançamento	65
5.3 Interface de configuração dos parâmetros genéticos e limites de busca.	66
5.4 Seleção do controlador nebuloso e saidas gráficas	67

5.5	Resposta passo de um sistema de segunda ordem.	70
5.6	Erros de posição angular no apontamento do satélite com controlador otimizado pelo AG (Teste 1).	74
5.7	Erros de velocidade angular nos eixos X , Y e Z (Teste 1).	75
5.8	Erros de posição angular no apontamento do satélite com controlador otimizado pelo AG (Teste 2).	76
5.9	Erros de velocidade angular nos eixos X , Y e Z (Teste 2).	76
6.1	Torques perturbadores devidos ao dipolo magnético, pressão de ra- dição e arrasto atmosférico multiplicados por (β_i)	80
6.2	Erros angulares de referência.	81
6.3	Erros angulares com PD original na simulação de 1600 segundos. . .	82
6.4	Informações do AG na otimização do controlador PD.	84
6.5	Ganhos para o controlador PD, obtidos através do uso de um AG. . .	84
6.6	Erros obtidos com uso do controlador PD otimizado.	85
6.7	Erros angulares com PD otimizado na simulação de 1600 segundos. .	85
6.8	Conjuntos nebulosos para o mapeamento das conclusões das regras de um controlador nebuloso (ilustração).	89
6.9	Primeiro modelo do tipo Mamdani: Informações do processo de con- vergência do AG na aprendizagem do controlador.	90
6.10	Primeiro modelo do tipo Mamdani: Centros dos conjuntos nebulosos encontrados pelo AG.	91
6.11	Primeiro modelo do tipo Mamdani: Termos nebulosos das variáveis de entrada e saída encontrados com AG.	92
6.12	Primeiro modelo do tipo Mamdani: Superfície de controle da variável de saída torque em função das entradas	93
6.13	Primeiro modelo do tipo Mamdani: Erros angulares da simulação de uma órbita.	93
6.14	Primeiro modelo de Mamdani: Erros angulares da simulação de 1600 segundos.	94

6.15	Segundo modelo do tipo Mamdani: Informações do processo de convergência do AG na aprendizagem do controlador.	94
6.16	Segundo modelo do tipo Mamdani: Centros dos conjuntos nebulosos.	95
6.17	Segundo modelo do tipo Mamdani: Termos nebulosos das variáveis de entrada e saída encontrados com AG.	96
6.18	Segundo modelo do tipo Mamdani: Superfície de controle resultante do mapeamento da variável torque.	97
6.19	Segundo modelo do tipo Mamdani: Erros angulares da simulação de uma órbita.	97
6.20	Segundo modelo de Mamdani: Erros angulares da simulação de 1600 segundos.	98
6.21	Erros angulares com controlador PD original e variação de 30% no momento de inércia de Y.	101
6.22	Erros obtidos com controlador PD otimizado e variação de 30% no momento de inércia de Y.	102
6.23	Erros do primeiro modelo de Mamdani obtidos com variação de 30% no momento de inércia de Y.	102
6.24	Erros do segundo modelo de Mamdani obtidos com variação de 30% no momento de inércia de Y.	103
6.25	Torques de saída dos controladores.	105

LISTA DE TABELAS

Pág.

3.1	Principais T-normas e T-conormas duais	40
3.2	Principais operadores de implicação	41
5.1	Ganhos encontrados nos testes	77
6.1	Regras do primeiro CN do tipo Mamdani	91
6.2	Regras do segundo CN do tipo Mamdani	96
6.3	Resultados da integral dos erros e soma total desta integral nos três eixos	99
6.4	Integral dos erros, com variação de 30% no momento de inércia do eixo Y	100
6.5	Valores da função de avaliação	104

CAPÍTULO 1

INTRODUÇÃO

As pesquisas atuais mostram uma tendência tecnológica no desenvolvimento de pequenos satélites artificiais, que são utilizados comumente para garantir um meio rápido e simples de alcançar o espaço. Estes satélites têm sistemas altamente autônomos para o controle de atitude [1] [2], manobra e órbita [3] [4], e são desenvolvidos a baixo custo rapidamente [5] [6] [7].

Dois projetos de pequenos satélites atualmente em desenvolvimento no Instituto Nacional de Pesquisas Espaciais (INPE) são o Satélite de Aplicações Científicas (SACI 1) [8], satélite que têm seu sistema de controle baseado na estabilização por rotação, e o Franco Brasileiro que tem seu sistema de controle baseado na estabilização em três eixos. Este último utilizou-se de um controlador proporcional/derivativo (PD), cujos ganhos foram determinados através do método da alocação de pólos e cujas medidas dos sensores foram processadas por um filtro de Kalman [9] [10] [11]. É importante observar que em contraste à complexidade de um sistema de controle é imperativo que este tenha os custos de projeto e operacionais reduzidos [12] [13]. Portanto, torna-se fundamental a investigação de diferentes técnicas de desenvolvimento e implementação de um controlador. Tais inovações podem trazer a vantagem da versatilidade com respeito aos componentes de hardware a serem utilizados, no que diz respeito às interfaces entre controladores, sensores e atuadores.

A denominada lógica nebulosa é uma das tecnologias atuais bem sucedidas para o desenvolvimento de sistemas de controle sofisticados [14] [15] [16] [17] [18]. Com sua utilização, requerimentos complexos podem ser implementados em controladores simples, de fácil manutenção e baixo custo. O aumento de autonomia de satélites é um objeto de estudos contínuos para a melhoria de seu desempenho e redução de seus custos operacionais, assim, como a redução do consumo de combustível por ser este um recurso limitado. Uma abordagem para a obtenção desta autonomia é o uso

de controladores nebulosos, especialmente quando o modelo matemático está sujeito a incertezas [(?)] [20] [5] [21] [22] [23] [24].

Um controlador nebuloso compõe-se de um conjunto de regras de produção do tipo **Se** <premissa> **Então**<conclusão>, que definem ações de controle em função das diversas faixas de valores que as variáveis de estado do problema podem assumir [25] [26]. Estas faixas (usualmente mal definidas) de valores são modeladas por conjuntos nebulosos e denominados de termos linguísticos.

A maior dificuldade na criação de controladores nebulosos encontra-se na definição dos termos linguísticos e das regras. Uma das maneiras de sanar este problema consiste em lançar mão dos chamados modelos "neuro-fuzzy" [27] [28] [29] [30], em que estes parâmetros são aprendidos com a apresentação de pares (entrada, saída desejada) a uma rede neural cujos nós computam basicamente operadores de intersecção e união. Uma outra maneira de se aprender os parâmetros consiste no uso de algoritmos genéticos (AG) [31] [32] [33] [34] [35] [36] [37] [38] [39].

Algoritmos genéticos são estratégias de busca adaptativa, baseados em um modelo altamente abstrato da evolução biológica [40]. São utilizados em problemas de otimização, em que se busca, se não uma solução ótima, ao menos uma solução suficientemente adequada para um dado problema [41] [42].

Nestes algoritmos, uma população de indivíduos (soluções potenciais) sofrem uma série de transformações unárias (mutação) e de ordem mais alta (crossover). Estes indivíduos competem entre si pela sobrevivência: um esquema de seleção, que favorece os indivíduos mais aptos seleciona os que sofrerão transformações, dando origem à próxima geração. Depois de algumas gerações, o algoritmo converge e o melhor indivíduo representa uma solução próxima da ótima.

Algoritmos genéticos têm sido usados em muitas aplicações ligadas a controle, em particular, envolvendo o desenvolvimento de controladores nebulosos, tanto no exterior [43] [44] [32] [45] [34] [46] [47] [48] [49] [50] quanto no Brasil [51] [36] [39].

O objetivo deste trabalho é desenvolver controladores para a fase de apontamento de um satélite artificial a roda de reação, empregando algoritmos genéticos para sintonizar seus parâmetros. Neste sentido, foi desenvolvido primeiramente um algoritmo genético para aprender os ganhos de um controlador do tipo PD, o que permitiu montar e testar os experimentos com o simulador projetado em [9]. Em seguida, foi desenvolvido um algoritmo genético para aprender os parâmetros de um controlador nebuloso do tipo Mamdani [52] [53] [54] [55] [56].

Durante o desenvolvimento do trabalho, foram implementadas diversas funções de avaliação (função de “fitness”), dentre as quais selecionou-se a mais qualificada, conforme as características da resposta de controle. A função selecionada é uma medida global do desempenho de cada solução em relação ao simulador do satélite e utilizada nos algoritmos genéticos desenvolvidos tanto para o controlador PD quanto para o controlador nebuloso.

O projeto foi implementado em “personal computer” (PC) da série Pentium através do ambiente Matlab da MathWorks Inc, com a utilização dos pacotes relacionados a cada aspecto do problema [57] [58] [59] [60] [61] [62] [63] [64], incluindo um simulador desenvolvido em [9], que dispõe das características da órbita descrita pelo satélite.

Para proporcionar uma forma mais eficiente de introduzir as informações e possibilitar uma visualização imediata dos resultados, foi desenvolvida uma interface gráfica que faz a ligação entre os diversos módulos empregados no desenvolvimento deste trabalho.

Este documento é organizado da seguinte maneira:

- O Capítulo 2 apresenta os algoritmos genéticos, a ferramenta de busca utilizada neste trabalho. Além de se comentar as idéias que originaram os algoritmos genéticos e suas aplicações, são descritos os passos, as operações e outros fatores importantes no ciclo do processo de busca realizado por esta ferramenta;
- No Capítulo 3, são apresentadas as técnicas e paradigmas utilizados em con-

- Ao final, o Capítulo 7 apresenta as principais conclusões e considerações deste trabalho, discutindo-se algumas sugestões e propostas para definir condições e metas para trabalhos futuros.

trole nebuloso. O estudo evolui de acordo com a sequência das pesquisas desenvolvidas nesta área. Inicia-se com as definições e operações propostas para conjuntos nebulosos e estes princípios são então relacionados com a construção de controladores nebulosos. Na sequência, descreve-se a estrutura e as componentes do controle nebuloso e, ao final deste capítulo apresentam-se diversos modelos encontrados na literatura;

- O Capítulo 4 refere-se especificamente ao satélite, suas condições de projeto e operação. Apresentam-se informações das características do modelo e da sua dinâmica, com as especificações e requisitos do projeto, assim como o sistema de controle original desenvolvido em [9];
- No Capítulo 5, são apresentados primeiramente o ambiente de desenvolvimento e a interface gráfica. Esta última faz a ligação entre os diversos pacotes que compõem o ambiente, assim como também proporciona a visualização dos resultados nos experimentos. Os vários módulos da interface são comentados, ilustrados e as restrições desta ferramenta são discutidas. Concentram-se ainda neste capítulo, as discussões relevantes sobre os fatores críticos a serem tratados na elaboração da função de “fitness” utilizada pelo algoritmo genético. Descreve-se a finalidade do algoritmo genético no controlador PD, bem como as funções de “fitness” avaliadas. Ao final são feitas considerações sobre os primeiros experimentos, realizados no intuito de verificar o acoplamento das rotinas e a funcionalidade do algoritmo genético;
- No Capítulo 6, são apresentados os controladores desenvolvidos utilizando um algoritmo genético. São explorados os critérios adotados nos experimentos, as formas de projeto dos controladores PD e nebulosos, e os resultados obtidos em condições de operação. Os gráficos dos resultados auxiliam a verificação do desempenho de cada controlador e as soluções obtidas são comparadas à solução descrita em [9]. As análises dos resultados esclarecem pontos relevantes dos projetos e suas respostas;

CAPÍTULO 2

ALGORITMOS GENÉTICOS

As linhas de pesquisa em otimização distinguem dois importantes grupos de métodos de otimização, os determinísticos e os estocásticos. Os mais eficientes métodos determinísticos em geral precisam do cálculo de derivadas da função objetivo para determinar a direção de busca na pesquisa do ponto de mínimo. A solução obtida comumente é um ponto de mínimo local, próximo ao ponto de início, não garantindo assim o mínimo global.

Os métodos estocásticos, por sua vez, não necessitam do cálculo de derivadas e são capazes de encontrar a solução global, com a desvantagem de que normalmente o número de avaliações da função objetivo é superior ao requerido pelos métodos determinísticos.

Os Algoritmos Genéticos (AG) constituem um método de otimização classificado como estocástico. Baseiam-se na analogia entre otimização e a evolução natural das espécies, combinando os conceitos de adaptação seletiva e sobrevivência dos indivíduos mais capazes [39].

Trata-se de um dos paradigmas de uma nova linha de pesquisa alternativa em Ciência da Computação, conhecida como Computação Evolutiva (CE), também considerada uma técnica emergente de Inteligência Artificial (IA) por suas características particulares. Entre outras possibilidades, um AG permite interpretar processos de aprendizado, quando aplicado a populações de indivíduos que podem ter o seu desempenho verificado por meio de uma função de avaliação específica [65] [66].

A moderna teoria da evolução combina a genética e as idéias sobre a seleção natural, criando o princípio básico de Genética Populacional: a variabilidade entre indivíduos em uma população de organismos que se reproduzem sexualmente é produzida pela mutação e pela recombinação genética.

Nos anos 50 e 60, muitos biólogos começaram a desenvolver simulações computacionais de sistemas genéticos. Entretanto, foi John Holland que começou a desenvolver as primeiras pesquisas no tema e em 1975 publicou o livro “Adaptation in Natural and Artificial Systems”, considerado uma das mais importantes referências para Algoritmos Genéticos [67]. Desde então, estes algoritmos vêm sendo aplicados com sucesso nos mais diversos problemas de otimização e aprendizado de máquina [68].

Esses algoritmos usam uma base conceitual comum para a evolução de estruturas individuais por meio de operadores de *seleção*, *mutação* e *cruzamento*, que dependem da percepção do desempenho das estruturas individuais, no sentido definido pelo seu ambiente, isto é, operam sobre uma população de estruturas que evoluem de acordo com leis de seleção e operadores genéticos.

O indivíduo na população recebe uma medida conforme sua adequação ao ambiente e maior facilidade de reprodução é dada aos indivíduos mais aptos. Além disso, um processo de mutação introduz uma leve perturbação em alguns dos indivíduos de uma geração, o que tende a levar a um aprimoramento das gerações subsequentes [66].

Assim, um algoritmo genético é um modelo de aprendizado de máquina que deriva o seu comportamento de uma metáfora do processo evolutivo natural. No princípio, uma população finita de candidatos a soluções (cromossomas) são gerados, usualmente de maneira aleatória ou através de um processo heurístico.

Os parâmetros envolvidos na solução do problema são codificados nos cromossomas. Em essência, estes cromossomas são formados por um conjunto de “strings” de caracteres, análogos aos cromossomas de quatro bases (timina, guanina, adenosina e citosina) existentes no DNA (ácido desóxi-ribonucléico) natural. Os cromossomas criados são submetidos a um processo de evolução.

Por sua vez, a evolução na natureza não é um processo dirigido, não existindo evidências de que o objetivo da evolução seja, por exemplo, a produção de populações humanas ou equivalentes. O processo de evolução natural ocorre com diferentes

indivíduos competindo pelos recursos disponíveis em seu ambiente. Assim alguns indivíduos se sobressaem aos outros, aumentando a probabilidade de propagação do seu material genético.

Como na natureza, onde a codificação de informação genética (genoma) permite reprodução assexuada com descendentes geneticamente idênticos aos progenitores (clonagem), também nos algoritmos genéticos não se descarta a ocorrência de situações desse tipo. No entanto, é mais comum o uso do modelo sexuado, em que pode-se ter descendentes com genes extremamente diferentes dos progenitores comparados individualmente, porém, herdando as características gerais da espécie.

A seleção para acasalamento resulta da adequação do indivíduo e sua habilidade de competir por recursos disponíveis no ambiente. A operação de cruzamento (“crossover”) é responsável pela recombinação das informações dos pais durante a reprodução, permitindo que as próximas gerações herdem as características da espécie. A seleção e o cruzamento, no processo de reprodução dessa natureza, são os que mais contribuem para a evolução de uma população.

Nos algoritmos genéticos pode-se empregar uma função simples para medir a adequação de cada indivíduo e selecioná-los probabilisticamente para participação nas operações genéticas. Alternativamente, algumas implementações empregam um modelo onde certos indivíduos, escolhidos aleatoriamente em um subgrupo, competem entre si e os de melhor desempenho são selecionados.

A mutação também é relevante no processo evolutivo, porém não deve ser empregada de maneira a descaracterizar o algoritmo genético como uma busca randômica para a solução de um problema. Apesar do fato que algoritmos genéticos usam processos estocásticos, observa-se facilmente a solução não-aleatória [66].

O modelo genético computacional pode ser implementado através de vetores de bits ou caracteres que representam os cromossomos e as operações genéticas são implementadas por operadores simples de manipulação de bits. Em geral, os algoritmos genéticos utilizam cadeias de caracteres (“strings”) de tamanho fixo, ao contrário

do que ocorre na área de Programação Genética, que não fixa a representação e tipicamente não há a codificação dos problemas [69] [66].

2.1 ESTÁGIOS DE UM ALGORITMO GENÉTICO

Descreve-se em seguida um algoritmo genético. Maiores detalhes podem ser encontrados em [69] [35] [66] [39].

Um ciclo de um algoritmo genético consiste dos estágios de criação de uma população de soluções potenciais codificadas (indivíduos), avaliação dos indivíduos, seleção dos indivíduos mais capazes e geração de uma nova população através de manipulações genéticas, como cruzamento e mutação.

Em geral, um AG modela uma solução possível para o problema na forma de um vetor com um número fixo de posições (cromossoma), onde cada posição assume um valor em $\{0,1\}$.

Inicialmente, o algoritmo genético utiliza uma população inicial de cromossomas, oriunda de um processo aleatório. Uma função, a denominada função objetivo (“fitness”), verifica o quanto a solução representada por cada um dos cromossomas da população em análise se aproxima daquela ideal, medindo os efeitos que ela causa ao ser aplicada como solução do problema. Uma nova população é então gerada, através dos operadores genéticos. Normalmente, substitui-se a população progenitora pela gerada, o que mantém o tamanho da população fixo.

Os principais operadores são a seleção, que escolhe alguns cromossomas como aptos a passarem suas informações para a próxima geração, o cruzamento que toma dois cromossomas selecionados e casando-os aleatoriamente gera filhos que tem características de ambos os pais, e a mutação, que introduz uma pequena modificação em um cromossoma selecionado. As chances de um cromossoma ser selecionado são maiores quanto melhor é seu desempenho em relação a uma função objetivo. Usando-se esse conceito de adequação, as operações genéticas são executadas no sentido do aprimoramento da população.

Repete-se o ciclo usando a nova população e cada nova iteração neste ciclo dá origem a uma geração. Essa evolução ocorre até que alguma condição de parada seja satisfeita, como por exemplo quando um número máximo de gerações ou tempo limite é atingido, ou quando é encontrada uma solução estável.

Em seguida apresenta-se o pseudo-código de um AG. Durante esse processo, os melhores indivíduos, assim como alguns dados estatísticos, podem ser coletados e armazenados para avaliação.

Procedimento_AG

```

{t = 0;

  inicia_população(P, t);

  avaliação(P, t);

  repita até(t = d);

    {t = t + 1;

      seleção_dos_pais(P, t);

      cruzamento(P, t);

      mutação(P, t);

      avaliação(P, t);

      sobrevivem(P, t);

    }

}
```

onde t = tempo atual, d = tempo determinado para finalizar o algoritmo; e P = população.

A Figura 2.1 ilustra o esquema do ciclo típico de um algoritmo genético com a se-

quência das operações realizadas sobre a composição de uma geração, como proposto por [70].

ESTRUTURA DE UM ALGORITMO GENÉTICO

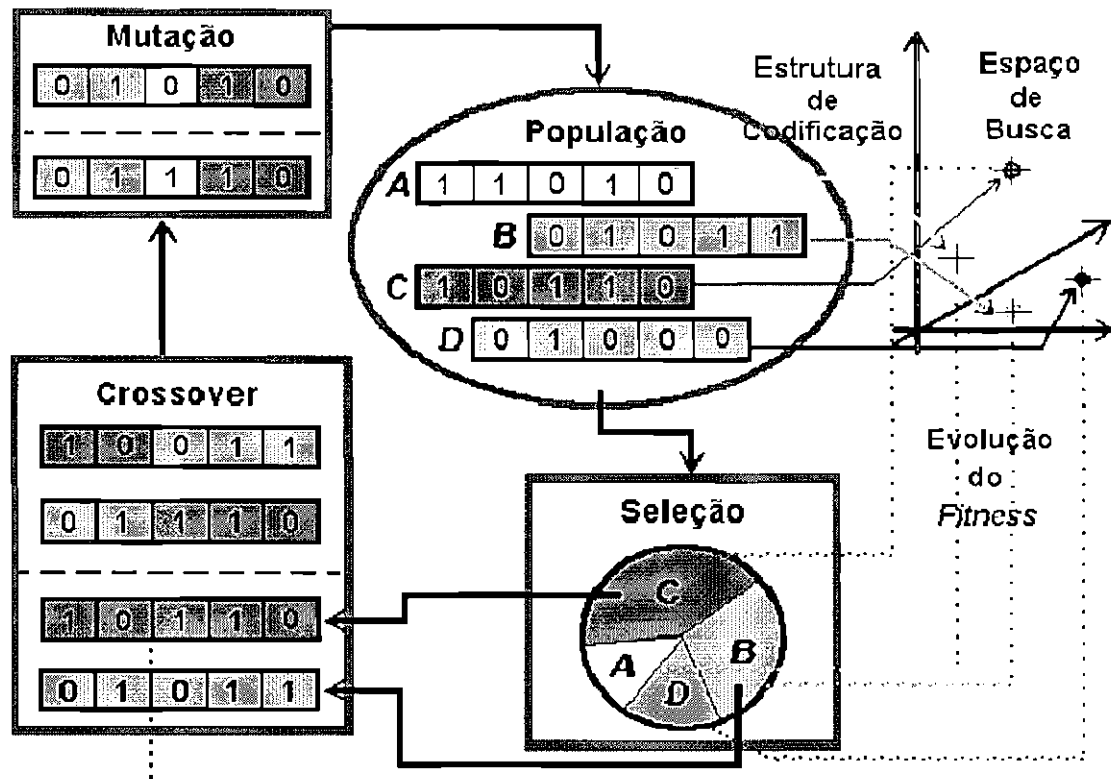


Fig. 2.1 - Estrutura básica de um Algoritmo Genético.

FONTE: Adaptada de Hoffmann (1997).

2.1.1 Operadores Genéticos

2.1.1.1 Esquemas de Seleção

Os esquemas de seleção fornecem um mecanismo para a escolha dos indivíduos mais capacitados, identificando os indivíduos mais aptos a passar informação genética para a próxima geração. O mais comum é a utilização de uma seleção proporcional, como o método da roleta.

Na seleção proporcional, um indivíduo com valor de aptidão f_i , gera por volta de f_i/\bar{f}

descendentes, onde \bar{f} é a média dos valores da função de adaptação da população. Deste modo, o indivíduo com valor de aptidão f_i maior que a média \bar{f} da população, usualmente gera mais descendentes que os de valor f_i menor que a média \bar{f} .

Na roleta, cada indivíduo corresponde a um setor circular de ângulo $2\pi \cdot (f_i/\bar{f})$. Um número aleatório entre 0 e 2π determina qual o setor e o indivíduo correspondente, que pela própria natureza da roleta, tem a tendência de manter o carácter proporcional nessa determinação.

A Figura 2.2 ilustra a configuração de uma roleta, com 5 cromossomas c_i e sua probabilidade de seleção.

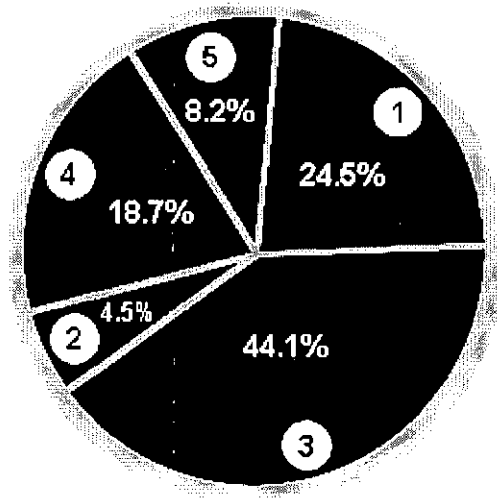


Fig. 2.2 - Ilustração de seleção pelo método da roleta.

2.1.1.2 Cruzamento

O principal mecanismo responsável pela evolução é o cruzamento. Entre os indivíduos selecionados, dois são escolhidos para se cruzarem. Esse cruzamento depende de uma probabilidade p_c e a decisão para executá-la é tomada usando-se um gerador de números aleatórios no intervalo $[0, 1]$. Somente se o número gerado for superior a p_c o cruzamento será realizado.

Para realizar o cruzamento, outro número aleatório k ($1 \leq k \leq (l - 1)$) é utilizado para determinar a posição de corte na cadeia de bits (cromossoma) de tamanho l . Então, o material genético à direita desse ponto é permutado entre os cromossomas selecionados. A Figura 2.3 ilustra este procedimento para dois cromossomas de tamanho 8 ($l = 8$ bits), com o ponto de corte na posição $k = 4$.

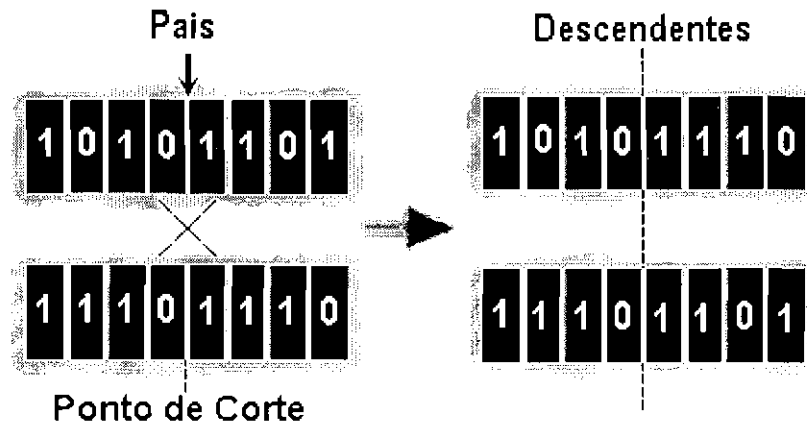


Fig. 2.3 - Representação da operação de *crossover*.

Como resultado, têm-se duas novas possíveis soluções com características herdadas de seus genitores, constituindo, assim, um dos principais mecanismos para obtenção de novos pontos do espaço de busca, onde se efetuará os testes.

2.1.1.3 Mutação

A mutação altera bits numa cadeia de caracteres com a probabilidade p_m . Esta operação introduz novas informações, conferindo ao processo de busca da solução uma característica que o cruzamento por si só não consegue. Isto permite vasculhar novos pontos no espaço de busca, aumentando a probabilidade de se encontrar o ótimo global.

Esse processo é realizado bit a bit. Para cada bit considerado (bit da posição k), gera-se um número aleatório no intervalo $[0, 1]$. A operação é executada se esse número é superior à probabilidade p_m , e então, o valor do bit é alterado (Figura 2.4).



Fig. 2.4 - Representação do processo de mutação na posição $k = 4$.

2.1.2 Parâmetros Genéticos

Os parâmetros genéticos são entidades que determinam o desempenho do algoritmo genético:

- **Tamanho da população:** Relaciona-se com a eficiência do algoritmo genético. Uma população pequena acarreta em soluções pouco satisfatórias. Já populações grandes previnem uma convergência prematura para soluções locais, porém, demandam maior esforço computacional e mais tempo devido ao número de avaliações da função objetivo;
- **Probabilidade de cruzamento (p_c):** É um valor percentual do número de indivíduos que realizam o cruzamento em relação ao total de indivíduos da população. Valores altos de p_c provocam a eliminação de indivíduos com bom desempenho antes mesmo que a seleção possa produzir aperfeiçoamentos;

Contudo, um valor baixo para p_c dificulta a convergência, tornando-a mais lenta devido à pouca exploração das características genéticas. Em geral, esse valor varia entre 0.5 e 0.95 [39] [66];

- **Probabilidade de mutação (p_m):** É o valor percentual do número de bits escolhidos aleatoriamente para serem alterados em relação ao número

total de bits da população. A mutação é responsável pela diversidade da população, assim, um percentual baixo restringe as características dessa população, enquanto que p_m elevada resulta numa busca essencialmente randômica. Este valor, normalmente, varia entre 0.001 e 0.05 [39].

Existem diversas áreas utilizando algoritmos genéticos e na solução de problemas de otimização multidimensional, onde a “string” de caracteres representando o cromossoma pode ser usada para codificar os valores dos diferentes parâmetros envolvidos na tarefa, um em cada gene componente do cromossoma [66] [69] [71] [35]. Discussões mais abrangentes e detalhadas sobre algoritmos genéticos podem ser encontradas em [69] e [71].

CAPÍTULO 3

CONTROLADORES NEBULOSOS

Os controladores nebulosos baseiam-se na teoria dos conjuntos nebulosos [72], que foi desenvolvida a partir de 1965 com os trabalhos de Lotfi Zadeh, professor na Universidade da Califórnia em Berkeley.

As técnicas de controle nebuloso originaram-se com as pesquisas e projetos de E. H. Mamdani [54] [56] e ganharam espaço como área de estudo em diversas instituições de ensino, pesquisa e desenvolvimento do mundo, sendo até hoje uma importante aplicação da teoria dos conjuntos nebulosos.

O termo “lógica nebulosa” é usualmente empregado na área de controle para denominar a modelagem de informação “nebulosa” e os mecanismos de inferência que permitem deduzir informações.

Formalmente, um conjunto nebuloso A do universo de discurso Ω é definido por uma função de pertinência $\mu_A : \Omega \rightarrow [0, 1]$. Essa função associa a cada elemento x de Ω o grau $\mu_A(x)$, com o qual x pertence a A [72]. A função de pertinência $\mu_A(x)$ indica o grau de compatibilidade entre x e o conceito expresso por A :

- $\mu_A(x) = 1$, x é completamente compatível com A ;
- $\mu_A(x) = 0$, x é completamente incompatível com A ;
- $0 < \mu_A(x) < 1$, x é parcialmente compatível com A , com grau $\mu_A(x)$;

Um conjunto A da teoria dos conjuntos clássica pode ser visto como um conjunto nebuloso específico, denominado usualmente de “crisp”, para o qual $\mu_A : \Omega \rightarrow \{0, 1\}$, ou seja, a pertinência é do tipo “tudo ou nada”, “sim ou não”, e não gradual como para os conjuntos nebulosos.

A diferença entre estes conceitos em relação à variável idade é ilustrada na Figura 3.1a e na Figura 3.1b, que descrevem respectivamente a representação do conceito “adolescente” através de um conjunto “crisp” e de um conjunto nebuloso.

O conjunto “crisp” A não exprime completamente o conceito de “adolescente”, pois uma pessoa com 12 anos e 11 meses seria considerada completamente incompatível com este conceito. Na verdade, qualquer intervalo “crisp” que se tome para representar este conceito é arbitrário.

Já o conjunto nebuloso B permite exprimir que qualquer pessoa com idade entre 13 e 17 anos é um adolescente, acima de 19 ou abaixo de 11 não é considerado um adolescente, e no intervalo $[11, 13]$ (respectivamente $[17, 19]$) é considerado tanto adolescente quanto mais próxima de 13 (respectivamente de 17) é sua idade.

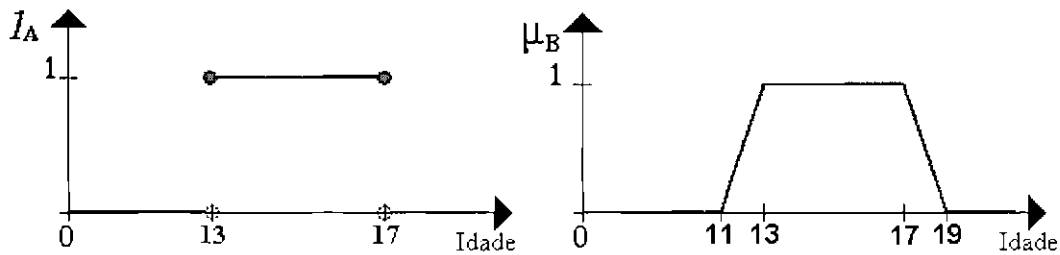


Fig. 3.1 - a) Função característica do conjunto “crisp” adolescente. b) Função trapezoidal característica do conjunto nebuloso adolescente.

3.1 DEFINIÇÕES GERAIS

A cardinalidade de um conjunto nebuloso A é expressa como:

- Para Ω discreto

$$|A| = \sum_{x \in \Omega} \mu_A(x) \quad (3.1)$$

- Para Ω contínuo

$$|A| = \int_{\Omega} \mu_A(x) \quad (3.2)$$

Pode-se obter a representação aproximada de um conjunto nebuloso A em Ω através de conjuntos “crisp” em Ω . Estes subconjuntos, denotados por A_{α} e denominados de cortes de nível ou cortes- α , são definidos como:

$$A_\alpha = \{x \in \Omega / \mu_A(x) \geq \alpha\} \quad (3.3)$$

Os casos extremos destes conjuntos são o suporte de A , denotado como $Su(A)$, que agrupa elementos de Ω que são de alguma forma compatíveis com o conceito expresso por A , e o núcleo de A , denotado por $Nu(A)$, que agrupa elementos de A que são completamente compatíveis com o conceito expresso por A .

$$Su(A) = \{x \in \Omega / \mu_A(x) > 0\} = \lim_{\alpha \rightarrow 0} A_\alpha \quad (3.4)$$

$$Nu(A) = \{x \in \Omega / \mu_A(x) = 1\} = A_1 \quad (3.5)$$

A altura de A é expressa pelo maior grau de compatibilidade dos elementos de Ω em relação a A :

$$Al(A) = \sup_{x \in \Omega} \mu_A(x) \quad (3.6)$$

Um conjunto nebuloso A é dito normalizado se e somente se $Al(A) = 1$. A Figura 3.2 ilustra a cardinalidade, a altura, o suporte, o núcleo, e o corte de nível 0.5 de um conjunto nebuloso A [73] [74] [75].

Um conjunto nebuloso A é dito ser *convexo* em $\Omega \subseteq \mathfrak{R}$ se e somente se, seus cortes- α são convexas, isto é:

$$\forall x, y \in \Omega, \mu_A(\lambda x + (1 - \lambda)y) \geq \min(\mu_A(x), \mu_A(y)) \quad \text{para } \lambda \in [0, 1] \quad (3.7)$$

Seja A um conjunto nebuloso convexo, com $Su(A) = [s_{\inf}, s_{\sup}]$ e $Nu(A) = [n_{\inf}, n_{\sup}]$. Um intervalo nebuloso é um conjunto nebuloso normalizado e convexo em \mathfrak{R} tal que a função que descreve $\mu_A(x)$ entre s_{\inf} e n_{\inf} e aquela entre n_{\sup} e s_{\sup} são estritamente monotônicas. O conjunto nebuloso A da Figura 3.2 é um intervalo nebuloso. Um número nebuloso é um intervalo nebuloso unimodal.

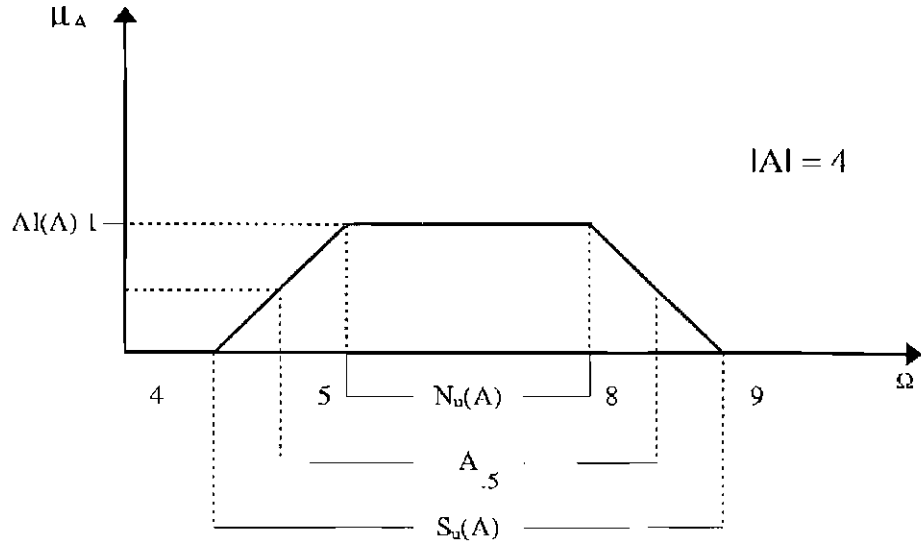


Fig. 3.2 - Cardinalidade, altura, suporte, núcleo e o corte de nível 0.5 do conjunto nebuloso A .

3.2 OPERAÇÕES EM CONJUNTOS NEBULOSOS

Similarmente às operações nos conjuntos “crisp”, existe a necessidade de proceder às operações de *intersecção*, *união* e *negação*, entre outras, nos conjuntos nebulosos.

3.2.1 Operadores de Intersecção, União e Complemento

Sejam A e B conjuntos nebulosos definidos em Ω . Pode-se expressar a intersecção (disjunção) destes conjuntos, como um outro conjunto $E = A \cap B$. Da mesma forma, pode-se expressar a união (conjunção) como um conjunto $F = A \cup B$.

Na teoria dos conjuntos nebulosos, a intersecção é implementada por uma família de operadores denominados de *t-normas*, e a união é implementada por uma família de operadores denominados de *t-conormas* ou *S-normas* [76]. Uma função $\nabla : [0, 1]^2 \rightarrow [0, 1]$, é dita ser comutativa, associativa e monotônica se ∇ satisfaz as seguintes propriedades, respectivamente:

$$i) \nabla(a, b) = \nabla(b, a) \quad \forall a, b \in [0, 1] \quad (\text{comutatividade});$$

$$ii) \nabla(a, \nabla(b, c)) = \nabla(\nabla(a, b), c) \quad \forall a, b, c \in [0, 1] \quad (\text{associatividade});$$

iii) $\nabla(a, b) \leq \nabla(c, d)$ se $a \leq c$ e $b \leq d \quad \forall a, b, c, d \in [0, 1]$ (*monotonicidade*);

Um operador $\top : [0, 1]^2 \rightarrow [0, 1]$ é denominado de *t-norma* se \top é comutativo, associativo e monotônico e verifica a seguinte propriedade:

iv) $\top(a, 1) = a$ (*elemento neutro 1*);

Da mesma maneira, uma *t-conorma* \perp é uma função $\perp : [0, 1]^2 \rightarrow [0, 1]$ que é comutativa, associativa e monotônica e verifica a propriedade:

v) $\perp(a, 0) = a$ (*elemento neutro 0*).

Uma *t-norma* \top e uma *t-conorma* \perp são duais em relação a uma operação de negação $\neg : [0, 1] \rightarrow [0, 1]$ se elas satisfazem as relações de De Morgan, isto é, se $\neg(\top(a, b)) = \perp(\neg a, \neg b)$ e $\neg(\perp(a, b)) = \top(\neg a, \neg b)$. O principal operador de negação é dado por $\neg a = 1 - a$, mas outros operadores podem ser encontrados na literatura [72] [77].

É importante notar que as *t-normas* e *t-conormas* se reduzem aos operadores clássicos de união e intersecção quando os conjuntos são conjuntos “crisp”.

A Tabela 3.1 indica as *t-normas* e *t-conormas* mais utilizadas e as Figuras 3.3 e 3.4 ilustram alguns destes operadores, em relação a dois conjuntos nebulosos A e B .

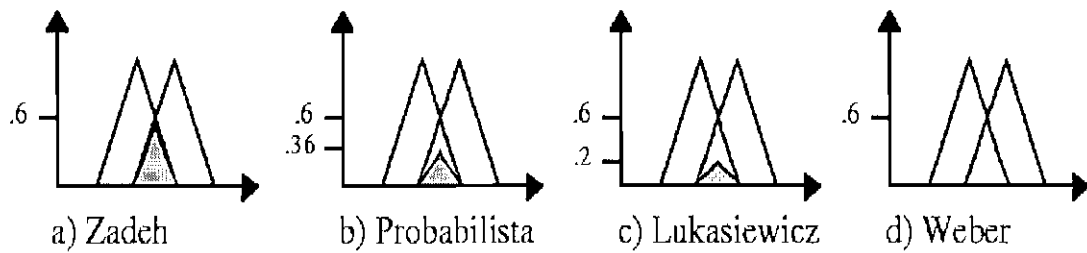


Fig. 3.3 - Ilustração das principais *t-normas*.

A maior *t-norma* é o *min* e a menor é a *t-norma* de Weber. Nos trabalhos iniciais de [72], são sugeridos o uso do operador de *mínimo* para a intersecção e do operador de *máximo* para a união de conjuntos nebulosos.

**TABELA 3.1 - PRINCIPAIS \top -NORMAS E
 \top -CONORMAS DUAIS**

$t - norma$	$t - conorma$	negação	nome
$\min(a, b)$	$\max(a, b)$	$1 - a$	Zadeh
$a \cdot b$	$a + b - ab$	$1 - a$	probabilista
$\max(a + b - 1, 0)$	$\min(a + b, 1)$	$1 - a$	Lukasiewicz
$\begin{cases} a, & \text{se } b = 1 \\ b, & \text{se } a = 1 \\ 0, & \text{senão} \end{cases}$	$\begin{cases} a, & \text{se } b = 0 \\ b, & \text{se } a = 0 \\ 1, & \text{senão} \end{cases}$	$1 - a$	Weber

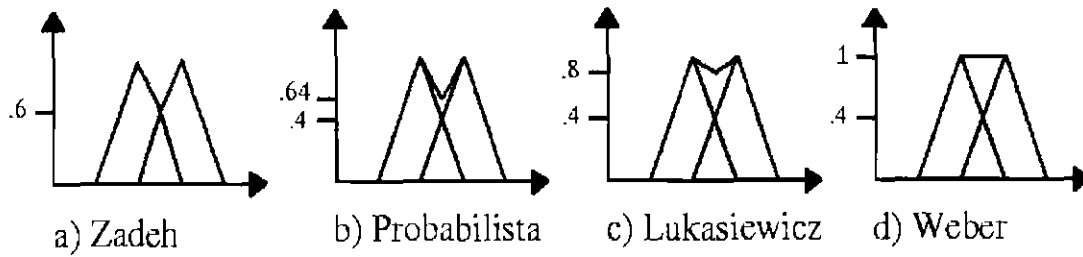


Fig. 3.4 - Ilustração das principais $t - conormas$.

3.2.2 Operadores de Implicação

Os operadores de implicação $I : [0, 1]^2 \rightarrow [0, 1]$ são usados para modelar regras de inferência do tipo **Se** $\langle premissa \rangle$ **Então** $\langle conclusão \rangle$. Considerando A e B dados por $\mu_A : X \rightarrow [0, 1]$, $\mu_B : Y \rightarrow [0, 1]$, a relação $A \rightarrow B$ é expressa como:

$$\mu_{A \rightarrow B}(x, y) = I(\mu_A(x), \mu_B(y)) \quad (3.8)$$

Existem três grandes classes de implicações propriamente ditas: As implicações S , que são da forma $I_S(a, b) = \perp(\neg a, b)$, as implicações R , que são tais que $I_R(a, b) = \sup \{c \in [0, 1] / \top(a, c) \leq b\}$, e as implicações QM , que são da forma $I_{QM}(a, b) = \perp(\neg a, \top(a, b))$, onde \top é uma $t - norma$, \perp é uma $t - conorma$ e \neg é uma negação.

As *t-normas* em si não são implicações propriamente ditas, mas são muito empregadas na prática como implicações, notadamente em aplicações de controle nebuloso. Um estudo mais detalhado pode ser encontrado em [73] [78].

A Tabela 3.2 mostra os principais operadores de implicação e a Figura 3.5 ilustra o usos de alguns destes operadores.

TABELA 3.2 - PRINCIPAIS OPERADORES DE IMPLICAÇÃO

<i>Implicação</i>	<i>nome</i>
$\max(1 - a, b)$	Kleene–Diemes
$\min(1 - a + b, 1)$	Lukasiewicz
$\begin{cases} 1, & \text{se } a \leq b \\ 0, & \text{senão} \end{cases}$	Rescher–Gaines “ <i>Sharp</i> ”
$\begin{cases} 1, & \text{se } a \leq b \\ b, & \text{senão} \end{cases}$	Brower–Gödel
$\begin{cases} \min(b/a), & \text{se } a \neq 0 \\ 1, & \text{senão} \end{cases}$	Goguen
$1 - a + ab$	Reichenbach “ <i>Estocástica</i> ”
$\max(1 - a, \min(a, b))$	Zadeh–Wilmott
$\min(a, b)$	Mamdani
$a \cdot b$	Larsen

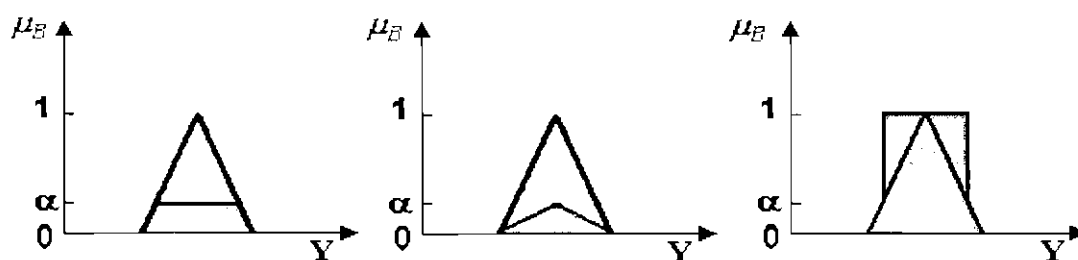


Fig. 3.5 - Ilustração das implicações $I(\alpha, \mu_B(y))$: i) Mamdani, ii) Larsen, iii) Gödel.

3.3 PROJETO DE CONTROLADORES NEBULOSOS

Ao contrário dos controladores convencionais em que o algoritmo de controle é descrito analiticamente por equações algébricas ou diferenciais, através de um modelo matemático, em controle nebuloso utilizam-se de regras lógicas no algoritmo de controle, com a intenção de descrever numa rotina a experiência humana, intuição e heurística para controlar um processo [72].

Uma *variável linguística* pode ser definida por uma quádrupla $(X, \Omega, T(X), M)$, onde X é o nome da variável, Ω é o universo de discurso de X , $T(X)$ é o conjunto de termos linguísticos, ou seja, um conjunto de nomes para valores de X , e M é um conjunto de regras semânticas para associar cada X com seu significado [74].

Para se atribuir um significado aos termos linguísticos, associa-se cada um deles a um conjunto nebuloso definido sobre Ω , o universo de discurso da variável linguística X .

Então, como é ilustrado na Figura 3.6, a variável linguística velocidade pode, por exemplo, assumir como valores um dos termos do conjunto:

$$\{\text{Negativa Alta}, \text{Negativa Baixa}, \text{Zero}, \text{Positiva Baixa}, \text{Positiva Alta}\}$$

O grau com que um valor x^* em Ω satisfaz o termo linguístico A é a pertinência de x^* em A , dada por $\mu_A(x^*)$.

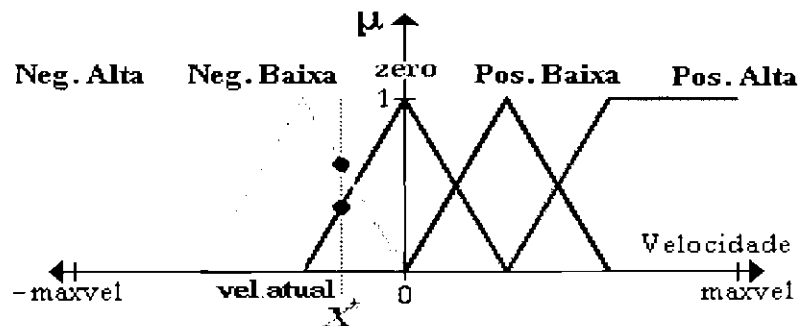


Fig. 3.6 - Termos linguísticos mapeiam a variável **Velocidade**.

FONTE: Adaptada de Bauer (1998).

Os controladores nebulosos são robustos e de grande adaptabilidade, incorporando conhecimento que outros sistemas não conseguem acomodar [75]. Também são versáteis, principalmente quando o modelo físico é complexo e de difícil representação matemática. No entanto, provar determinadas propriedades de robustez é uma tarefa difícil neste tipo de abordagem.

Em geral, os controladores nebulosos encontram maior utilidade em sistemas não-lineares, variantes no tempo, sendo capazes de superar perturbações e plantas com níveis de ruídos. Além disso, mesmo em sistemas onde a incerteza se faz presente de maneira intrínseca, agregam uma robustez característica.

A Figura 3.7 é a representação da estrutura básica de um controlador nebuloso como foi descrito em [15]. Muitas variações são propostas na literatura de acordo com o objetivo do projeto, mas esse é um modelo geral o suficiente para a identificação dos módulos que o compõem, fornecendo uma idéia do fluxo da informação.

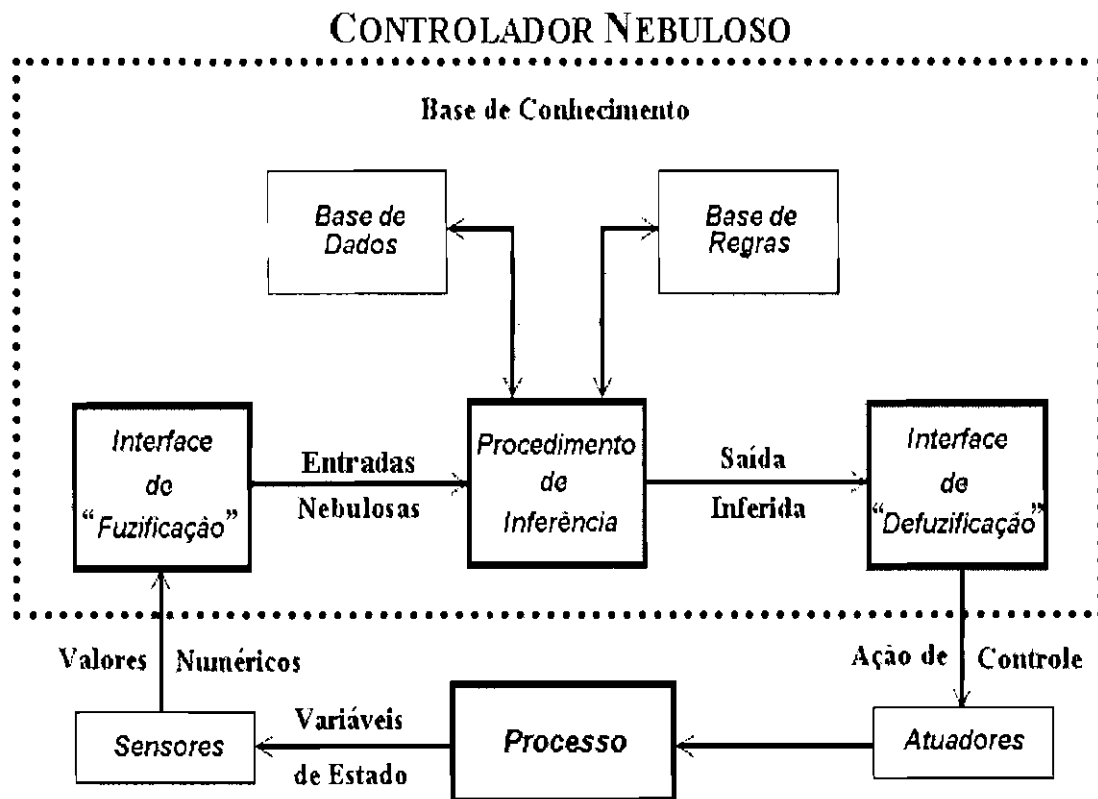


Fig. 3.7 - Estrutura de um Controlador Nebuloso.

3.3.1 Interface de “Fuzificação”

A *interface de “fuzificação”* faz a identificação dos valores das variáveis de entrada, as quais caracterizam o estado do sistema (variáveis de estado), e as normaliza em um universo de discurso padronizado. Estes valores são então “fuzificados”, com a transformação da entrada “crisp” em conjuntos nebulosos para que possam se tornar instâncias de variáveis linguísticas.

3.3.2 Base de Conhecimento

A *base de conhecimento* consiste de uma base de dados e uma base de regras, de maneira a caracterizar a estratégia de controle e as suas metas.

Na *base de dados* ficam armazenadas as definições sobre discretização e normalização dos universos de discurso, e as definições das funções de pertinência dos termos nebulosos.

A *base de regras* é formada por estruturas do tipo **Se** <premissas> **Então** <conclusões>, como por exemplo:

Se *Erro é Grande e Δ Erro é Positivo* **Então** *Velocidade é Positiva Pequena*

Estas regras, juntamente com os dados de entrada, são processados pelo *procedimento de inferência*, o qual infere as ações de controle de acordo com o estado do sistema, aplicando o operador de implicação, conforme o procedimento de inferência que será descrito na Seção 3.3.3.

Segundo [15], as regras de controle podem ser de avaliação do estado e avaliação do objetivo. A primeira infere uma ação de controle como função do estado do processo num certo instante e a outra, conhecida como controle preditivo [79] [80] [81], infere ações de controle presentes e futuras considerando os objetivos [15] [82].

Em um dado controlador nebuloso, é importante que existam tantas regras quantas forem necessárias para mapear totalmente as combinações dos termos das variáveis,

isto é, que a base seja completa, garantindo que exista sempre ao menos uma regra a ser disparada para qualquer entrada. Também são essenciais a consistência, onde procura-se evitar a possibilidade de contradições e a interação entre as regras, gerenciada pela função de implicação de modo a contornar as situações de ciclo.

As premissas são relacionadas pelos conectivos lógicos, dados pelo *operador de conjunção* (**e**) e o *operador de disjunção* (**ou**). Em geral as regras tem a forma de um sistema de múltiplas entradas e múltiplas saídas (MIMO), mas que pode ser transformado em vários sistemas com múltiplas entradas e uma saída (MISO). Por exemplo, a regra MIMO

$$R : \text{Se } x_1 \text{ é } A_1 \text{ e } \dots \text{e } x_n \text{ é } A_n \text{ Então } y_1 \text{ é } C_1 \text{ e } \dots \text{e } y_m \text{ é } C_m$$

é em m regras MISO:

$$R_j : \text{Se } x_1 \text{ é } A_1 \text{ e } \dots \text{e } x_n \text{ é } A_n \text{ Então } y_j \text{ é } C_j$$

Em geral não se aceitam conectivos “ou” na conclusão [73].

3.3.3 Procedimento de Inferência

Um controlador nebuloso é um sistema especialista simplificado onde a consequência de uma regra não é aplicada como antecedente de outra [14]. Assim, o processo de inferência consiste em:

- 1) Verificação do grau de compatibilidade entre os fatos e as cláusulas nas premissas das regras;
- 2) Determinação do grau de compatibilidade global das premissas de cada regra;
- 3) Determinação do valor da conclusão e/ou sua crença em cada regra, em função do grau de compatibilidade da regra com os dados e a ação de controle constante na conclusão (precisa ou não);

- 4) Agregação dos valores obtidos como conclusão nas várias regras, obtendo-se uma ação de controle global.

Os tipos básicos de controladores nebulosos são os modelos clássicos, compreendendo o modelo de Mamdani e o de Larsen, e os modelos de interpolação, compreendendo o modelo de Takagi-Sugeno e o de Tsukamoto (vide [14] [15] [82]).

Os modelos diferem quanto à forma de representação dos termos na premissa, quanto à representação das ações de controle e quanto aos operadores utilizados para implementação do controlador.

3.3.3.1 Controle Nebuloso Clássico

Nos modelos clássicos, a conclusão de cada regra especifica um termo nebuloso dentre um conjunto fixo de termos (geralmente em número menor que o número de regras). Estes termos são usualmente conjuntos nebulosos convexos como triângulos, funções em forma de sino (“bell-shaped”) e trapézios.

Dado um conjunto de valores para as variáveis de estado, o sistema obtém um conjunto nebuloso (muitas vezes sub-normalizado), como o valor da variável de controle. Este conjunto nebuloso representa uma ordenação no conjunto de ações de controle aceitáveis naquele momento. Finalmente, uma ação de controle global é selecionada dentre aquelas aceitáveis em um processo conhecido como defuzificação.

Sejam as regras R_j codificadas como: $R_j : \text{Se } x_1 \text{ é } A_{1,j} \text{ e } \dots \text{ e } x_n \text{ é } A_{n,j} \text{ Então } y_j \text{ é } C_j$

No modelo clássico, o processamento de inferência é feito da seguinte maneira:

- ▷ **Passo 1:** Seja x_i uma variável de estado, definida no universo X_i , a realização de x_i é definida como o valor $x_i^* \in X_i$ que esta assume em X_i em um dado momento;
- ▷ **Passo 2:** A *compatibilidade* da i – ésima premissa da j – ésima regra com x_i^* , ou seja, a compatibilidade de x_i^* , $1 \leq i \leq n$, com $A_{i,j}$ da regra R_j , $1 \leq j \leq m$, é definida por:

$$\alpha_{i,j} = \mu_{A_{i,j}}(x_i^*) \quad 1 \leq i \leq n \quad 1 \leq j \leq m \quad (3.9)$$

- ▷ **Passo 3:** Com as premissas de uma dada regra avaliadas, a *compatibilidade global* α_j da regra R_j , $1 \leq j \leq m$, com os x_i^* é determinada com uma *t-norma* \top (operador de conjunção \top):

$$\alpha_j = \top(\alpha_{1,j}, \dots, \alpha_{n,j}) \quad 1 \leq j \leq m \quad (3.10)$$

- ▷ **Passo 4:** O α_j assim obtido é relacionado com o respectivo conjunto nebuloso C_j do conseqüente da regra R_j , dando origem a um conjunto C'_j , $1 \leq j \leq m$, através de um *operador de implicação* I :

$$\mu_{C'_j}(y) = I(\alpha_j, \mu_{C_j}(y)) \quad \forall y \in Y \quad (3.11)$$

- ▷ **Passo 5:** Um operador ∇ faz a *agregação* das contribuições das várias regras acionadas C'_j num único conjunto nebuloso C' :

$$\mu_{C'}(y) = \nabla(\mu_{C'_1}(y), \dots, \mu_{C'_m}(y)) \quad \forall y \in Y \quad (3.12)$$

O operador ∇ é usualmente uma *t-conorma*, quando o operador de implicação I é uma *t-norma*, e uma *t-norma* em caso contrário.

3.3.3.1.1 Controladores Nebulosos Clássicos de Mamdani e de Larsen

Os modelos clássicos seguem estritamente os passos mostrados na Seção 3.3.3.1 anteriormente, sendo que no modelo de Mamdani temos $\top(a, b) = \min(a, b)$, $I = \min(a, b)$ e $\nabla(a, b) = \max(a, b)$ e no modelo de Larsen temos $\top(a, b) = a * b$, $I = a * b$ e $\nabla(a, b) = \max(a, b)$.

A Figura 3.8 e a Figura 3.9 ilustram o processo de raciocínio do modelo de Mamdani e do modelo de Larsen, respectivamente.

Os controladores de Mamdani e Larsen necessitam da utilização de uma interface de defuzificação para gerar a ação de controle, isto é, escolher um único valor no suporte de C' , o que será visto na Seção 3.3.4.

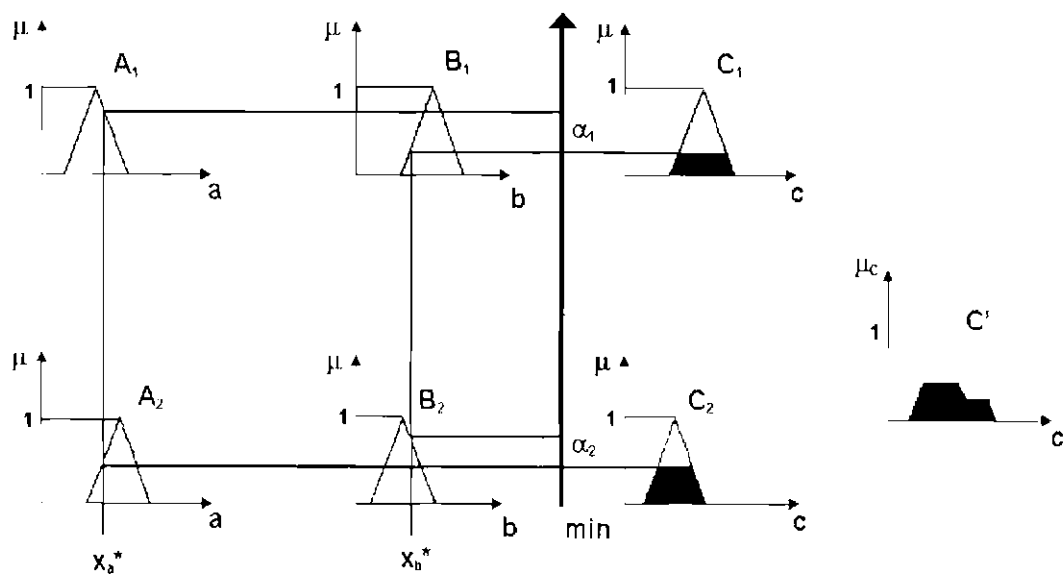


Fig. 3.8 - Modelo clássico de Mamdani.

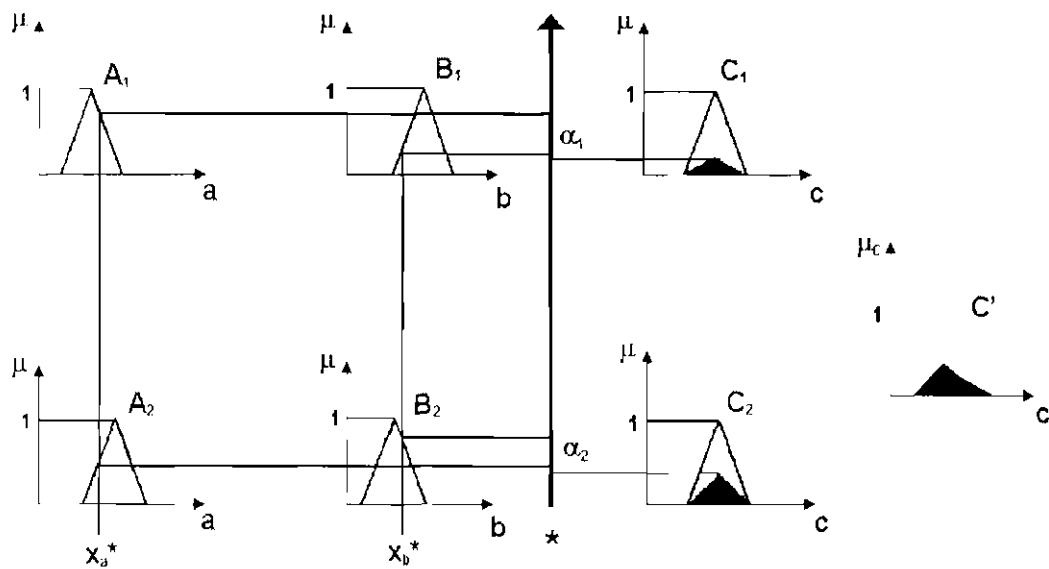


Fig. 3.9 - Modelo clássico de Larsen.

3.3.3.2 Controle Nebuloso por Interpolação

Nos modelos de interpolação, cada conclusão é dada através de uma função estritamente monotônica, usualmente diferente para cada regra. No modelo de Takagi-Sugeno, a função é uma combinação linear das entradas, tendo como parâmetros um

conjunto de constantes. No esquema de Tsukamoto, a função é geralmente não linear, tendo como domínio os possíveis graus de compatibilidade entre cada premissa e as entradas.

Em ambos os esquemas, obtém-se, para cada regra, um único valor para a variável de controle. Finalmente, uma ação de controle global é obtida fazendo-se uma média ponderada dos valores individuais obtidos (os pesos são dados pelo próprio grau de compatibilidade entre a premissa da regra e as entradas, normalizadas).

Para os modelos de interpolação, também são válidos os passos 1, 2 e 3 descritos nos modelos clássicos. No entanto, a operação de implicação do passo 4 determina uma ação de controle precisa para cada regra. Essas ações individuais são interpoladas no passo 5, gerando uma ação de controle única e precisa.

3.3.3.2.1 Modelo de Tsukamoto

O modelo de Tsukamoto [14] [73] exige que pelo menos os conjuntos nebulosos C_j , que estão associados com os consequentes das regras, sejam funções monotônicas. No passo 4, o método de interpolação obtém um valor preciso y'_j relativo à ação de controle da regra R_j , que é dado por:

$$y'_j = \mu_{C_j}(\alpha_j) \quad (3.13)$$

Por sua vez, no passo 5, os valores obtidos como conclusão nas várias regras são agregados em uma única ação de controle precisa y' , através de uma média ponderada, como:

$$y' = \frac{\sum_{j=1}^n (\alpha_j \cdot y'_j)}{\sum_{j=1}^n \alpha_j} \quad (3.14)$$

Neste caso, a interface de “defuzificação” não é utilizada. A Figura 3.10 representa uma interpretação gráfica do modelo de interpolação de Tsukamoto.

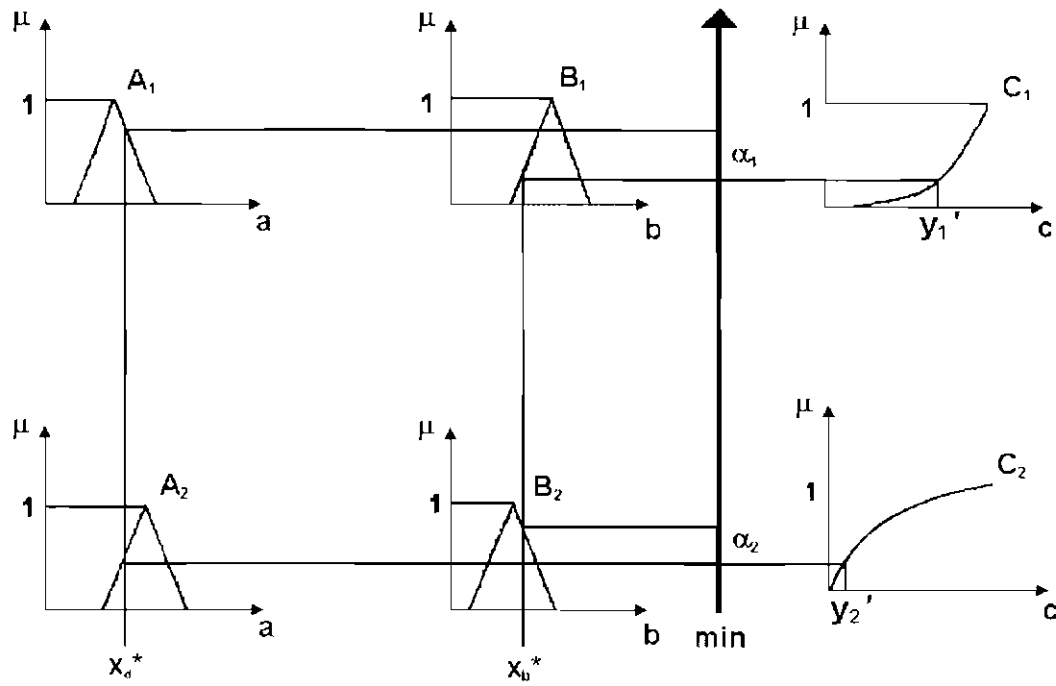


Fig. 3.10 - Modelo de interpolação de Tsukamoto.

3.3.3.2.2 Modelo de Takagi-Sugeno

O modelo de Takagi e Sugeno [14] [73] exige que todos os termos nebulosos $A_{i,j}$ na premissa sejam funções monotônicas e que as conclusões das regras sejam dadas por funções:

$$f_j(x_1^*, x_2^*, \dots, x_m^*) = d_0 + d_1 \cdot x_1^* + \dots + d_m \cdot x_m^* \quad (3.15)$$

onde cada d_k é uma constante.

A ação de controle obtida por cada regra R_j é dada por:

$$y_j' = f_j(x_1^*, x_2^*, \dots, x_m^*) \quad (3.16)$$

A ação de controle y' é então obtida pela Equação 3.14, como no modelo de Tsukamoto. Na Figura 3.11, ilustra-se a inferência através do método de Takagi e Sugeno de duas regras MISO.

C' obtido no passo 4.

O procedimento compreende a identificação do domínio das variáveis de saída num correspondente universo de discurso e com a ação de controle nebulosa inferida evolui-se uma ação de controle não nebulosa.

Os métodos de “defuzificação” mais utilizados são:

- 1) Primeiro Máximo (SOM): Encontra o valor de saída através do ponto em que o grau de pertinência da distribuição da ação de controle atinge o primeiro valor máximo;
- 2) Método da Média dos Máximos (MOM): Encontra o ponto médio entre os valores que têm o maior grau de pertinência inferido pelas regras;
- 3) Método do Centro da Área (COA): O valor de saída é o centro de gravidade da função de distribuição de possibilidade da ação de controle.

A Figura 3.12 é um exemplo para o método do centro da área.

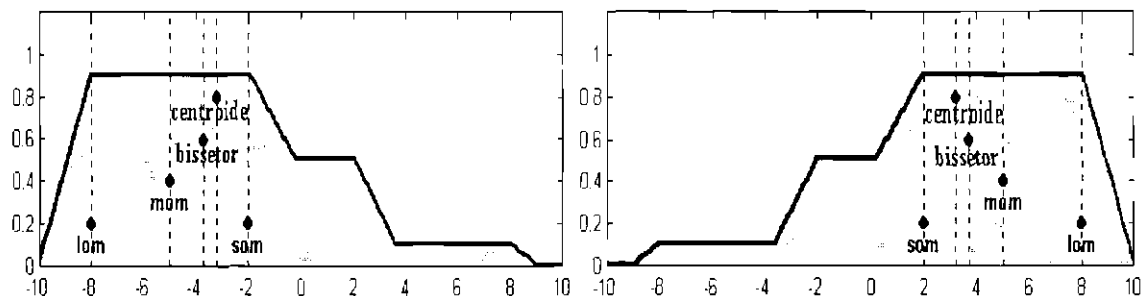


Fig. 3.12 - Defuzificação pelo Método do Centro de Gravidade.

FONTE: Adaptada de Bauer (1998).

A seleção do método está relacionada diretamente com as características do processo controlado e o comportamento de controle necessário. O método do último máximo ou a média dos máximos, por exemplo, que assemelham o efeito de um controlador “bang-bang”, podem conduzir à ações de controle inadequadas (solavancos) ao modo

de operação e assim, causar danos de ordem prática em equipamentos como os atuadores.

Existem ainda outros métodos, mas observe-se que as diferenças são dadas em termos de velocidade e eficiência, características que devem ser analisadas em conjunto com os requisitos do projeto [73] [83] [75] [14].

É importante lembrar que os controladores nebulosos do tipo interpolação não necessitam de uma interface de “defuzificação”, pois já obtém-se diretamente os valores precisos para as entradas do processo controlado [73] [83] [14].

CAPÍTULO 4

MODELO DO SATÉLITE

Este trabalho utiliza um simulador desenvolvido em [9], para um modelo semelhante ao satélite Franco–Brasileiro. As características deste modelo de satélite são apresentadas, bem como um controlador PD projetado para a fase de apontamento do satélite, também desenvolvido em [9].

4.1 CARACTERÍSTICAS DO SATÉLITE

Considera-se o satélite como um corpo rígido com produtos de inércia nulos. O sistema de controle utiliza-se de três giros, um sensor solar e um sensor de estrelas. A dinâmica rotacional do satélite pode ser representada pelo conjunto de equações diferenciais como segue [9]:

$$\begin{aligned}\dot{\hat{\xi}} &= \varpi \\ \dot{\hat{\varpi}} &= -K_P * \hat{\xi} - K_D * \hat{\varpi} + T_P(t) \\ \dot{\hat{\xi}} &= \hat{\varpi}\end{aligned}\tag{4.1}$$

onde $\hat{\xi}$, $\hat{\varpi}$ e $T_P(t)$ representam as estimativas de rotação (ξ), velocidade angular (ϖ) e os torques perturbadores, respectivamente. A estimativa da velocidade angular ($\hat{\varpi}$) é determinada com:

$$\hat{\varpi} = o - \hat{b}_k\tag{4.2}$$

onde (o) é a saída do giro dada por:

$$o(t) = \varpi + b_k + n_k, \quad t \in [t_k, t_{k+1})\tag{4.3}$$

na qual b_k representa o “bias” de longo (deriva) período e n_k “bias” de curto (ruído) período.

Das duas equações anteriores (Equação 4.2 e Equação 4.3), resulta:

$$\hat{\varpi} = \varpi + b_k - \hat{b}_k + n(\sigma_n^2) \quad (4.4)$$

onde σ é a variância do ruído n .

Pode-se propagar no tempo o “bias” de longo período e sua estimativa com:

$$b_{k+1} = e^{-\lambda \Delta t} * b_k + n(2 * \lambda * \Delta t * \sigma_d^2) \quad (4.5)$$

$$\hat{b}_{k+1} = e^{-\lambda \Delta t} * \hat{b}_k$$

onde λ representa a correlação e σ_d a variância de longo período.

Fazendo $D_k = b_k - \hat{b}_k + n(\sigma_n^2)$, o conjunto das três primeiras equações (Equações 4.1) resulta em:

$$\dot{\xi} = \varpi$$

$$\dot{\varpi} = -K_P * \hat{\xi} - K_D * (\varpi + D_k) + T_P(t) \quad (4.6)$$

$$\dot{\hat{\xi}} = (\varpi + D_k)$$

Os dados orbitais de projeto utilizados são:

- Altitude do apogeu: 1500 km;
- Altitude do perigeu: 400 km;
- Inclinação: 7°;
- Argumento do perigeu: 90°;
- Longitude do nodo ascendente: -90° e
- Anomalia média: -90°.

O sistema de coordenadas para o controle de atitude do satélite adota um *referencial externo*, definido como:

- Z_E orientado para o polo Norte da Eclíptica;
- X_E apontando para a posição média do Sol e
- Y_E orientado de maneira a formar um sistema dextrógiro.

Os momentos de inércia do satélite para os três eixos são:

- $I_X = 7 \text{ kg.m}^2$;
- $I_Y = I_Z = 10 \text{ kg.m}^2$.

Para representar os três maiores elementos envolvidos (satélite, Terra e Sol) apresenta-se uma imagem Figura 4.1 com o fim ilustrativo, a qual não tem relação com o projeto proposto.

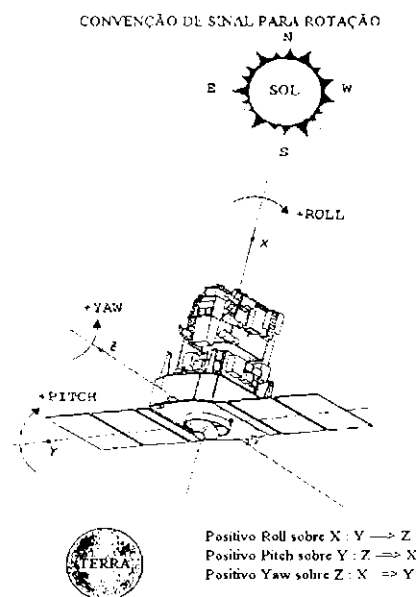


Fig. 4.1 - Ilustração do sistema de coordenadas de um satélite.

FONTE: Adaptado de Duarte (1998).

As perturbações devido ao arrasto atmosférico, pressão de radiação solar e campo magnético mostradas na Figura 4.2, são consideradas nas equações da dinâmica do satélite.

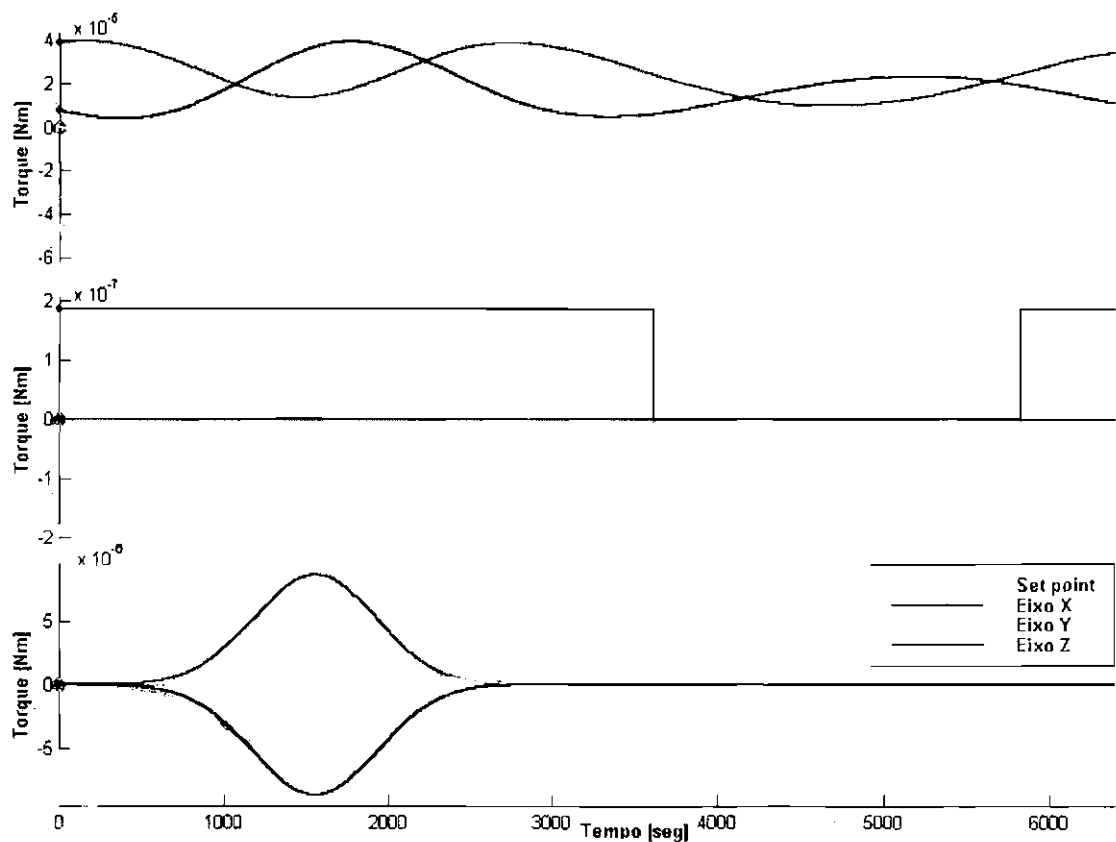


Fig. 4.2 - Torques perturbadores agindo sobre o modelo do satélite, simulados para uma órbita: a) Torque Magnético b) Pressão de Radiação Solar c) Arrasto Atmosférico.

4.2 CONTROLADOR PD ORIGINAL

Um controlador PD para o controle de atitude do satélite na fase de apontamento para o Sol foi projetado através do método de alocação de pólos e um filtro de Kalman, utilizando as Equações 4.6 [9]. No presente trabalho o filtro de Kalman não é considerado nos experimentos.

Para o desenvolvimento do controle de atitude durante a fase de apontamento, em [9] optou-se pelo uso de um controlador do tipo PD para cada eixo, conforme a equação que segue:

$$f(t) = K_P * e(t) + K_D * \Delta e(t) \quad (4.7)$$

onde $e(t)$ é o sinal erro, $\Delta e(t) = \frac{d}{dt}(e(t))$, e K_P e K_D são os ganhos proporcional e derivativo, respectivamente.

Na determinação dos ganhos desse controlador foi utilizado o método da alocação de pólos, visando atender requisitos de projeto tais como o tempo de pico (T_p) e o tempo de acomodação (T_s).

Para projetar o controlador PD descrito em [9] foram fixados ganhos proporcionais iguais para os eixos X e Y , como hipótese de simplificação, o que não implica em prejuízo algum, já que os efeitos das perturbações são atenuados através do controle do eixo Z . As simulações procedidas por [9] seguiram os requisitos:

1) Precisão de apontamento:

- 0.15° para os eixos y e z ;
- 0.5° para o eixo x .

2) Estabilidade:

- $0.05^\circ/s$ para os três eixos.

Os ganhos encontrados no modelo original [9] foram:

- Ganho proporcional para os eixos X e Y : $K_{P_X} = K_{P_Y} = 0.0263$;
- Ganho derivativo para os eixos X e Y : $K_{D_X} = K_{D_Y} = 0.08$;
- Ganho proporcional para o eixo Z : $K_{P_Z} = 0.0272$;
- Ganho derivativo para o eixo Z : $K_{D_Z} = 0.1$.

As Figuras 4.3 e 4.4 apresentam os resultados do erro de posição angular (ζ) e erros de velocidade angular (ω), obtidos na simulação de uma órbita (6400 seg.) utilizando este controlador PD.

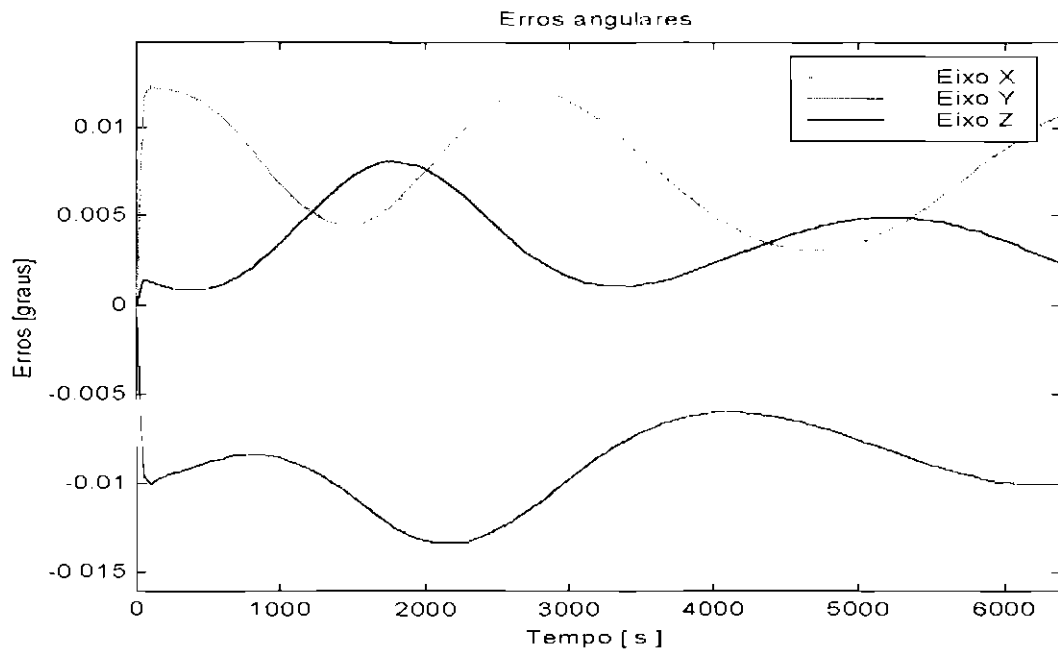


Fig. 4.3 - Erros de posição angular no apontamento do satélite com o controlador PD original, simulado para uma órbita.

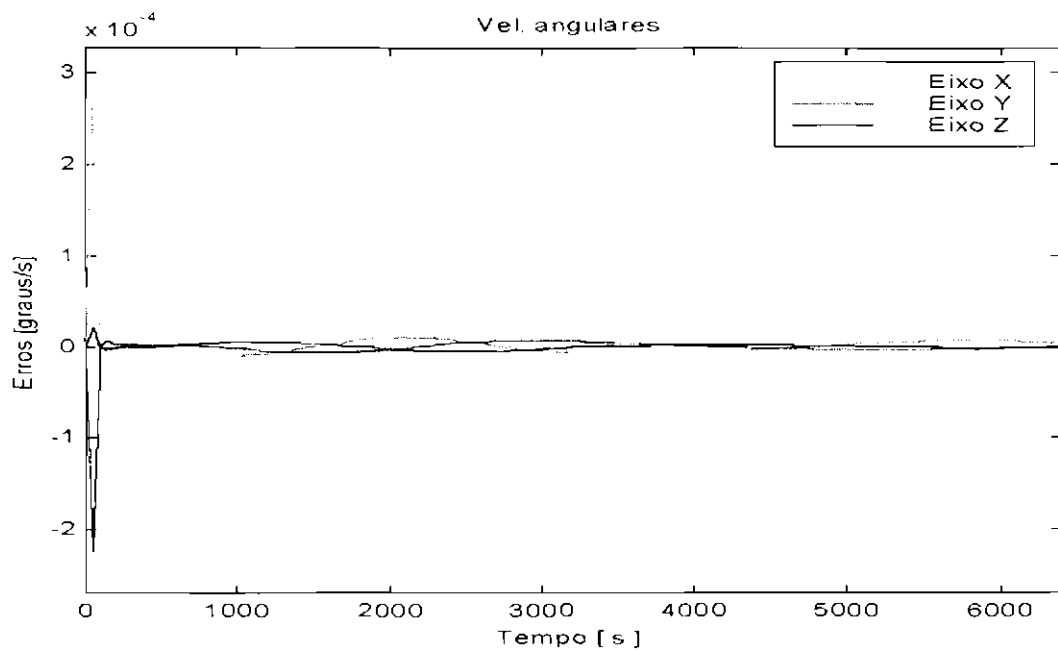


Fig. 4.4 - Erros de velocidade angular desenvolvidos nos eixos X, Y e Z utilizando controlador PD original.

CAPÍTULO 5

PLATAFORMA COMPUTACIONAL

O modelo inicial de controle e simulação, tomado como base no desenvolvimento deste trabalho está descrito em [9]. O simulador utiliza as ferramentas disponíveis no pacote integrado MATLAB, e foi desenvolvido com o auxílio dos Toolbox Control Systems [57] [58] [60] [61] [62] [59] [63].

Este software é um interpretador, e portanto, as rotinas geradas podem ser utilizadas em plataformas distintas sem exigir alterações no código fonte. O ambiente computacional é amigável e versátil, possuindo uma interface poderosa que permite uma rápida inferência e conta-se ainda com a disponibilização de saídas gráficas que é outra característica conveniente às tarefas científicas.

Essas mesmas ferramentas foram utilizadas neste trabalho, não sendo necessárias modificações nos cálculos efetuados pelo código original do simulador. Conjuntamente ao simulador, foram utilizados o Toolbox de Fuzzy, também projetado pela MathWorks e um algoritmo genético desenvolvido por [64], utilizando como referência o modelo e a estrutura descrita em [69].

Existem outras diversas ferramentas computacionais disponíveis comercialmente e muitas disponibilizadas livremente por instituições educacionais. Alguns exemplos de sistemas nebulosos são o “Fuzzy Logic Development System by Inform” (Fuzzy-TECH), “Fuzzy Logic Toolkit by Logic Programming Associates” (FLINT), “Original C code by Roger Jang” (ANFIS) e “A Design Environment for Fuzzy Logic Control Systems” (Xfuzzy). Para simulações de órbita pode-se citar o “Satellite Tool Kit” (STK) e referências para AGs são o “Genetic Algorithm C++ class library” (GAGS), “Illinois Genetic Algorithms Laboratory” (IlliGAL), General Architecture for Genetic Algorithms (GAGA), “GA Optimization Toolbox for Matlab” (GAOT) e o “Genetic Algorithm Optimized for Portability and Parallelism System by Michigan State University” (GALOPPS), entre muitos outros. Estas ferramentas podem

ser em alguns casos utilizadas em conjunto e constituem-se em alternativas para o desenvolvimento de projetos e pesquisas.

Como o código fonte produzido em [9] pode ser manipulado facilmente, as modificações necessárias para este trabalho foram efetuadas progressivamente para possibilitar o acoplamento das diversas rotinas, procedendo paralelamente sua validação e subseqüentes comparações de resultados.

Um fato que merece atenção refere-se a necessidade de flexibilidade do algoritmo genético, já que o mesmo é responsável pela manipulação de mais de uma variável, devendo codificá-las em um mesmo cromossoma e ainda assim, manter a individualidade de cada variável, com tamanho suficiente para conseguir boa representatividade.

Para possibilitar a criação dos controladores através de AG, foi necessário primeiramente proceder à integração entre as rotinas do pacote de simulação da dinâmica do satélite e o algoritmo genético, onde as soluções procuradas são tratadas como parâmetros codificados em cromossomas. Esse processo deve fazer o intercâmbio dos dados entre as diferentes rotinas, sem alterar os cálculos envolvidos, para garantir a integridade das simulações.

Num primeiro momento os algoritmos de simulação foram adaptados para responder aos ganhos encontrados pelo algoritmo genético e não mais somente aos ganhos obtidos com o método de alocação dos pólos, como no modelo original.

Várias tarefas de grande relevância foram realizadas nessa etapa do processo, envolvendo a integração do algoritmo genético aos algoritmos da simulação com o controle proporcional/derivativo. Esse conjunto teve que ser ajustado para trabalhar simultânea e sincronizadamente. Além disso, muitos dos parâmetros envolvidos em um algoritmo genético são empíricos e tiveram que ser testados, como por exemplo o tamanho dos cromossomas ou o número de indivíduos em cada população.

Para conseguir eficiência no processo de busca, uma atenção especial foi direcionada

ao desenvolvimento da função responsável pela avaliação e validação dos resultados obtidos com o controle, através da evolução da simulação.

Mesmo com os sistemas trabalhando em conjunto, a função de avaliação permaneceu como um objeto de estudo contínuo durante toda a construção da plataforma e execução dos experimentos. Isto porque o aperfeiçoamento das gerações depende desta função que, por sua vez, tende conduzir a resultados satisfatórios quando o algoritmo genético converge para um ponto satisfatório no espaço de busca.

Completado o processo de acoplamento entre as rotinas do pacote de simulação e o algoritmo genético, proporcionou-se o intercâmbio das informações produzidas pelos diferentes algoritmos. Alguns testes para verificação da execução das tarefas foram então executados e a estrutura inicial mostrou-se suficientemente funcional.

Na fase inicial de desenvolvimento deste trabalho, uma interface em linha de “prompt” foi utilizada para fornecer as entradas de dados. Apesar de ser consideravelmente eficiente, introduzir informações neste modo é significativamente desconfortável, tendo em vista o número de repetições dos testes.

Uma interface gráfica foi então desenvolvida com a finalidade de atenuar a dificuldade de interação entre o usuário e o modelo de otimização, compondo desta forma uma ferramenta de uso mais simples e visual.

As próximas seções detalham a interface gráfica e os testes iniciais desenvolvidos para verificar a funcionalidade da plataforma.

5.1 INTERFACE GRÁFICA

A interface gráfica foi concebida de forma modular para propiciar maior versatilidade na alteração das informações necessárias às rotinas de otimização do modelo de controle. Além disso, como o próprio modelo do satélite pode sofrer pequenas alterações, esta flexibilidade garante facilidade de reutilização.

A Figura 5.1 retrata uma chamada à função **control** (*control.m*), em ambiente Ma-

tlab, que compõe o módulo principal da interface gráfica. Nesta interface observam-se diferentes botões que acionam outras interfaces para os módulos secundários. Também existem duas entradas de dados na janela principal, a primeira correspondente ao número de órbitas a simular e a outra ao intervalo de tempo para verificação da posição orbital.

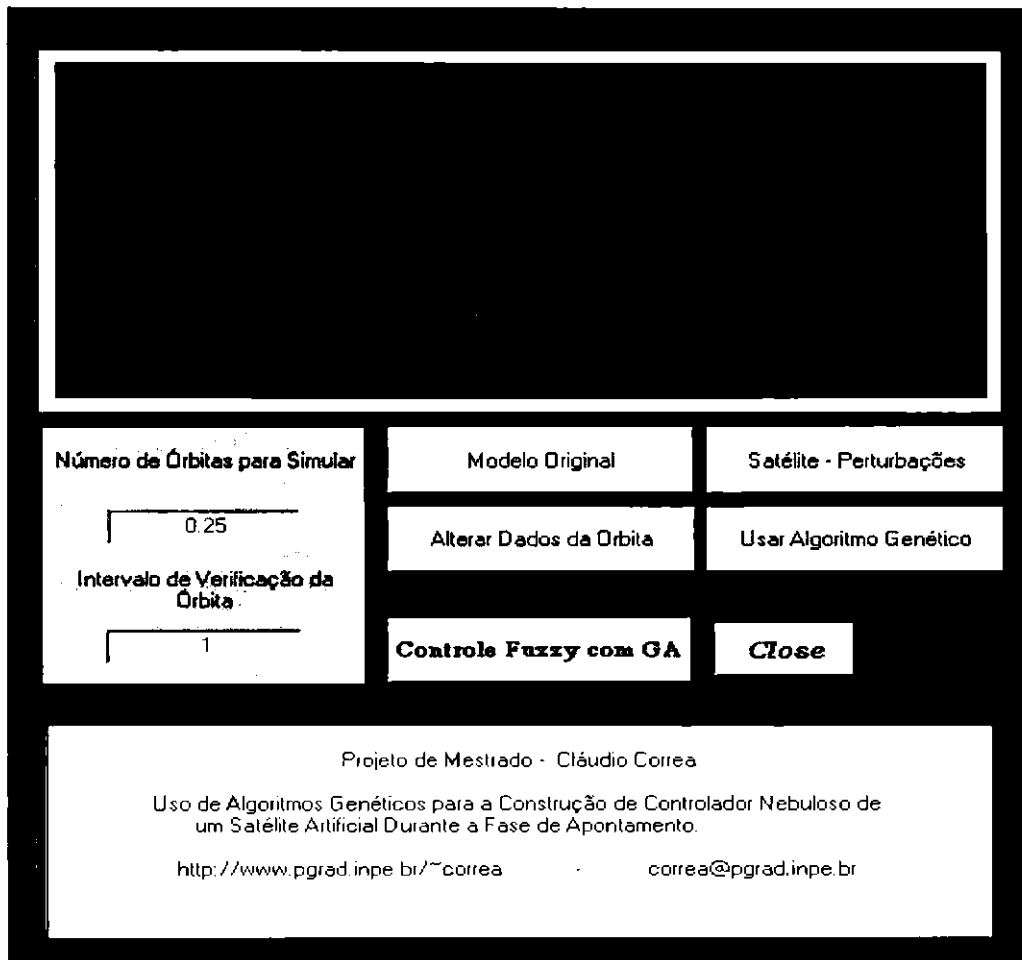


Fig. 5.1 - Janela principal da interface gráfica.

5.1.1 Módulos da Interface

Os módulos intermediários da interface permitem a seleção de itens ou atualização de valores referentes ao satélite, perturbações, órbita, algoritmo genético e sistema de controle, conforme as Figuras 5.2a, 5.2b, 5.4a e 5.3:



Fig. 5.2 - a) Interface para alteração de informações do satélite e perturbações. b) Seleção e atualização dos dados da órbita, do lançamento e cálculos dos ganhos através do método de Alocação dos Pólos.

Esta ferramenta foi construída não só com o propósito de conferir maior convivialidade ao manuseio das entradas, mas principalmente para disponibilizar recursos de trabalho para outras tarefas e desenvolvimentos.

O código fonte produzido em formato texto possui comentários relevantes ao entendimento das equações e modelos implementados. Todo o conjunto de rotinas pode ser facilmente alterado devido às características da programação adotada. Existem apenas funções, que são chamadas e executadas através do botão **Run**, visível na janela principal da interface após a seleção de uma das opções.

Os modelos matemáticos empregados nas rotinas podem, sem maior esforço, receber complementação ou aperfeiçoamento. Necessita-se para isto, fazer somente interferências locais. Estas mesmas funções estão disponíveis para uso independente

Número de Indivíduos (/população)	20	<div>Mutação (%)</div> <div>3.3</div>
Número de Gerações	25	
Limite Inferior do Universo de Discurso []	[0 0 0 0 0]	
Limite Superior do Universo de Discurso []	[2 2 2 2 2]	
Tamanho do Cromossomo []	[20 20 20 20 20]	
		<div>Crossover (%)</div> <div>90</div>
		<div>Tolerância de Término</div> <div>1e-005</div>
Close		
<p>The algorithm implemented here is taken from Genetic Algorithms in Search, Optimization, and Machine Learning, David E. Goldberg, Addison-Wiley Publishing Company, Inc., 1989.</p>		

Fig. 5.3 - Interface de configuração dos parâmetros genéticos e limites de busca.

ou em outros algoritmos. É possível ainda, acrescentar novos cálculos, como por exemplo, de outras perturbações ou incluir ruído, entre outros.

Todas as funções são atreladas através da interface gráfica, que executa as opções selecionadas e ao final da otimização disponibiliza os resultados através da saída gráfica (Figura 5.4b) para uma rápida verificação visual. Estes mesmos resultados são armazenados em arquivo, permanecendo disponíveis para outras eventuais análises.

A utilização da ferramenta exige pouco conhecimento do usuário. Existem entradas de dados simples, onde é necessário fornecer um único valor, tomando o cuidado de observar a unidade de medida adotada. Alguns campos de entrada solicitam a

Fig. 5.4 - a) Seleção do controlador nebuloso e do número de termos nebulosos. b) Saída gráfica dos valores de erro e perturbações.

informação de um vetor, nestes, basta seguir o modelo do valor “standard”. Os casos de erro são, em sua maioria, explicitados na linha de “prompt”. Uma ressalva importante é a necessidade de que cada valor fornecido seja seguido por um ENTER, para que ocorra o armazenamento e posterior validação da informação fornecida.

A simplicidade da ferramenta não implica em simplicidade de modelo. Para proteger constantes e outros detalhes importantes, não existe a disponibilidade dessas informações na interface gráfica, porém, com o auxílio de um editor de texto é possível editar o código fonte onde essas informações são declaradas.

Os dois módulos secundários que estão relacionados com a seleção do tipo de controlador nebuloso e ao algoritmo genético, são relativamente mais rígidos, no sentido de que proporcionam um número menor de escolhas e alterações possíveis. Inicialmente, para familiarizar e facilitar sua utilização e de outro modo, para não induzir uma excessiva variação de parâmetros. Por sua vez, isto implica na necessidade de

maior conhecimento sobre as técnicas de controle e otimização.

5.1.2 Restrições da Interface Gráfica

Uma característica desejada é a portabilidade do código fonte. Trabalhando em ambiente interpretado, geralmente isso não representa um problema, porém, com a criação da interface gráfica pode ocorrer o comprometimento deste atributo. Esta dificuldade refere-se à diferentes plataformas (hardware), já que não há disponibilidade do ambiente Matlab em equipamentos variados nesta instituição. Em PC, plataforma “Complex Instructions Set Computing” (CISC), várias verificações foram efetuadas, sem maiores complicações.

O principal problema encontrado neste primeiro momento está relacionado com a estrutura vinculada ao modelo de otimização. Os valores necessários para as avaliações efetuadas na função de avaliação, são obtidos por meio da simulação das ações de controle desenvolvidas durante a projeção da órbita. Neste caso, são aplicadas equações diferenciais para a propagação da órbita e esta é uma tarefa que demanda um tempo de processamento relativamente elevado. Por sua vez, a referida função de avaliação é executada diversas vezes pelo AG.

Esse tempo não pode ser determinado com precisão porque está atrelado fundamentalmente com as configurações de hardware, mas também depende em grande parte do desempenho do AG e o número de gerações necessárias à sua convergência. De forma bastante grosseira, em máquinas simples (Pentium 133Mhz, Win95), pode ser solicitado tempo de processamento superior a uma semana. Mas em máquinas de maior poder de processamento (Sparc 200 Mhz Ultra 2, Pentium NT 200 Mhz), esse tempo é consideravelmente reduzido para horas.

Tal característica não deve ser interpretada como um motivo para o desentusiasmo em relação a esta técnica. Tendo em vista o atual avanço tecnológico e as previsões para os próximos desenvolvimentos, este fato tem uma relevância significativamente decrescente.

Alternativas e outros detalhes serão mais explorados nos capítulos de sugestões e conclusões como forma de evidenciar tendências e possíveis soluções, bem como para elucidar outras perspectivas.

5.2 TESTES OPERACIONAIS DA PLATAFORMA

Nesta seção apresenta-se a fase inicial do trabalho que consistiu no acoplamento dos diversos módulos que compõem a plataforma e no desenvolvimento de uma função de avaliação aceitável. Os testes para verificação da funcionalidade da plataforma foram feitos utilizando o controlador PD desenvolvido em [9], porém sem considerar todas as especificações de projeto.

5.2.1 Considerações para o Desenvolvimento da Função de Avaliação

A função de avaliação determina uma medida para o desempenho do controlador projetado com os ganhos especificados pelo algoritmo genético, ou em outras palavras, uma *nota*. Por exigir conhecimento conceitual, seu projeto apresenta um certo grau de dificuldade, que aumenta conforme o número necessário de variáveis avaliadas. Para garantir desempenho e estabilidade, também deve-se levar em conta os aspectos considerados relevantes na resposta do controlador.

O objetivo da utilização de um otimizador é a obtenção de um controlador que satisfaça as condições gerais de operação. Assim, a avaliação deve ser baseada em variáveis que caracterizem uma análise global. O tempo de acomodação (**T_s**), tempo de pico (**T_p**), tempo de subida (**T_r**), “overshoot” percentual (**M_p**) e as especificações do erro da resposta em regime nominal nos fornecem medidas capazes de descrever o desempenho do controlador conforme sugerido por [57] [58] [85] e outros. A Figura 5.5 apresenta uma representação ilustrativa dessas variáveis para um sistema de segunda ordem.

Apesar de sua importância, estas variáveis não são suficientes para se projetar um controlador satisfatório. Entre outras razões, estas variáveis podem ser conflituosas entre si, por exemplo, ao se minimizar o “overshoot”, a tendência é de se aumentar o

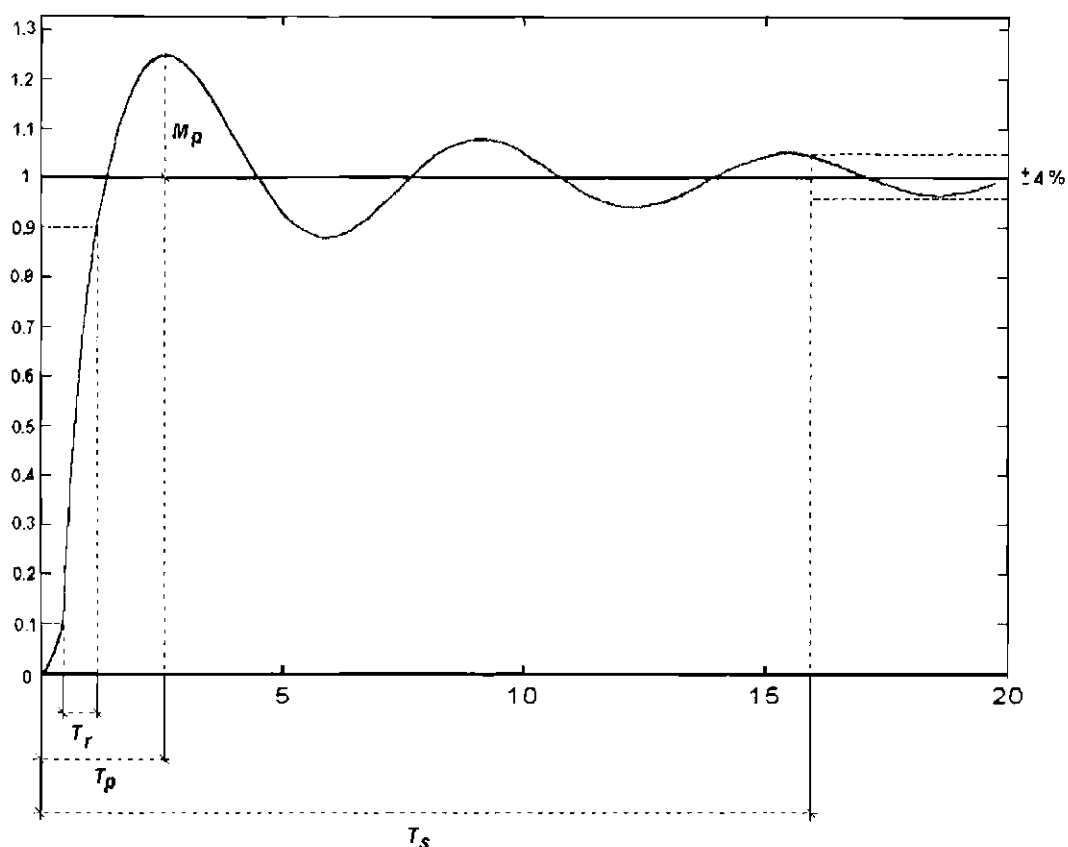


Fig. 5.5 - Resposta passo de um sistema de segunda ordem.

tempo de acomodação. Nestes casos, é necessário se encontrar um ponto de equilíbrio a contento. Além disso, elas não conseguem representar o comportamento oscilatório (estabilidade) da resposta.

Desta maneira, é preciso relacionar estas variáveis em uma função que forneça uma “medida” da qualidade parcial resultante, ou seja, somente do desempenho. De outra forma, deve-se fazer um estudo similar para garantir as exigências de estabilidade requeridas na resposta.

Ainda segundo [57] [85], dentre tantos outros autores, uma condição necessária e suficiente para que o sistema seja estável é que todos os pólos da função de transferência tenham parte real negativa, ou seja, os pólos devem ficar no semi-plano esquerdo do plano “ s ” para que o sistema seja estável. Uma maneira de verificar a estabilidade

do sistema sem calcular explicitamente os pólos, é através do método de estabilidade de Routh-Hurwitz que fornece um mecanismo estruturado para determinar o número de pólos instáveis da equação característica de malha fechada.

A união dessas análises numa mesma função por meio da agregação de todas as variáveis deve produzir o conceito global do sistema de controle projetado, descartando as respostas não satisfatórias dando-lhes notas pouco significativas e valorizando as melhores respostas, tanto quanto mais se aproximarem do ótimo. Isso deve possibilitar o desenvolvimento de um sistema que atenda às condições de desempenho e estabilidade requeridas, garantindo as características necessárias e a confiabilidade.

É importante notar que uma vez encontrada uma função de avaliação para avaliar um tipo de controlador, ela poderá ser utilizada para avaliar qualquer outro tipo de controlador produzido por um algoritmo genético.

Neste trabalho em particular, uma mesma função de avaliação é utilizada em um algoritmo genético que aprende os ganhos de um controlador do tipo PD, e em um algoritmo genético que aprende os termos de um controlador nebuloso.

5.2.2 Testes Iniciais de Funções de Avaliação

Nesta etapa do trabalho teve-se como objetivo maior verificar o funcionamento do conjunto de rotinas acopladas e obter-se uma familiarização com as variáveis envolvidas, sem uma preocupação maior com as soluções encontradas.

Efetivamente, as primeiras tentativas de se desenvolver uma função de avaliação para avaliar os controladores codificados em cada cromossoma, não produziram resultados satisfatórios. Apesar disto, estas tentativas são apresentadas em seguida, para se mostrar a evolução do trabalho.

O modelo do satélite leva em conta as perturbações devidas a três torques, provocados pelo arrasto atmosférico, pressão de radiação solar e os resíduos magnéticos (vide Figura 4.2). Estes torques ocasionam os erros no apontamento do satélite, e a função do controlador é de reagir a estas perturbações enviando um sinal de

comando aos atuadores que, por sua vez, geram uma ação no sentido de corrigir o apontamento ao longo da trajetória orbital.

O controlador PD responde proporcionalmente aos ganhos do erro de posição angular ($\zeta(t)$) e da derivada deste erro ($\Delta\zeta(t)$), que equivale à velocidade angular (ω). O ajuste do controlador é feito através da seleção de ganhos que conduzam a uma resposta satisfatória, ou seja, a função de controle deve manter as variáveis controladas tão próximas quanto possível dos valores desejados.

Inicialmente, considerou-se um número relativamente grande de variáveis, dados pelos erros máximos nos vetores ζ e ω (eixos X, Y, Z) ao longo da órbita e os valores Ts, Tp, Tr, Mp . A primeira função de avaliação assumiu a forma dada pela expressão (Equação 5.1):

$$fitness = \frac{1}{\max(\zeta_X^{\max}, \zeta_Y^{\max}, \zeta_Z^{\max}) * \max(\omega_X^{\max}, \omega_Y^{\max}, \omega_Z^{\max}) * \psi} \quad (5.1)$$

onde $\psi = Ts * Tp * Tr * Mp$, adotando-se também ζ_i^{\max} e ω_i^{\max} definidos como:

$$\begin{aligned} \zeta_i^{\max} &= \max_t(\zeta_i(t)), \quad i \in \{X, Y, Z\} \\ \omega_i^{\max} &= \max_t(\omega_i(t)) \quad i \in \{X, Y, Z\} \end{aligned}$$

Os erros $\zeta_i(t)$ e $\omega_i(t)$, $i \in \{X, Y, Z\}$ são calculados como a diferença entre o vetor de estado atual e o vetor de estado desejado para cada instante simulado.

A concepção desta função baseou-se no fato de que quanto menores os valores Ts, Tp, Tr, Mp , e quanto menores os erros, melhor pode ser considerado o controlador que os gerou (notar que o uso da fração se deve ao fato de que o algoritmo genético adotado é tal que a função de avaliação deve ter valor máximo com o melhor cromossoma).

Esta função apresentou dois problemas importantes. Em primeiro lugar, ela é pouco eficiente, já que a avaliação simultânea de vários parâmetros confere um incremento de dificuldade no processo de busca, além de aumentar o custo computacional devido à complexidade dos cálculos. Mas o principal problema é que as restrições adotadas

permitem a accitação de indivíduos da população pouco qualificados.

Apesar de ocorrer a convergência do algoritmo genético, as respostas para o erro de posição e velocidade angular na simulação de uma órbita (6400 *seg.*), resultaram no melhor dos testes, num controle estável, mas com desempenho inaceitável para os três eixos de coordenadas (X, Y, Z). Mesmo empregando-se um espaço de busca pouco restrito, isto é, infringindo algumas especificações de projeto, os resultados não atenderam aos requisitos básicos, o que se deve à deficiência da função de avaliação na avaliação das variáveis analisadas.

A literatura mostra que, em geral, as funções simples conduzem a melhores resultados. Numa segunda tentativa de se construir uma função de avaliação satisfatória optou-se por utilizar somente as duas variáveis principais (ζ, ω) nos cálculos, visando minimizá-las. Além disso, foram eliminados todos os candidatos à solução que infringissem qualquer uma das especificações de projeto (Capítulo 4):

1) Especificações para o erro de posição (ζ):

- Eliminar cromossomas para os quais $\zeta_X > \nu_X^\zeta = 0.5^\circ$, ou $\zeta_Y > \nu_Y^\zeta = 0.15^\circ$, ou $\zeta_Z > \nu_Z^\zeta = 0.15^\circ$;

2) Especificações para a velocidade angular (ω):

- Eliminar cromossomas para os quais $\omega_X > \nu_X^\omega = 0.05^\circ/s$, ou $\omega_Y > \nu_Y^\omega = 0.05^\circ/s$, ou $\omega_Z > \nu_Z^\omega = 0.05^\circ/s$.

onde ν_i^ζ e ν_i^ω , $i \in \{X, Y, Z\}$ denotam respectivamente as especificações de projeto em relação aos erros angulares de posição e velocidade, em cada um dos três eixos.

Também fixaram-se ganhos derivativos iguais nestes mesmos eixos, com o objetivo de simplificar os cálculos. Desta forma foram necessários encontrar somente quatro ganhos. Os cálculos envolvidos nas rotinas de simulação originais não foram alterados.

Para se verificar a funcionalidade e validade do modelo proposto, foram desenvolvidos testes onde o algoritmo genético faz uso de uma população composta por 50

indivíduos, $p_c = 0.8$, $p_m = 0.008$, um número máximo de 35 gerações possíveis na evolução e uma sensibilidade de três casas decimais para detecção de variação da função de avaliação.

Cada cromossoma, representado por um vetor de 80 posições, codifica 4 ganhos, modelados por 4 genes distintos e de tamanhos iguais entre si. Um intervalo $[0, 15]$ (universo de discurso) consideravelmente maior que o aceitável para a variação dos ganhos foi empregado em cada gene, a fim de verificar o desempenho das funções de avaliação.

▷ **Primeiro teste:**

Foram necessárias somente três gerações para a convergência do AG, com a avaliação dos indivíduos realizada pela função

$$fitness = 1 - \max \left(\frac{\zeta_X^{\max}}{\nu_X^{\zeta}}, \frac{\zeta_Y^{\max}}{\nu_Y^{\zeta}}, \frac{\zeta_Z^{\max}}{\nu_Z^{\zeta}}, \frac{\omega_X^{\max}}{\nu_X^{\omega}}, \frac{\omega_Y^{\max}}{\nu_Y^{\omega}}, \frac{\omega_Z^{\max}}{\nu_Z^{\omega}} \right) \quad (5.2)$$

Os resultados para o erro de posição (ζ) e velocidade angular (ω) na simulação de uma órbita (6400 seg.), podem ser observados nas Figuras 5.6 e 5.7.

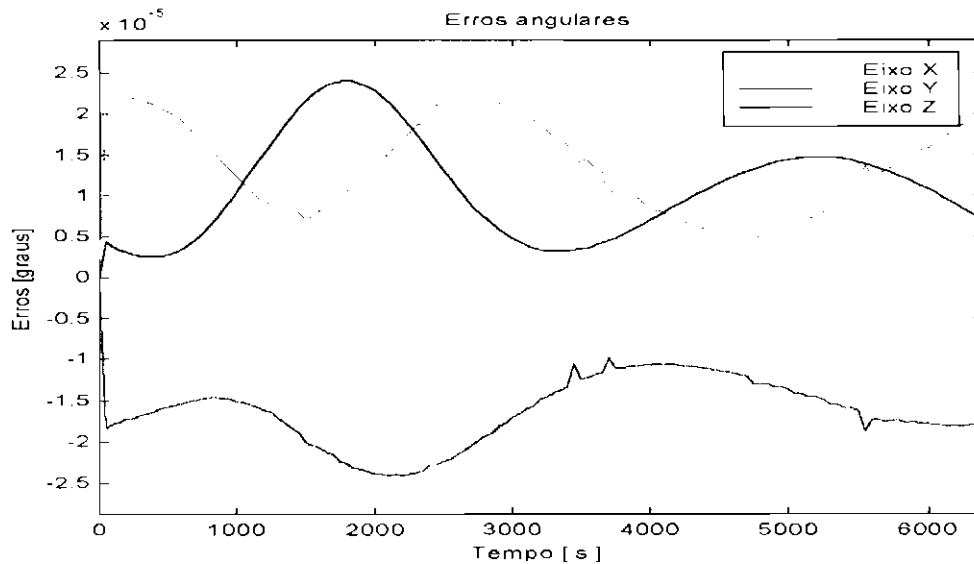


Fig. 5.6 - Erros de posição angular no apontamento do satélite com o controlador otimizado pelo AG, simulado para uma órbita (Teste 1).

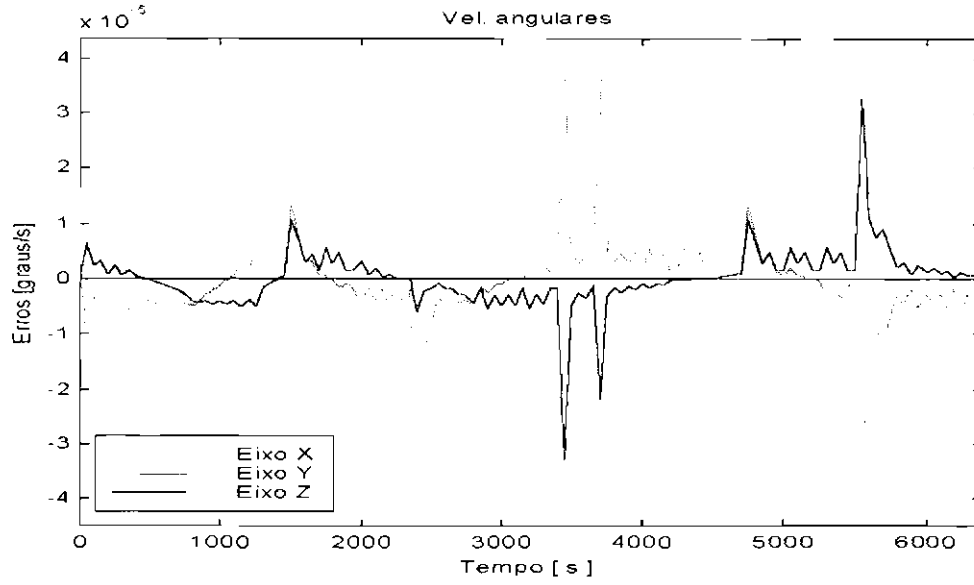


Fig. 5.7 - Erros de velocidade angular desenvolvidos nos eixos X , Y e Z . Simulação de uma órbita (Teste 1).

▷ **Segundo teste:**

Novamente, a convergência foi obtida após três gerações, utilizando a função de avaliação:

$$fitness = 1 - \frac{1}{2} \left(\frac{1}{3} \sum_{i \in \{X,Y,Z\}} \frac{\zeta_i^{\max}}{\nu_i^{\zeta}} + \frac{1}{3} \sum_{i \in \{X,Y,Z\}} \frac{\omega_i^{\max}}{\nu_i^{\omega}} \right) \quad (5.3)$$

$$fitness = 1 - \frac{1}{6} \left(\sum_{i \in \{X,Y,Z\}} \frac{\zeta_i^{\max}}{\nu_i^{\zeta}} + \sum_{i \in \{X,Y,Z\}} \frac{\omega_i^{\max}}{\nu_i^{\omega}} \right)$$

Os resultados da simulação de uma órbita, podem ser observados nas Figuras 5.8 e 5.9.

A concepção destas funções baseou-se no objetivo de se minimizar os erros de posição e velocidade angular. Notar que o uso da subtração se deve, mais uma vez, ao fato de que a função de avaliação do algoritmo genético adotado é tal que deve ter valor máximo com o melhor cromossoma. As constantes nos denominadores das frações

$\frac{\zeta_i^{\max}}{\nu_i^{\omega}}$ e $\frac{\omega_i^{\max}}{\nu_i^{\omega}}$ em ambas as equações têm o papel de conferir a mesma altura máxima para o erro tolerado nos três eixos e de trazer os valores internos das funções para o intervalo $[0, 1]$.

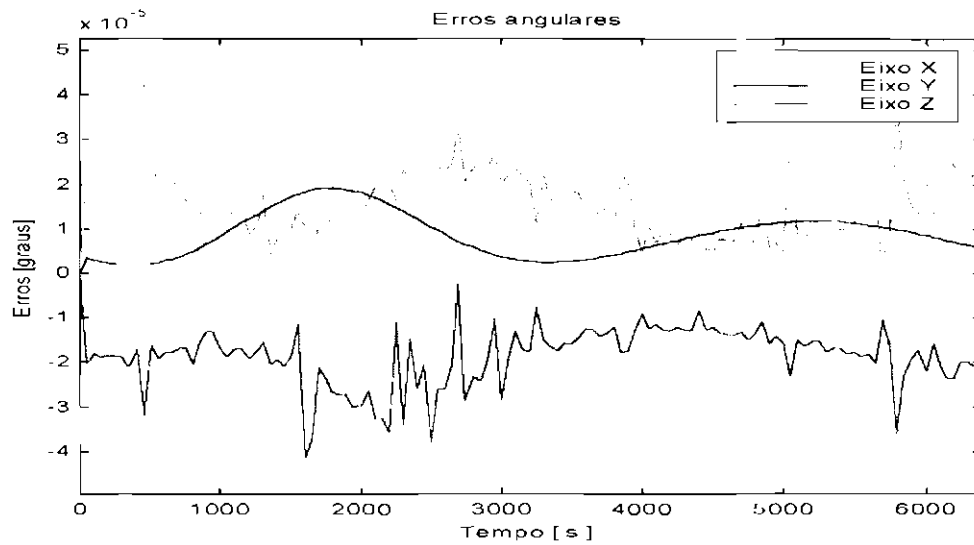


Fig. 5.8 - Erros de posição angular no apontamento do satélite com o controlador otimizado pelo AG, simulado para uma órbita (Teste 2).

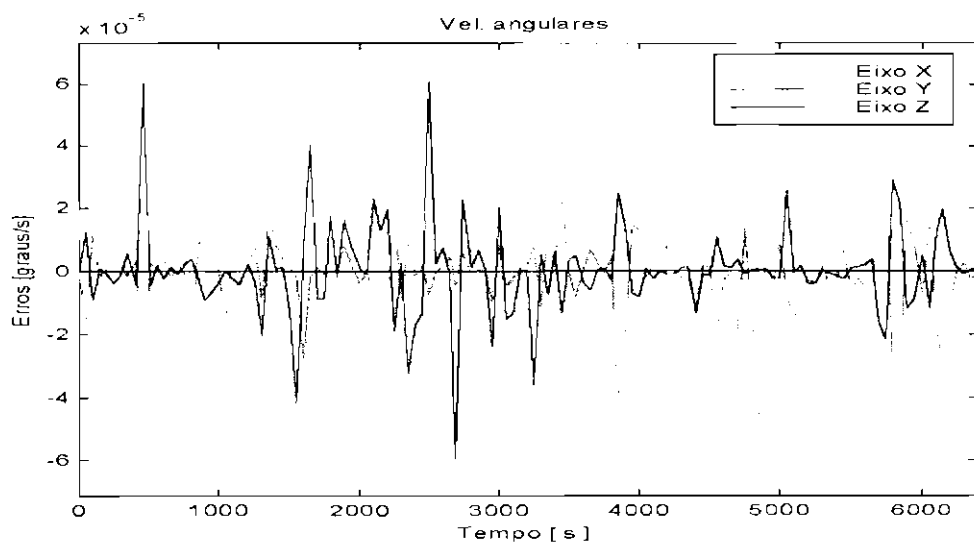


Fig. 5.9 - Erros de velocidade angular desenvolvidos nos eixos X, Y e Z. Simulação de uma órbita (Teste 2).

O segundo termo da subtração na Equação 5.2 obtém o maior erro normalizado e na Equação 5.3 o segundo termo da subtração representa uma média aritmética dos erros.

As Equações 5.2 e 5.3 são, portanto, normalizadas em $[0, 1]$, ou seja, as notas máxima e mínima que um candidato pode obter assumem os valores 1 e 0, respectivamente.

A Tabela 5.1 apresenta os ganhos do controlador PD original e os ganhos encontrados nos dois testes efetuados com o auxílio do AG.

TABELA 5.1 - GANHOS ENCONTRADOS NOS TESTES

VALORES DOS GANHOS			
Ganhos	Original	1º teste	2º teste
$K_{P_X} \quad K_{P_Y}$	0.0263	14.6705	12.9138
$K_{D_X} \quad K_{D_Y}$	0.08	14.1878	4.5851
K_{P_Z}	0.0272	9.1462	11.4086
K_{D_Z}	0.1	3.4182	2.9435

As convergências prematuras do algoritmo genético como observadas nos dois testes não são favoráveis à otimização e geralmente estão relacionadas com o ajuste dos parâmetros e a homogeneidade da população.

Com a finalidade de amenizar este problema, os experimentos seguintes fizeram uso de um maior número de indivíduos na população, o que confere melhor representabilidade do espaço de busca. Foi também efetuado um ajuste dos parâmetros de mutação e cruzamento, e uma modificação dos critérios de parada. Além disso, os parâmetros do modelo de simulação do satélite foram ajustados para conferir maior sensibilidade e realismo na representação da sua atitude em regime nominal.

CAPÍTULO 6

PROJETO DE CONTROLADORES UTILIZANDO UM AG

Após a verificação de funcionalidade no acoplamento dos diversos módulos que integram a plataforma, um algoritmo genético (AG) foi utilizado, para cada controlador desenvolvido no contexto do presente trabalho, na tarefa efetiva de resolver o problema de otimização do controle.

As duas próximas seções trazem respectivamente as condições em que foram projetados os controladores e a função de avaliação adotada no projeto. As seções seguintes apresentam um controlador PD e dois controladores nebulosos desenvolvidos com o auxílio de um algoritmo genético, sob as condições especificadas. Finalmente, na última seção é traçada uma comparação entre os controladores desenvolvidos neste trabalho e o proposto em [9].

6.1 CONDIÇÕES GERAIS DOS EXPERIMENTOS

Para proceder às otimizações efetivas, adotaram-se condições mais severas para o modo de operação do satélite que aquelas adotadas em [9], com a incrementação da ordem de grandeza das perturbações em relação ao modelo original. As perturbações devido ao arrasto atmosférico e à pressão de radiação solar foram multiplicadas por uma constante (β_i), correspondente ao valor necessário para atingir a mesma ordem de grandeza das perturbações devidas ao dipolo magnético residual, que até então possuía ação predominante sobre as demais. Essa operação não altera a *forma* das curvas descritas por essas perturbações, porém confere uma magnitude maior a estas curvas.

A Figura 6.1 ilustra na parte superior a perturbação devido ao campo magnético, na parte central está a perturbação provocada pela pressão de radiação solar e na parte inferior está o arrasto atmosférico.

Em [9], o vetor de estado do satélite no momento inicial representa uma condição

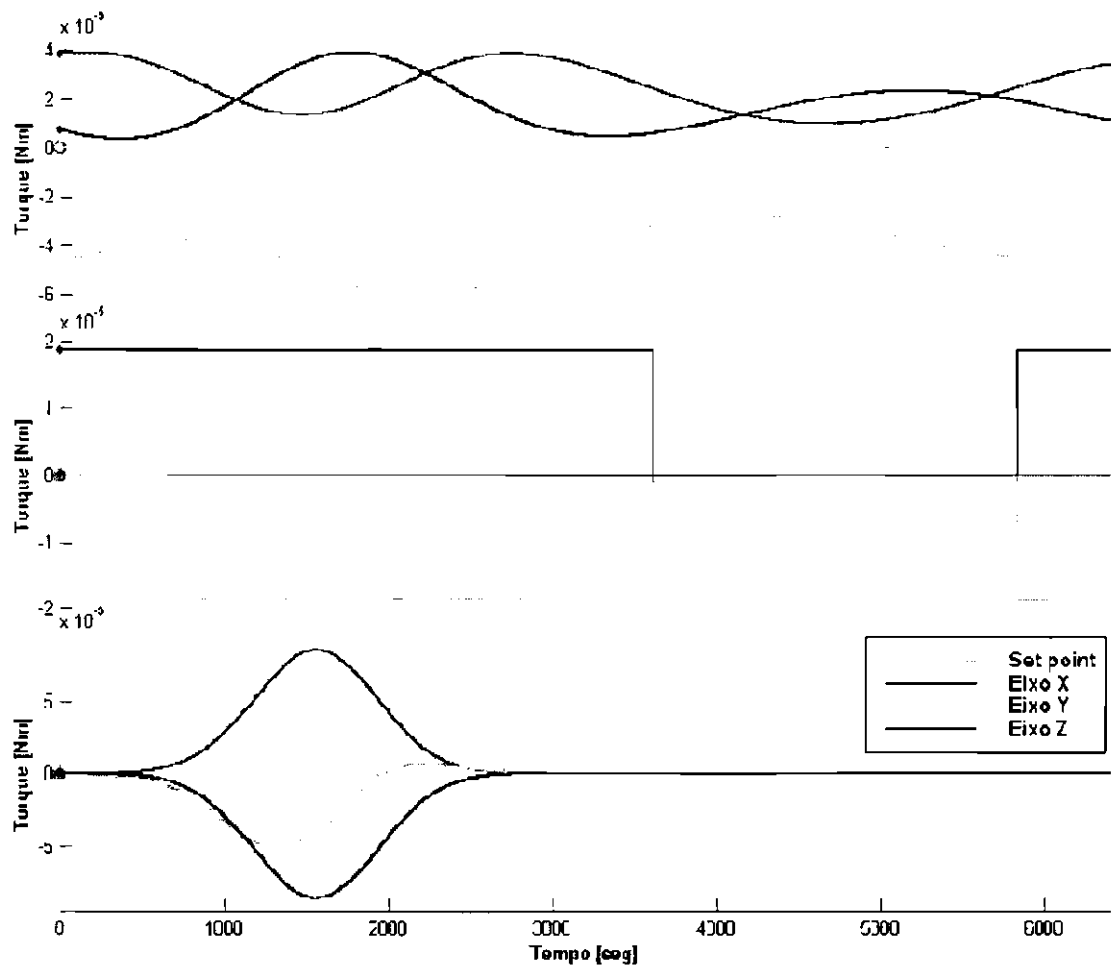


Fig. 6.1 - Torques perturbadores devidos ao dipolo magnético, pressão de radiação e arrasto atmosférico multiplicados por (β_i) .

de erro zero, tanto para o apontamento como para a velocidade. Neste trabalho, um erro inicial próximo aos limites impostos pelos requisitos de projeto foi adotado para a posição dos três eixos, mantendo-se as velocidades como na situação anterior. Os respectivos valores iniciais para os eixos são:

- $X(t_0) = 0.35^\circ$,
- $Y(t_0) = -0.12^\circ$ e
- $Z(t_0) = 0.12^\circ$.

onde t_0 é o instante inicial da simulação.

Para manter o apontamento do satélite dentro dos limites especificados, são adotadas rodas de reação como atuadores. Neste ponto mais uma restrição é imposta, ou seja, o máximo momento angular a ser absorvido pelas rodas é de 0.2 Nm [9], uma característica relevante que precisa ser considerada.

Com a finalidade de possibilitar uma análise de desempenho, a Figura 6.2 mostra a simulação de uma órbita, onde são projetados os erros de atitude relativos às ações de controle, calculadas conforme as leis de controle propostas em [9]. Estes valores servirão como uma referência para efetuar uma avaliação de desempenho, comparando com os dados obtidos através do uso dos controladores projetados com auxílio do algoritmo genético.

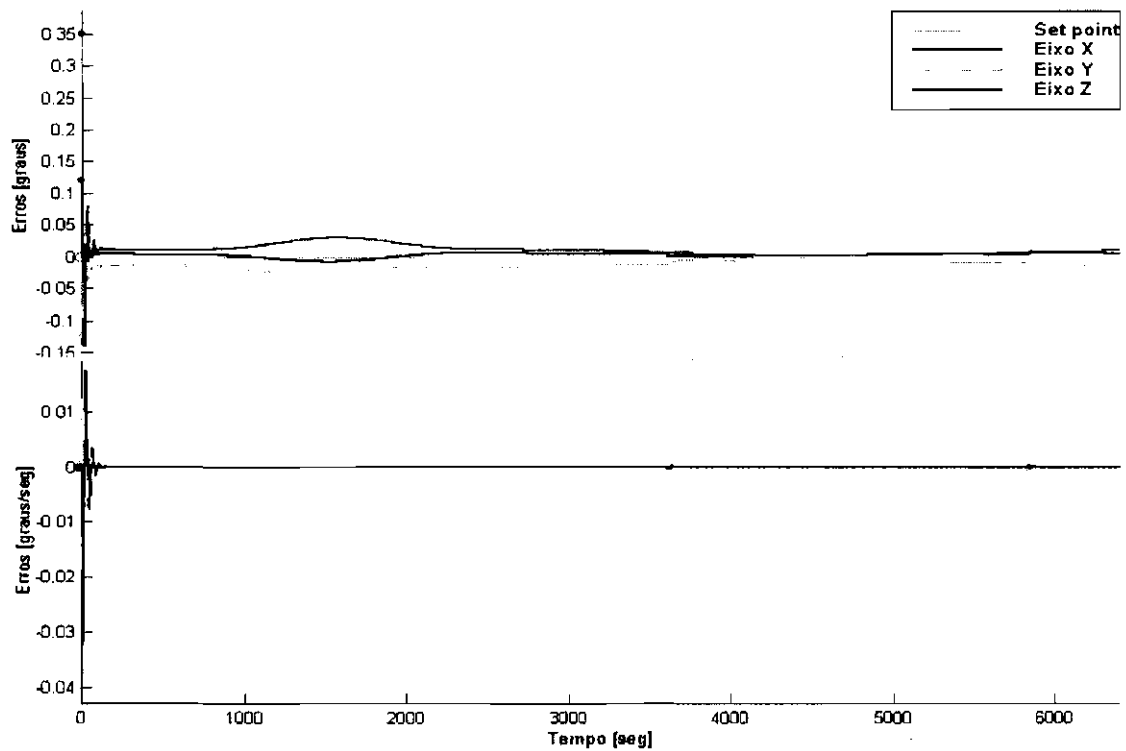


Fig. 6.2 - Erros angulares de posição e velocidade para referência nas novas simulações.

A Figura 6.3 é um “zoom” dos primeiros 1600 segundos de simulação.

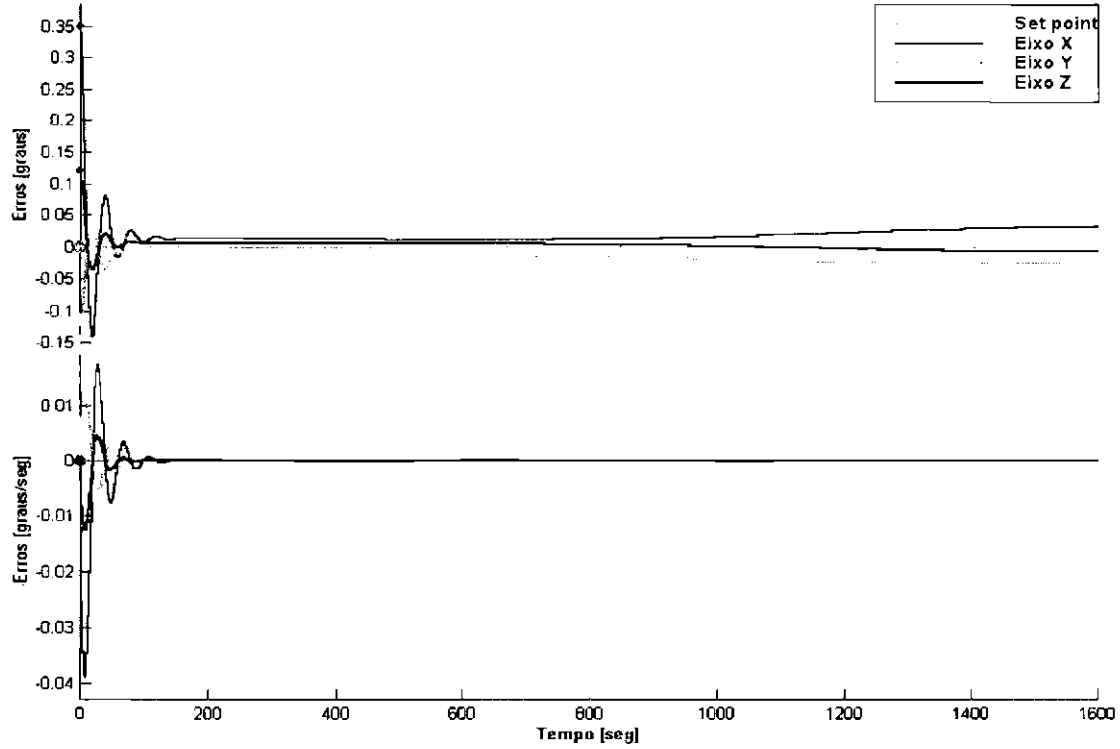


Fig. 6.3 - Erros angulares com PID original na simulação de 1600 segundos.

6.2 FUNÇÃO DE AVALIAÇÃO

Para proceder às otimizações efetivas, foi adotada uma função de avaliação derivada da Equação 5.3, onde ao invés da média dos erros máximos, utiliza-se a integral dos erros angulares ao longo da órbita:

$$fitness = 1 - \frac{1}{2} \left(\frac{1}{3} \sum_{i \in \{X,Y,Z\}} \frac{\int_{t_0}^{t_{max}} \zeta_i(t) dt}{\nu_i^\zeta} + \frac{1}{3} \sum_{i \in \{X,Y,Z\}} \frac{\int_{t_0}^{t_{max}} \omega_i(t) dt}{\nu_i^\omega} \right) \quad (6.1)$$

$$fitness = 1 - \frac{1}{6} \left(\sum_{i \in \{X,Y,Z\}} \frac{\int_{t_0}^{t_{max}} \zeta_i(t) dt}{\nu_i^\zeta} + \sum_{i \in \{X,Y,Z\}} \frac{\int_{t_0}^{t_{max}} \omega_i(t) dt}{\nu_i^\omega} \right)$$

onde t_{max} é o instante final de simulação. A integral $\int_{t_0}^{t_{max}} f(t) dt$ de uma função $f(t)$ é obtida pela função *trapz.m* do Matlab, a qual efetua o cálculo de uma integral

numérica através do método da soma trapezoidal.

Para proceder às otimizações, manteve-se uma coleção de valores para os parâmetros do AG em todos os testes: população composta por 30 indivíduos, $p_c = 0.9$, $p_m = 0.033$, sensibilidade de três casas decimais para detecção de variação da função de avaliação e como último critério de parada, um número máximo de 35 gerações na evolução.

Os cromossomas representados por vetores são separados em genes distintos, mas de tamanhos iguais entre si (número de campos = 20). O tamanho total do cromossoma varia de acordo com o número de parâmetros codificados, assim como o domínio atribuído a cada gene também varia conforme o parâmetro correspondente.

Os experimentos foram realizadas de maneira independente de um modelo para outro, isto é, os resultados de um modelo otimizado não influenciam a otimização de outro.

Na busca pelos parâmetros de cada controlador, o processo de otimização evolui somente 1/4 do período de uma orbita para avaliação, correspondente ao tempo de 1600 segundos.

6.3 PROJETO DE CONTROLADOR PD

Sob as condições estipuladas acima, foi utilizado um algoritmo genético com a função de avaliação dada na Equação 6.1 para encontrar os ganhos de um controlador PD para a fase de apontamento do satélite.

Os cromossomas foram compostos por 6 genes de 20 campos cada um para codificar valores no domínio $[0, 2]$, representando os ganhos proporcional e derivativo para a lei de controle de cada eixo. O algoritmo convergiu ao final de 23 gerações como mostrado na Figura 6.4.

Os ganhos obtidos pelo algoritmo genético estão mostrados na Figura 6.5 e os resultados da simulação utilizando os ganhos obtidos estão ilustrados na Figura 6.6.

Cost Function encoding as binary successful.

Generation	Maximum	Minimum	Mean	Std. dev.
0	0.969004	1e-006	0.156247	0.35557
1	0.968879	1e-006	0.856253	0.23629
2	0.966183	1e-006	0.738438	0.376438
3	0.970991	1e-006	0.90075	0.171807
4	0.971408	1e-006	0.80616	0.323134
5	0.979169	1e-006	0.745824	0.380573
6	0.976064	1e-006	0.798994	0.321628
7	0.979183	1e-006	0.843444	0.287478
8	0.977356	1e-006	0.90829	0.175362
9	0.979027	1e-006	0.790654	0.360621
10	0.974623	1e-006	0.856185	0.291328
11	0.975422	1e-006	0.861538	0.292683
12	0.978428	1e-006	0.926679	0.17573
13	0.980703	1e-006	0.860545	0.292738
14	0.976756	1e-006	0.858154	0.291489
15	0.976741	1e-006	0.865384	0.293664
16	0.977084	1e-006	0.798343	0.363867
17	0.975511	0.846402	0.948151	0.0364458
18	0.981839	1e-006	0.881174	0.242647
19	0.981832	1e-006	0.816887	0.32788
20	0.981057	1e-006	0.919223	0.176098
21	0.981844	1e-006	0.827541	0.330816
22	0.982147	1e-006	0.830412	0.331997
23	0.982543	1e-006	0.731808	0.411303

Genetic algorithm converged.

Fig. 6.4 - Informações do AG na otimização do controlador PD.

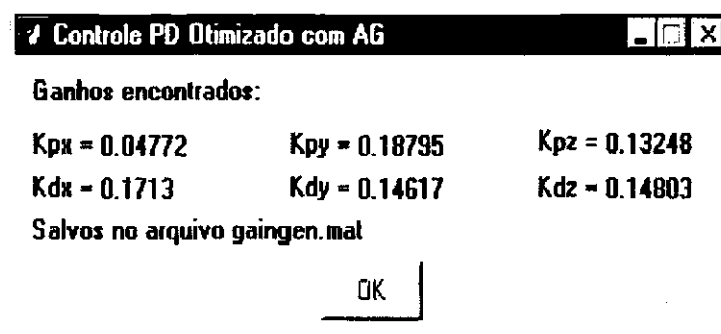


Fig. 6.5 - Ganhos para o controlador PD, obtidos através do uso de um AG.

A representação gráfica permite verificar que o controlador desenvolvido com o auxílio do AG apresenta uma variação menor nas respostas de erro quando comparado ao controlador PD não otimizado. Já nos primeiros instantes o número de oscilações diminui e consegue-se aproximar os erros a zero, mantendo-os assim até o final da simulação.

A Figura 6.7 é um “zoom” dos primeiros 1600 segundos de simulação.

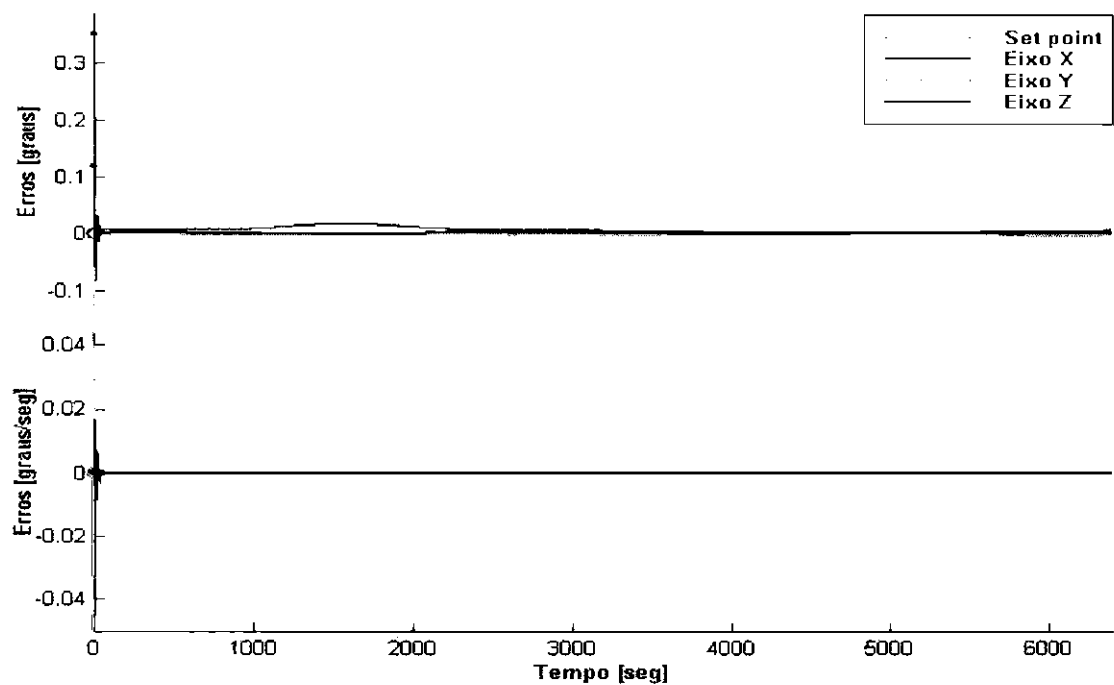


Fig. 6.6 - Erros obtidos com uso do controlador PD otimizado.

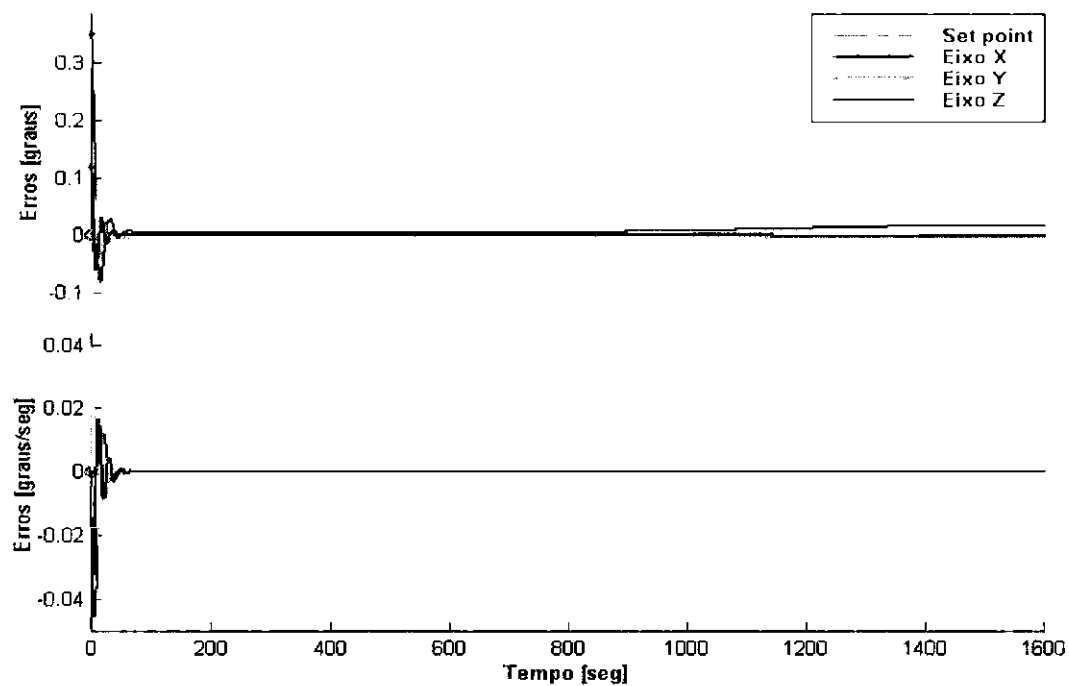


Fig. 6.7 - Erros angulares com PD otimizado na simulação de 1600 segundos.

6.4 PROJETO DE CONTROLADORES NEBULOSOS

A função de avaliação da Equação 6.1, foi utilizada também para projetar controladores nebulosos para a fase de apontamento do satélite, através do uso de um algoritmo genético.

Em um primeiro momento, foram concebidas arquiteturas tanto para controladores nebulosos do tipo Sugeno, quanto para os do tipo Mamdani. Como estes últimos apresentaram resultados iniciais mais promissores, decidiu-se concentrar os esforços no seu desenvolvimento, em detrimento dos primeiros.

Nesta seção são discutidas primeiramente algumas questões relacionadas com o projeto de um controlador nebuloso qualquer. Em seguida, são apresentados os dois controladores nebulosos do tipo Mamdani contruídos no contexto deste trabalho, utilizando-se um algoritmo genético.

6.4.1 Parâmetros, Sintonização e Monitoração de Controladores Nebulosos

De uma maneira geral, pode-se descrever as tarefas de construção de um controlador nebuloso brevemente como [75]:

- 1) Definição do modelo e das características operacionais, para estabelecer as particularidades da arquitetura do sistema (como sensores e atuadores) e definição das propriedades operacionais do controlador nebuloso do projeto, como o tipo de controlador, operadores a serem utilizados, defuzificador, etc;
- 2) Definição dos termos nebulosos de cada variável. Para garantir suavidade e estabilidade deve-se permitir que cada conjunto nebuloso se superponha aos vizinhos;
- 3) Definição do comportamento do controle, que envolve a descrição das regras que atrelam as variáveis de entrada às propriedades de saída do modelo.

No projeto de controladores nebulosos é necessária portanto a definição de alguns parâmetros, obtidos a partir da experiência do projetista ou através de experimentos. Tendo em vista um determinado processo, alguns destes parâmetros são fixos, os denominados parâmetros estruturais, e outros, os parâmetros de sintonização, são aqueles que variam com o tempo:

1) Parâmetros estruturais:

- Número de variáveis de entrada e saída;
- Variáveis linguísticas;
- Funções de pertinência parametrizadas;
- Intervalos de discretização e normalização;
- Estrutura da base de regras;
- Conjunto básico de regras;
- Recursos de operação sobre os dados de entrada.

2) Parâmetros de sintonização:

- Universo de discurso das variáveis;
- Parâmetros das funções de pertinência (altura, largura ou conjunto suporte);
- Ganhos e “offset” das entradas e saídas.

Propriedades da base de regras como a completude, consistência, interação e robustez precisam ser testadas. A robustez relaciona-se com a sensibilidade do controle frente a ruídos ou algum comportamento incomum não-modelado. Para ser medida, introduz-se um ruído aleatório de média e variância conhecidas e observa-se então, a alteração dos valores das variáveis de saída.

A sintonização é uma tarefa complexa devido a flexibilidade que decorre da existência de muitos parâmetros, exigindo esforço do projetista na obtenção do melhor desempenho do controlador. Alguns dos parâmetros podem ser alterados por mecanismos

automáticos de adaptação e aprendizado, contudo, normalmente é tarefa do projetista o treinamento e a sintonia da maioria dos parâmetros. Esta sintonização é feita através de busca, uma atividade típica em Inteligência Artificial.

A sintonização pode ser feita da seguinte maneira:

- 1) Desenvolve-se um controlador simples, que simule um controlador proporcional com:
 - Ganho reduzido;
 - As variáveis mais relevantes;
 - Baixo número de variáveis linguísticas;
- 2) Aumenta-se o ganho até verificar respostas indesejáveis;
- 3) Incrementa-se o conhecimento conforme a experiência resultante do processo:
 - Buscando-se novas variáveis linguísticas ou físicas para contornar as dificuldades;
 - Ajustando-se as funções de pertinência e os parâmetros do controlador;
 - Adicionando-se regras ou modificando a estrutura de controle.
- 4) Valida-se a coerência do conhecimento incorporado com novas condições de operação para o sistema.

Essas tarefas necessitam de plataformas sofisticadas, com interface poderosa e que permitam uma rápida inferência. Isto é proporcionado pelos pacotes integrados, dedicados a análise de modelos nebulosos [78].

6.4.2 Controladores Nebulosos Projetados pelo AG

Dois controladores nebulosos do tipo Mamdani foram implementados com o auxílio de um AG. No projeto dos diferentes tipos de controladores empregou-se o Toolbox de Controle Fuzzy para ambiente Matlab.

As estruturas iniciais dos controladores adotaram os mesmos critérios, e uma base de regras com um número fixo de regras e de termos nebulosos foi construída para cada caso.

A tarefa do algoritmo genético consistiu então em se encontrar os centros dos conjuntos nebulosos das premissas e conclusões de cada regra.

Para conferir maior suavidade à superfície que representa as ações de controle, foi utilizada a sobreposição dos conjuntos nebulosos. O ponto de centro de um conjunto corresponde respectivamente ao ponto final da base do conjunto anterior e ao ponto inicial da base do próximo conjunto (Figura 6.8).

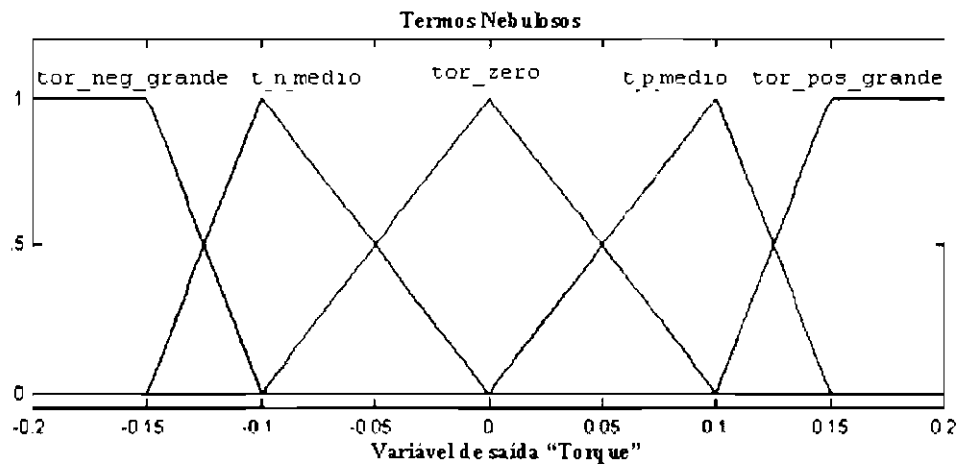


Fig. 6.8 - Conjuntos nebulosos para o mapeamento das conclusões das regras de um controlador nebuloso (ilustração).

Todos os conjuntos nebulosos centrais são de forma triangular e nas extremidades utilizam-se conjuntos trapezoidais. Além disso, adotou-se a simetria em relação ao valor zero de uma variável ($erro = 0$). Desta forma, os conjuntos nebulosos são projetados para os valores positivos da variável e então *espelhados* para os valores negativos correspondentes.

Essas características não representam limitações, já que são comumente encontradas e adotadas na literatura conforme [14] [86] [53] [35] e outros.

Nestes experimentos os cromossomas do AG codificam os valores relacionados aos possíveis centros dos conjuntos nebulosos, mapeados no domínio correspondente conforme as especificações de precisão no apontamento. De forma que os limites para a busca dos centros das variáveis erros angulares de posição e velocidade correspondem a $[0, 0.5]$ e $[0, 0.05]$, respectivamente. Para a variável de saída torque o domínio é o intervalo $[0, 0.2]$, os quais representam os maiores valores tolerados.

Os parâmetros obtidos na aprendizagem com o AG são tratados como porcentagens do intervalo de busca, tomando-se como referência o centro definido em zero. Assim o primeiro conjunto terá como centro a própria porcentagem encontrada, o segundo será o centro anterior mais a porcentagem obtida com o AG e assim sucessivamente.

6.4.2.1 Controlador de Mamdani com 3 termos nebulosos de saída

O primeiro controlador nebuloso projetado do tipo Mamdani, compõe-se de três conjuntos nebulosos para cada variável, de entrada ou saída. A Figura 6.9 traz as informações da eficiente convergência do algoritmo genético.

Cost function encoding as binary successful.				
Fitness statistics				
Generation	Maximum	Minimum	Mean	Std. dev.
0	0.982716	1e-006	0.0980872	0.299289
1	0.982717	1e-006	0.325936	0.468826
2	0.982766	1e-006	0.780132	0.396807
3	0.981201	1e-006	0.772799	0.394293
4	0.981109	1e-006	0.745607	0.418397
5	0.981105	1e-006	0.842219	0.336035
6	0.98352	1e-006	0.674748	0.449822

Genetic algorithm converged.

Fig. 6.9 - Primeiro modelo do tipo Mamdani: Informações do processo de convergência do AG na aprendizagem do controlador.

Deste processo de busca, os melhores centros encontrados para os conjuntos nebulosos podem ser observados na Figura 6.13.

A estrutura da base de regras deste controlador é apresentada a seguir e ilustrada na Tabela 6.4.2.1.

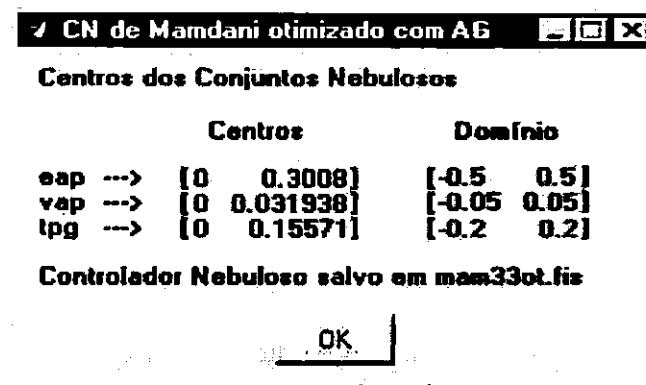


Fig. 6.10 - Primeiro modelo do tipo Mamdani: Centros dos conjuntos nebulosos encontrados pelo AG.

1. If *erro_pos_ang* is *pos_neg* and *erro_vel_ang* is *vel_neg* then *torque* is *tor_pos_grande*
2. If *erro_pos_ang* is *pos_neg* and *erro_vel_ang* is *vel_zero* then *torque* is *tor_pos_grande*
3. If *erro_pos_ang* is *pos_neg* and *erro_vel_ang* is *vel_pos* then *torque* is *tor_zero*
4. If *erro_pos_ang* is *pos_zero* and *erro_vel_ang* is *vel_neg* then *torque* is *tor_pos_grande*
5. If *erro_pos_ang* is *pos_zero* and *erro_vel_ang* is *vel_zero* then *torque* is *tor_zero*
6. If *erro_pos_ang* is *pos_zero* and *erro_vel_ang* is *vel_pos* then *torque* is *tor_neg_grande*
7. If *erro_pos_ang* is *pos_pos* and *erro_vel_ang* is *vel_neg* then *torque* is *tor_zero*
8. If *erro_pos_ang* is *pos_pos* and *erro_vel_ang* is *vel_zero* then *torque* is *tor_neg_grande*
9. If *erro_pos_ang* is *pos_pos* and *erro_vel_ang* is *vel_pos* then *torque* is *tor_neg_grande*

**TABELA 6.1 - REGRAS DO PRIMEIRO CN DO TIPO
MAMDANI**

BASE DE REGRAS			
Posição/VELOCIDADE	Negativa	Zero	Positiva
Negativa	torque_pos_grande	torque_pos_grande	torque_zero
Zero	torque_pos_grande	torque_zero	torque_neg_grande
Positiva	torque_zero	torque_neg_grande	torque_neg_grande

Os conjuntos nebulosos utilizados para fazer o mapeamento das variáveis de entrada e da variável de saída, após a aprendizagem com o AG, são mostrados na Figura 6.11.

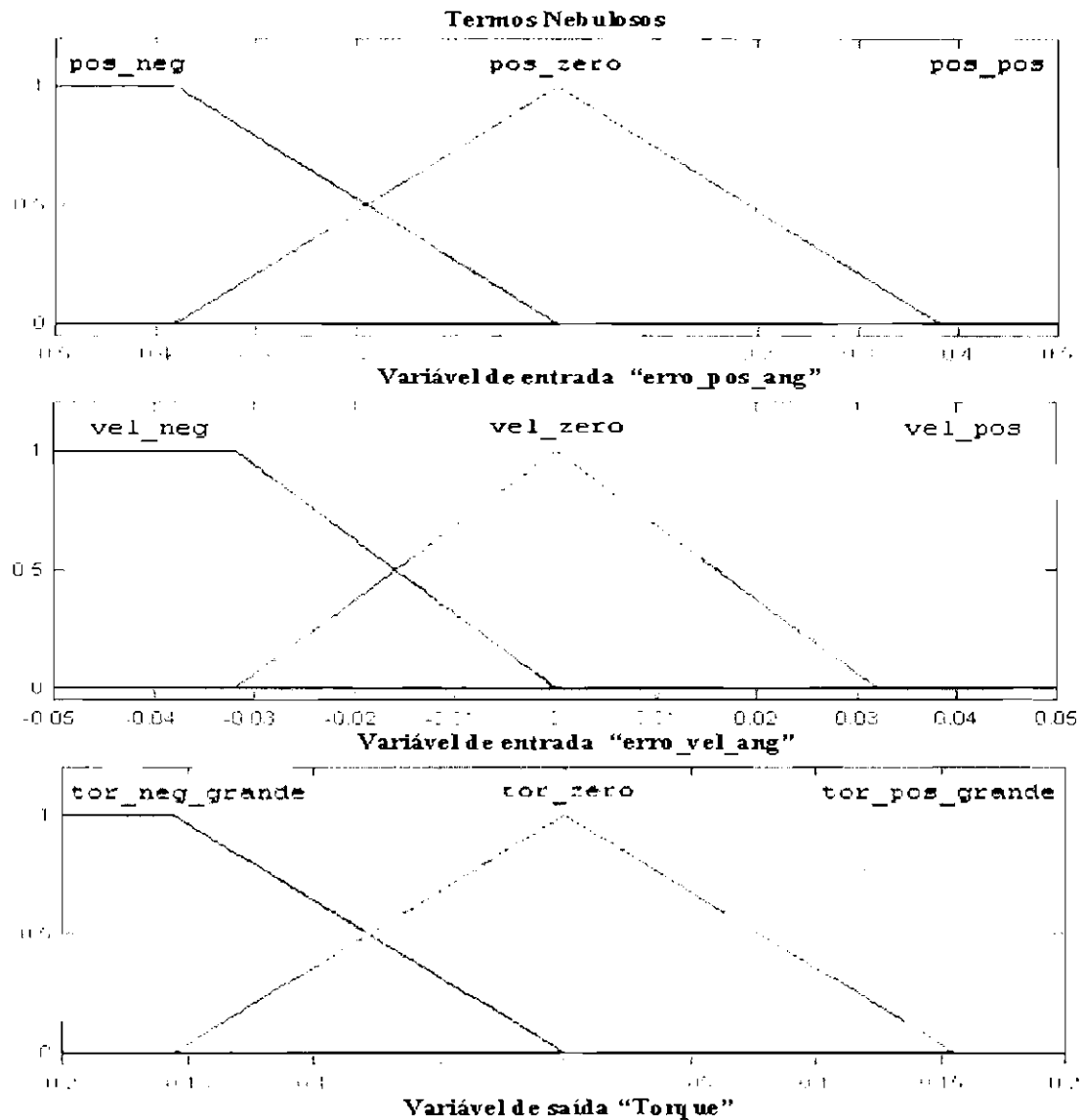


Fig. 6.11 - Primeiro modelo do tipo Mamdani: Termos nebulosos das variáveis de entrada e saída encontrados com AG.

A superfície de controle resultante é mostrada na Figura 6.12, onde são apresentados os torques de saída em função dos correspondentes valores das variáveis de entrada. O desempenho deste controlador pode ser acompanhado na Figura 6.13. A Figura

6.14 é um “zoom” dos primeiros 1600 segundos de simulação.

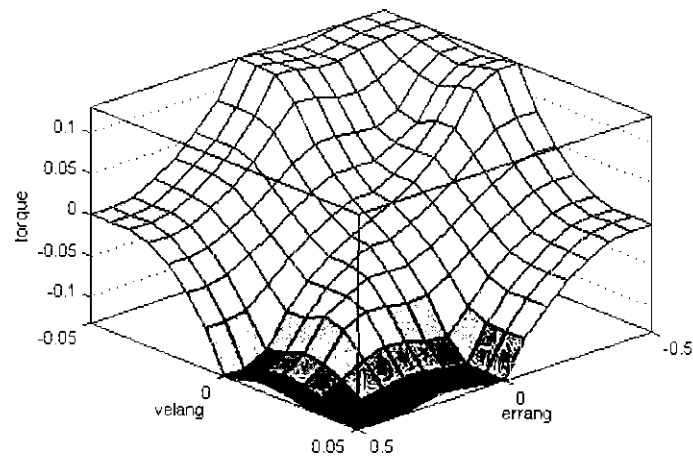


Fig. 6.12 - Primeiro modelo do tipo Mamdani: Superfície de controle resultante do mapeamento da variável de saída torque em função das entradas.

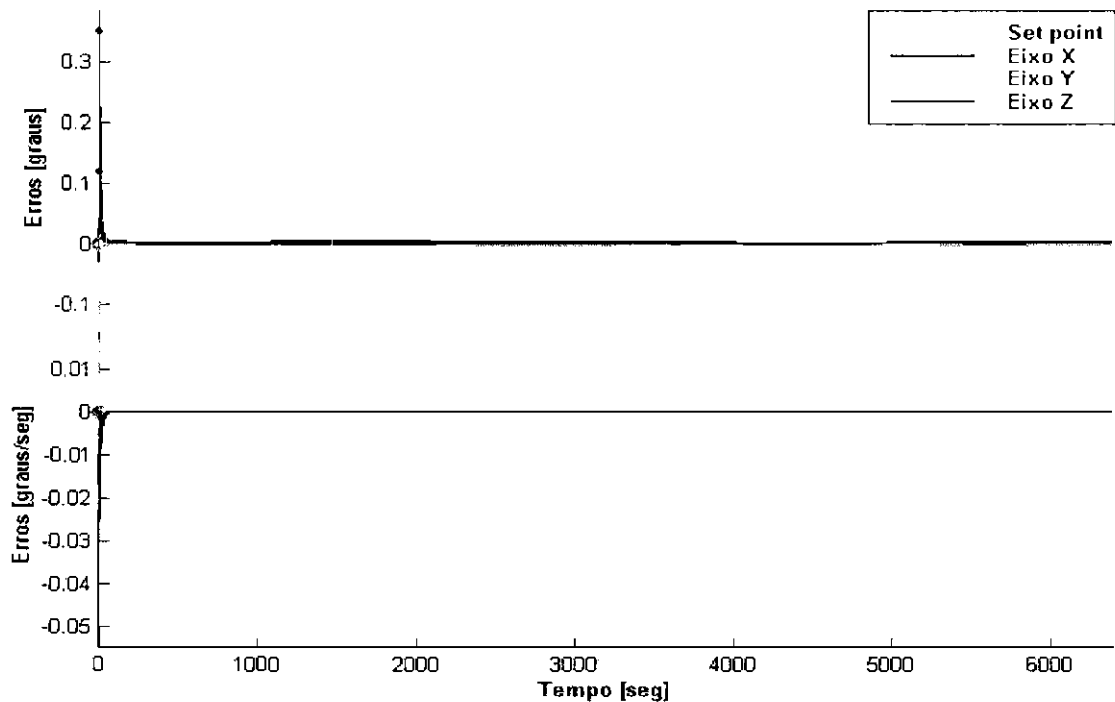


Fig. 6.13 - Primeiro modelo do tipo Mamdani: Erros angulares da simulação de uma órbita.

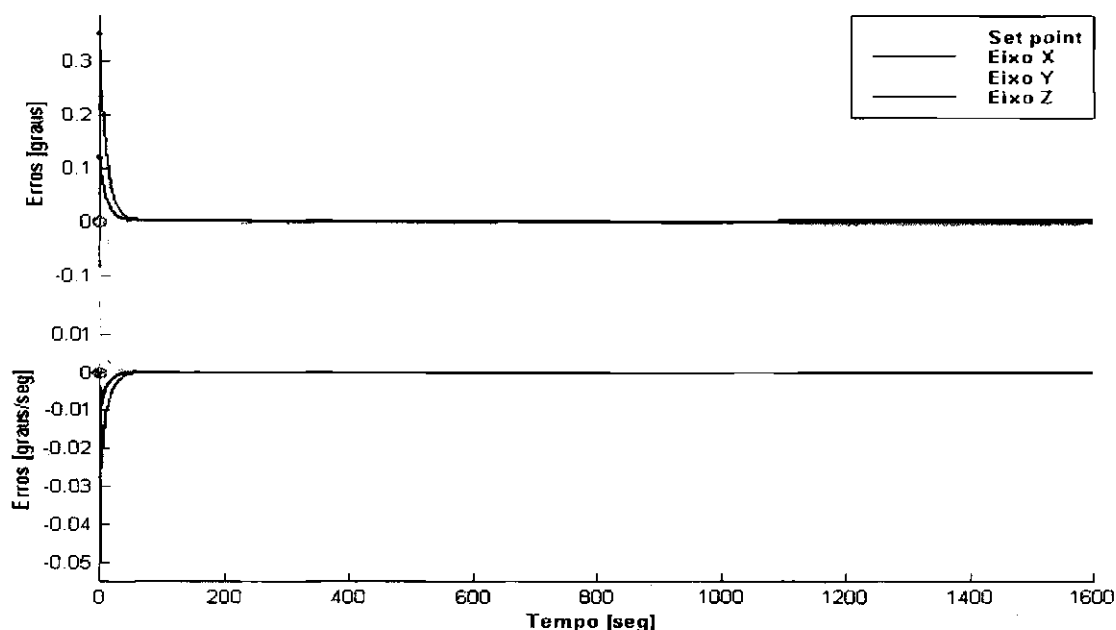


Fig. 6.14 - Primeiro modelo de Mamdani: Erros angulares da simulação de 1600 segundos.

6.4.2.2 Controlador de Mamdani com 5 termos nebulosos de saída

No segundo controlador do tipo Mamdani foram utilizados cinco conjuntos nebulosos para mapear a variável de saída. As duas variáveis de entrada permanecem com três conjuntos nebulosos cada uma. As informações de convergência do algoritmo genético são mostradas na Figura 6.15.

Cost function encoding as binary successful.

Fitness statistics				
Generation	Maximum	Minimum	Mean	Std. dev.
0	0.980585	1e-006	0.220628	0.406964
1	0.980587	1e-006	0.74136	0.380303
2	0.985445	1e-006	0.845815	0.291392
3	0.987191	1e-006	0.632316	0.431655
4	0.987785	1e-006	0.71451	0.405744
5	0.987642	1e-006	0.82661	0.33041
6	0.9877	1e-006	0.705269	0.432995
7	0.987585	1e-006	0.695323	0.427303
8	0.987847	1e-006	0.825813	0.330598

Genetic algorithm converged.

Fig. 6.15 - Segundo modelo do tipo Mamdani: Informações do processo de convergência do AG na aprendizagem do controlador.

Os centros dos conjuntos nebulosos resultantes da busca com o algoritmo genético estão na Figura 6.16.

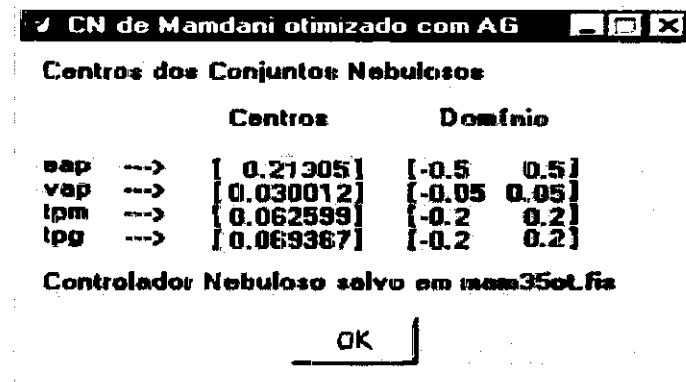


Fig. 6.16 - Segundo modelo do tipo Mamdani: Centros dos conjuntos nebulosos.

A estrutura que compõe a base de regras do segundo controlador nebuloso é apresentada na sequência e sua ilustração é mostrada na Tabela 6.4.2.2. Os conjuntos nebulosos que mapeiam as variáveis de entrada e saída, já otimizados, estão apresentados na Figura 6.17 e a superfície de controle pode ser acompanhada na Figura 6.12.

1. If *erro_pos_ang* is *pos_neg* and *erro_vel_ang* is *vel_neg* then *torque* is *tor_pos_grande*
2. If *erro_pos_ang* is *pos_neg* and *erro_vel_ang* is *vel_zero* then *torque* is *tor_pos_medio*
3. If *erro_pos_ang* is *pos_neg* and *erro_vel_ang* is *vel_pos* then *torque* is *tor_zero*
4. If *erro_pos_ang* is *pos_zero* and *erro_vel_ang* is *vel_neg* then *torque* is *tor_pos_medio*
5. If *erro_pos_ang* is *pos_zero* and *erro_vel_ang* is *vel_zero* then *torque* is *tor_zero*
6. If *erro_pos_ang* is *pos_zero* and *erro_vel_ang* is *vel_pos* then *torque* is *tor_neg_medio*
7. If *erro_pos_ang* is *pos_pos* and *erro_vel_ang* is *vel_neg* then *torque* is *tor_zero*
8. If *erro_pos_ang* is *pos_pos* and *erro_vel_ang* is *vel_zero* then *torque* is *tor_neg_medio*
9. If *erro_pos_ang* is *pos_pos* and *erro_vel_ang* is *vel_pos* then *torque* is *tor_neg_grande*

Os resultados do uso deste controlador estão mostrados na Figura 6.19 e a Figura 6.20 é um “zoom” dos primeiros 1600 segundos de simulação.

**TABELA 6.2 - REGRAS DO SEGUNDO CN DO TIPO
MAMDANI**

BASE DE REGRAS			
Posição/Velocidade	Negativa	Zero	Positiva
Negativa	torque_pos_grande	torque_pos_medio	torque_zero
Zero	torque_pos_medio	torque_zero	
Positiva	torque_zero	torque_neg_medio	torque_neg_grande

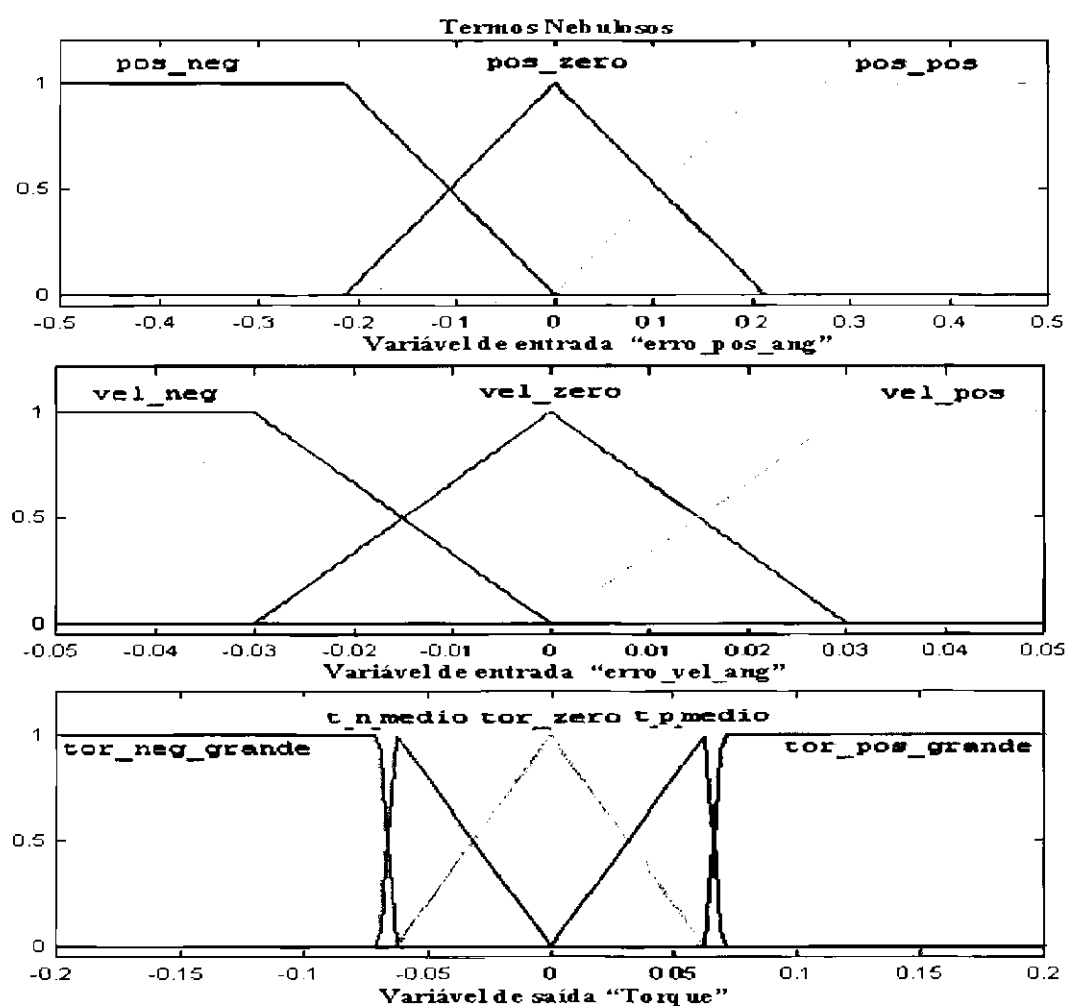


Fig. 6.17 - Segundo modelo do tipo Mamdani: Termos nebulosos das variáveis de entrada e saída encontrados com AG.

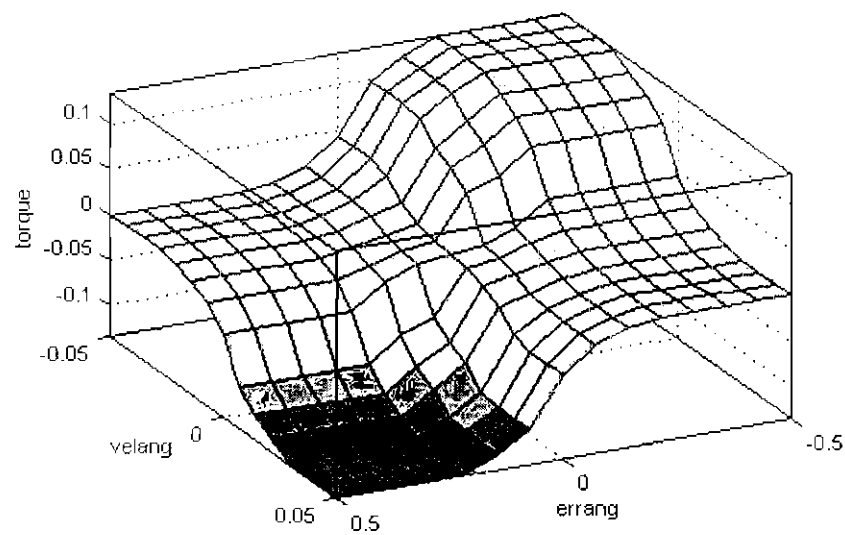


Fig. 6.18 - Segundo modelo do tipo Mamdani: Superfície de controle resultante do mapeamento da variável torque.

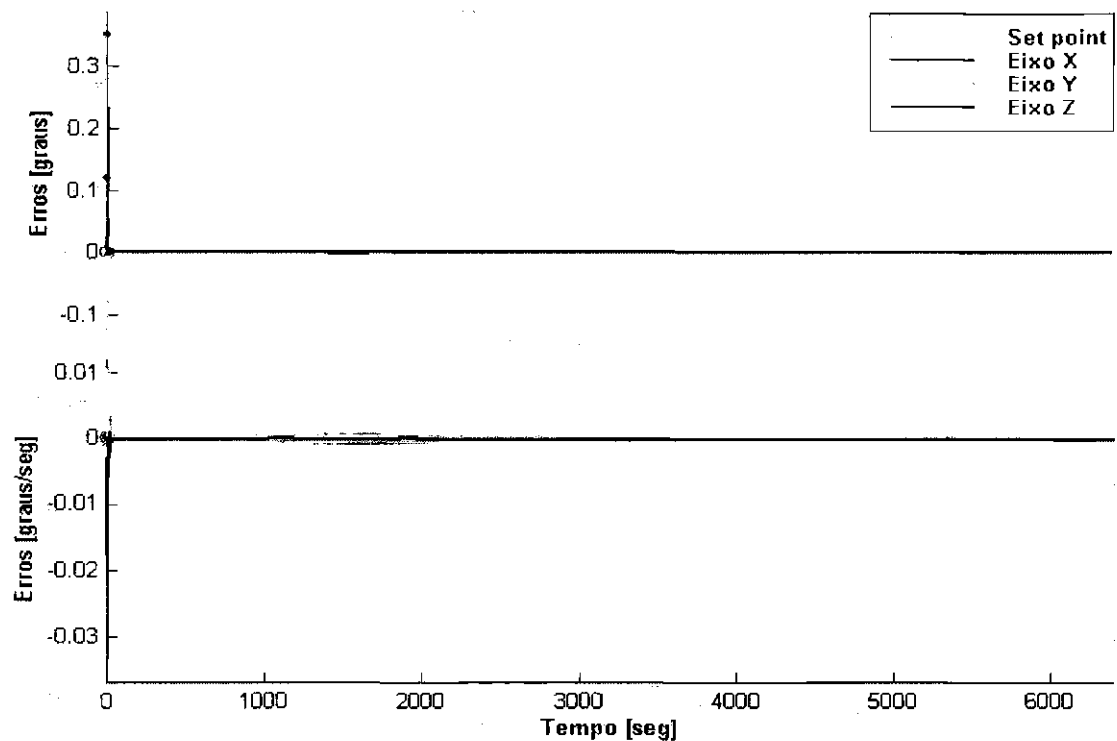


Fig. 6.19 - Segundo modelo do tipo Mamdani: Erros angulares da simulação de uma órbita.

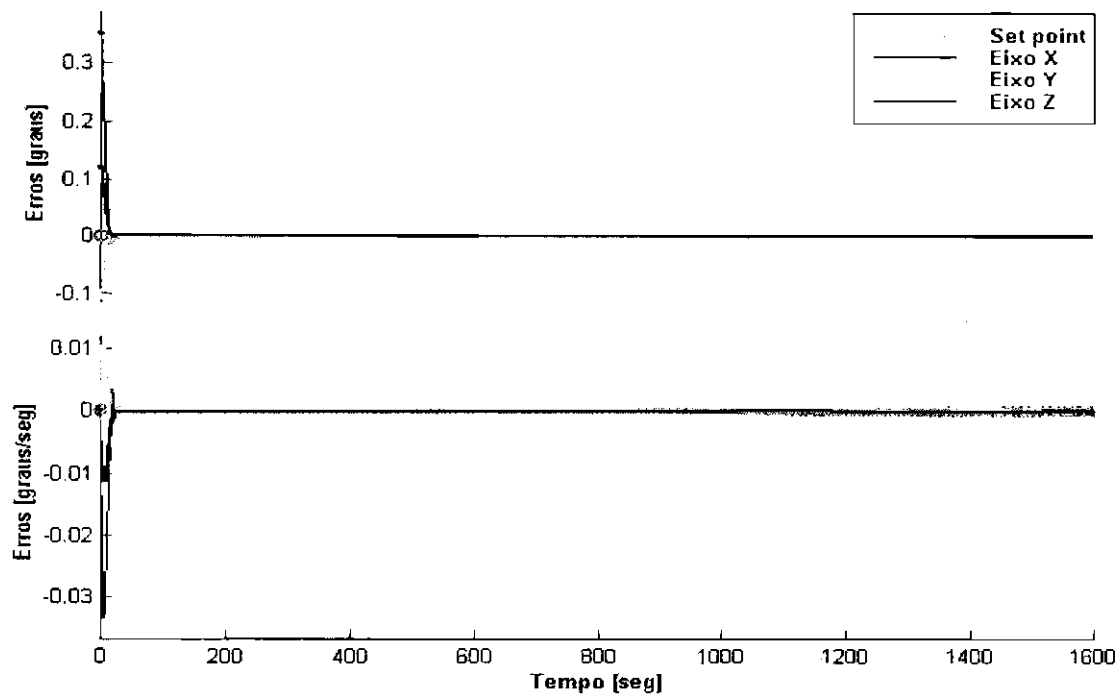


Fig. 6.20 - Segundo modelo de Mamdani: Erros angulares da simulação de 1600 segundos.

6.5 ANÁLISE DOS RESULTADOS

Uma das diferenças significativas entre os controladores desenvolvidos neste trabalho é o fato do controlador PD utilizar três leis de controle, uma para cada eixo, enquanto que nos controladores nebulosos, foi utilizada a mesma base de regras para calcular as ações de saída de cada um dos três eixos.

Para possibilitar a comparação entre os resultados dos erros devidos às respostas de controle em cada uma das simulações realizadas, a Tabela 6.3 apresenta o desempenho dos diversos controladores nos experimentos.

Nota-se na Tabela 6.3 que os controladores desenvolvidos usando AG tiveram desempenho nitidamente melhor que o controlador PD encontrado em [9]. Além disso, nota-se que os controladores nebulosos tiveram geralmente desempenho melhor do que os controladores PD, tanto em relação ao encontrado em [9] quanto em relação

ao PD otimizado com o AG.

**TABELA 6.3 -RESULTADOS DA INTEGRAL DOS ERROS E
SOMA TOTAL DESTA INTEGRAL NOS TRÊS EIXOS**

VALOR ABSOLUTO DA INTEGRAL (Método Trapezoidal)								
Controle	<i>Erros Angulares</i>							
	Posição				Velocidade			
	Eixo X	Eixo Y	Eixo Z	Total	Eixo X	Eixo Y	Eixo Z	Total
PD original	78.1237	84.5526	31.1589	193.8354	0.9383	0.3276	0.2972	1.5632
PD com AG	43.1680	12.6005	7.1152	62.8839	0.6149	0.4463	0.3818	1.4432
a) Mamdani	17.3987	16.5945	7.0337	41.0269	0.3584	0.1260	0.1280	0.6125
b) Mamdani	5.3271	4.1397	2.0544	11.5214	1.4207	1.3178	0.5677	3.3063

Neste sentido, a maior discrepância seria o valor total para o erro em velocidade angular ($erro = 3.3063$) apresentado pelo modelo de Mamdani com 5 conjuntos nebulosos na saída. No entanto, é importante observar que apesar do valor total do erro em velocidade ser superior aos outros modelos, ainda assim este controlador atende às especificações de projeto e possibilita um decréscimo significativo no erro de posição angular ($erro = 12.9957$).

Outra discrepância é que o erro de posição em Y ($erro = 16.5945$) no primeiro modelo de Mamdani, é ligeiramente superior ao erro ($erro = 12.6005$) do mesmo eixo no modelo PD otimizado.

Estas discrepâncias se devem provavelmente à maneira com que foi definida a função de avaliação. Uma análise mais detalhada desta função é feita no final desta seção.

6.5.1 Avaliação da Robustez dos Controladores

Para verificar a robustez dos controladores projetados, foi feita a simulação de uma órbita do satélite adotando-se as mesmas condições das otimizações, porém com uma variação de 30% no momento de inércia do eixo Y (Capítulo 4). Na Tabela

6.4 mostram-se os resultados dos erros angulares para posição e velocidade, obtidos através da simulação de uma órbita, para os controladores: PD original, para o controlador PD obtido pelo AG e para os controladores nebulosos de Mamdani com 3 e 5 conjuntos nebulosos de saída, respectivamente.

TABELA 6.4 -INTEGRAL DOS ERROS, COM VARIAÇÃO DE 30% NO MOMENTO DE INÉRCIA DO EIXO Y

INTEGRAL DOS ERROS COM VARIAÇÃO DE 30% NO MOMENTO DE INÉRCIA DO EIXO Y								
Controle	<i>Erros Angulares</i>							
	Posição				Velocidade			
	Eixo X	Eixo Y	Eixo Z	Total	Eixo X	Eixo Y	Eixo Z	Total
PD original	78.1237	120.1006	31.1589	229.3834	0.9383	0.3339	0.2972	1.5696
PD com AG	43.1680	17.5652	7.1152	67.8485	0.6149	0.44677	0.3818	1.4436
a) Mamdani	17.3987	23.1204	7.0337	47.5528	0.3584	0.1286	0.1280	0.6151
b) Mamdani	5.3263	5.6152	2.0541	12.9957	1.4204	1.8326	0.5675	3.8206

As Figuras 6.21, 6.22, 6.23, 6.24 correspondem a estas simulações. Nelas pode ser observado de forma gráfica a eficiência de cada controlador na tarefa de manter o apontamento com esta nova condição.

A reação dos controladores em relação à variação de massa no eixo Y é semelhante entre os modelos, tendo os controladores apresentado um aumento dos erros para o eixo Y.

Ainda assim, observa-se tanto graficamente, quanto através dos resultados da Tabela 6.4, que as características de estabilidade necessárias durante o regime de operação nominal permanecem praticamente inalteradas. À exceção do controlador PD original, que como nas simulações anteriores, apresenta certa dificuldade para manter os níveis de erros próximos a zero ao final da simulação de uma órbita, todos os outros controladores continuam exercendo sua função de manter o apontamento, mesmo

com essa alteração no momento de inércia.

A princípio, manter os resultados de erro dentro das especificações de projeto, mesmo sob as circunstâncias destes testes, não é uma tarefa de complexidade elevada, já que o satélite modelado é uma estrutura relativamente simples, onde não são considerados painéis flexíveis ou outros apêndices.

Dentre os controladores projetados, observa-se que o segundo modelo de Mamdani é sutilmente mais sensível a variações, conforme verifica-se através dos resultados dos erros para os eixos X e Z que também sofrem uma suave alteração. Mas ao contrário do eixo Y , os valores dos erros angulares de posição e velocidade nestes outros eixos são ainda menores quando comparados aos valores apresentados na Tabela 6.3.

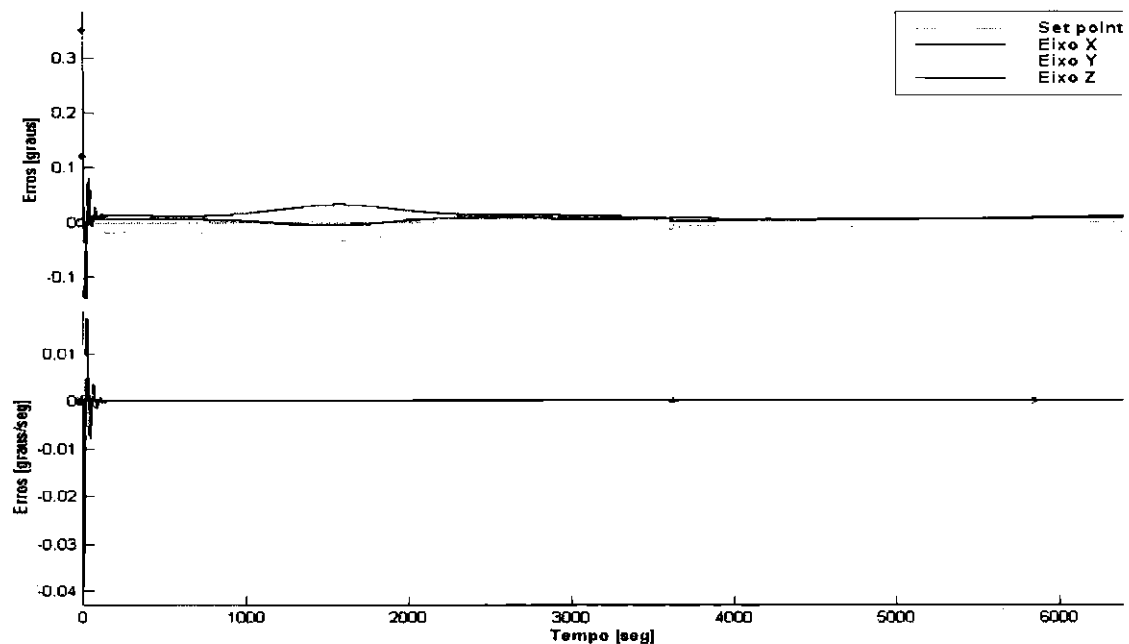


Fig. 6.21 - Erros angulares com controlador PD original e variação de 30% no momento de inércia de Y .

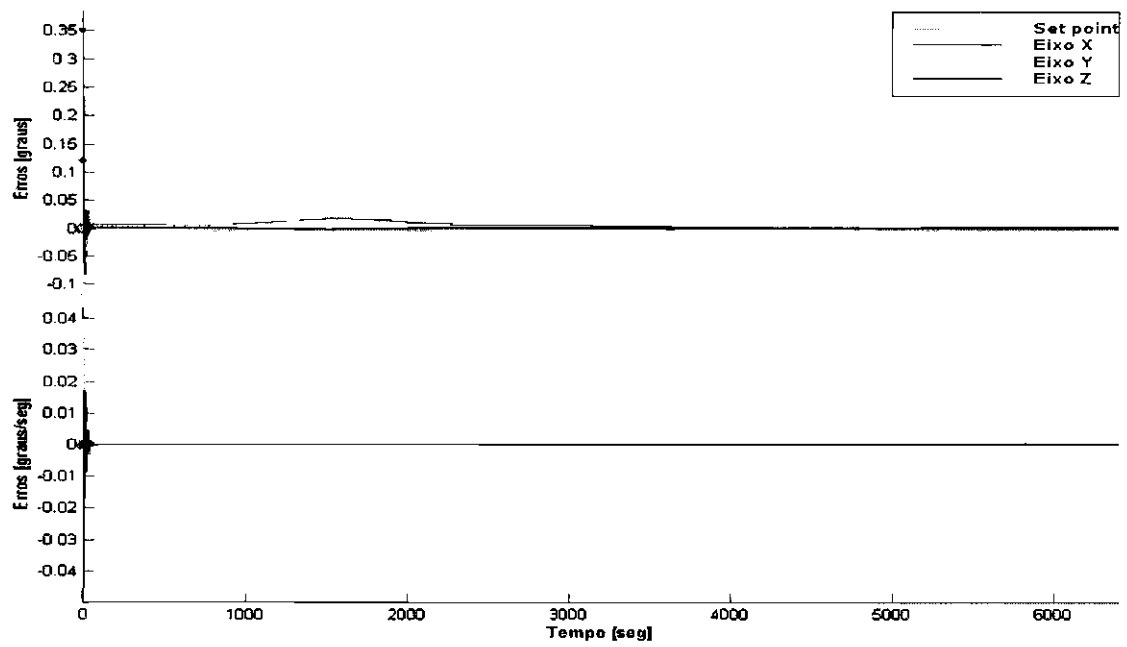


Fig. 6.22 - Erros obtidos com controlador PD otimizado e variação de 30% no momento de inércia de Y.

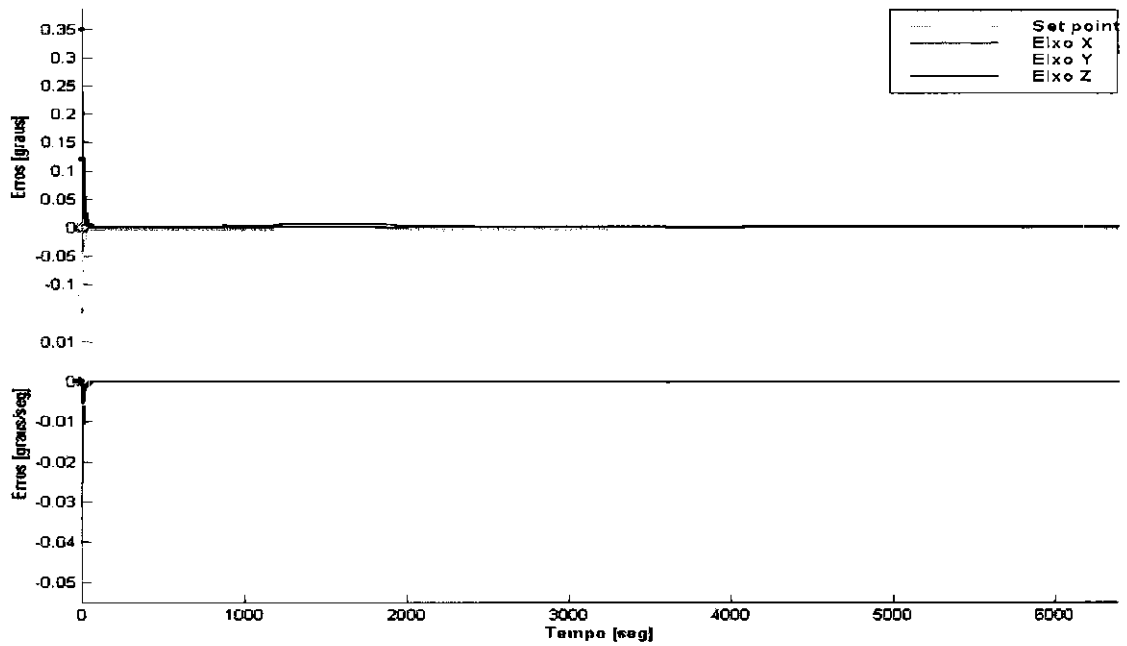


Fig. 6.23 - Erros do primeiro modelo de Mamdani obtidos com variação de 30% no momento de inércia de Y.

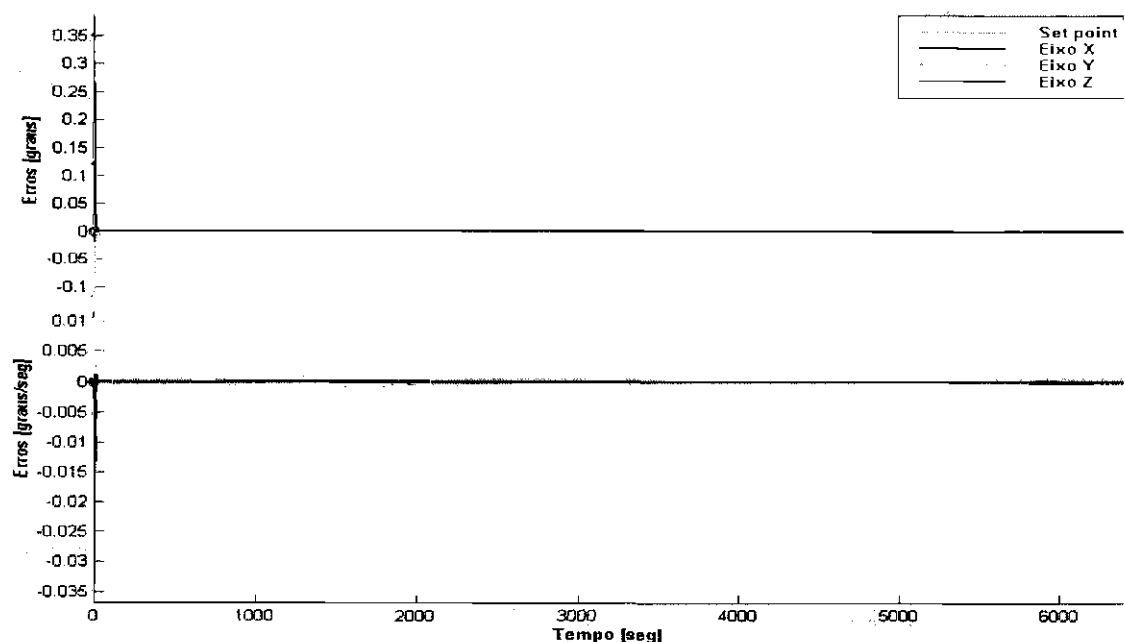


Fig. 6.24 - Erros do segundo modelo de Mamdani obtidos com variação de 30% no momento de inércia de Y.

6.5.2 Análise da Função de Avaliação

Como todos os controladores desenvolvidos utilizaram a mesma função de avaliação, torna-se possível a comparação dos resultados desta função em relação a cada um dos experimentos. Para se comparar o controlador PD original, foi criado um cromossoma com seus ganhos e o controlador obtido foi avaliado dentro do AG. O maior valor da função de avaliação obtido em cada experimento, além do valor da função de avaliação encontrado para o PD original encontram-se na Tabela 6.5.

Pode-se verificar que os resultados da Tabela 6.5 são coerentes com os resultados da Tabela 6.3, ou seja, os controladores considerados como melhores pelo algoritmo genético tiveram desempenho melhor também na prática, com a ressalva das discrepâncias apontadas anteriormente.

Estas discrepâncias se devem provavelmente ao fato de que a função de avaliação adotada utiliza médias aritméticas das integrais dos erros angulares em relação à

**TABELA 6.5 - VALORES DA FUNÇÃO DE
AVALIAÇÃO**

VALORES DE FITNESS DOS CONTROLADORES	
Controlador	<i>Fitness</i>
PD original	0.79649
PD com AG	0.98254
a) Mamdani	0.98352
b) Mamdani	0.987847

posição e à velocidade. É possível se esperar que o uso de uma \top -conorma no lugar de uma das médias aritméticas (ou de ambas) levasse a resultados melhores.

Em outras palavras, a função de avaliação adotada pode ser vista como:

$$1 - f \left(g_1 \left(h_1 \left(\frac{\zeta_i(t)}{\nu_i^\zeta} \right) \right), g_2 \left(h_2 \left(\frac{\omega_j(t)}{\nu_j^\omega} \right) \right) \right)$$

onde $i, j \in \{x, y, z\}$, h_1 e h_2 são integrais em função do tempo, e f , g_1 e g_2 são médias aritméticas em função de $\{x, y, z\}$. É possível então que substituindo f ou g_1 e g_2 por \top -conormas, que são funções mais restritivas que as médias, obtenhamos resultados melhores.

Além disso, seria interessante investigar a utilização de integrais no lugar de g_1 e g_2 e uma \top -conorma¹ no lugar de h_1 e h_2 . Isso significaria agregar primeiramente os erros relativos a cada momento de tempo, para só então agregá-los em relação à órbita completa.

Finalmente, é interessante observar que os resultados obtidos na Tabela 6.4 demonstram que a função de avaliação adotada nos experimentos foi capaz de generalização, pois os desempenhos relativos mantêm-se, apesar de condições de simulação sensivelmente alteradas.

¹Notar que a função de avaliação 5.2 utilizou $f = g_1 = g_2 = \max$ e não produziu bons resultados.

6.5.3 Avaliação da Variável de Saída (Torque) dos Controladores

Como forma de complementar as avaliações dos experimentos, apresenta-se em seguida uma análise dos torques de saída de cada controlador.

Na Figura 6.25 são mostradas as curvas do torque de saída dos quatro controladores, possibilitando um acompanhamento das suas formas e também da ordem de grandeza desta variável.

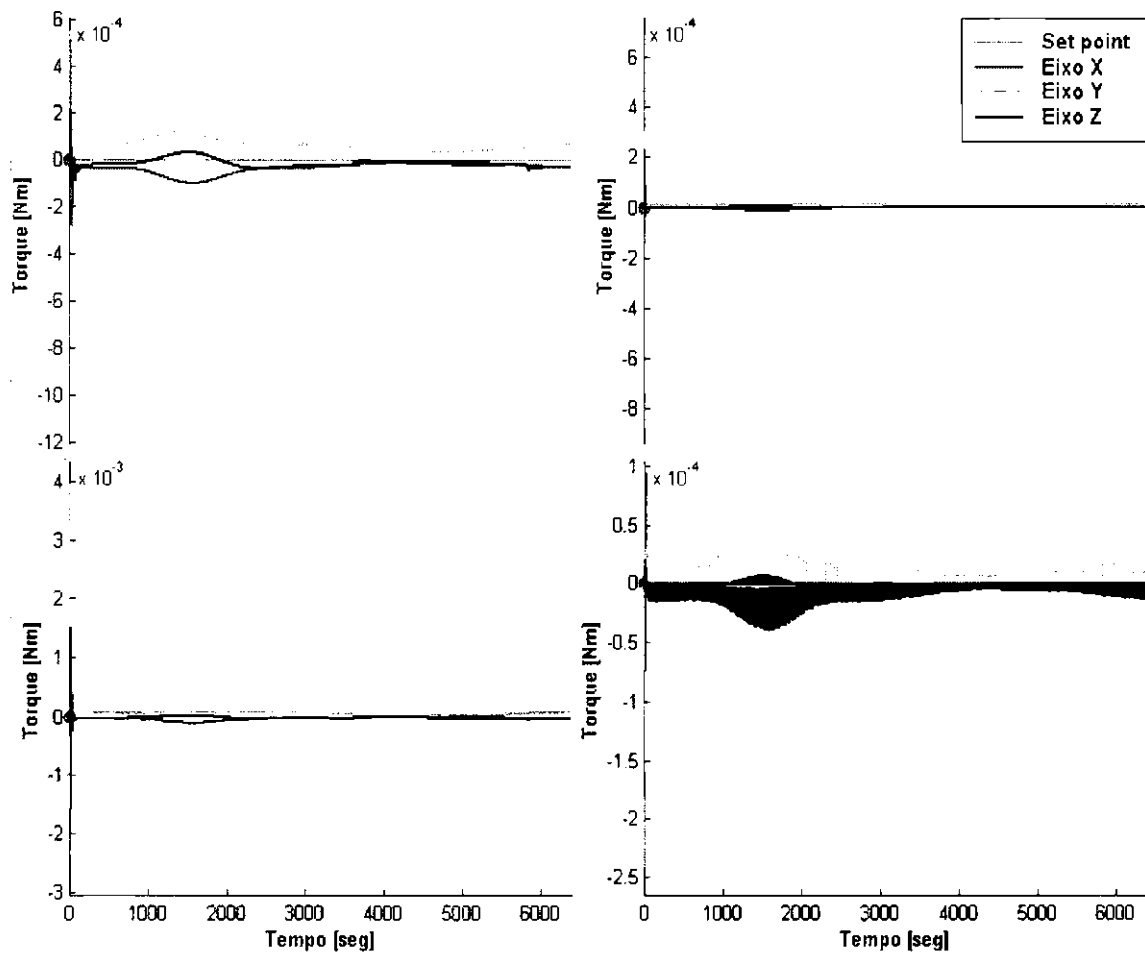


Fig. 6.25 - Torques de saída dos controladores estudados: a) Controlador PD original b) Controlador PD otimizado com AG c) Primeiro controlador nebuloso d) Segundo controlador nebuloso.

As Figuras 6.25a e 6.25b (canto esquerdo superior e inferior, respectivamente), apresentam os torques de saída para os dois controladores PDs. Pode-se observar uma maior suavização das curvas do controlador PD otimizado com o AG, que apresentou erros de posição e velocidade angulares menores que aqueles do controlador PD original. Porém, em consequência, a ordem de grandeza dos torques fornecidos pelo controlador PD otimizado aumentam aproximadamente dez vezes em relação ao PD original, justificando-se pela redução das oscilações na resposta de controle.

Os torques de saída dos controladores nebulosos (Figura 6.25c e 6.25d no canto direito superior e inferior, respectivamente), diferenciaram-se pouco pela ordem de grandeza dos torques, mas apresentaram discrepâncias significativas em relação às características oscilatórias. O segundo controlador nebuloso, projetado com cinco conjuntos nebulosos para mapear a variável torque, mostra-se pouco prático pois tem um comportamento muito oscilatório, o que pode comprometer o tempo de vida útil dos atuadores e comprometer suas funções.

Na Figura 6.25c (primeiro controlador nebuloso) são observadas características desejáveis na resposta de um controlador. A reação necessária ao momento inicial da simulação, onde os erros são maiores, é obtida de forma satisfatória, sem provocar grandes oscilações, mas mantendo os erros suficientemente inferiores aos outros controladores estudados. Essa característica de resposta mantém-se durante o período de simulação e conduz a resultados satisfatórios em relação às variáveis controladas e observadas.

Esses resultados podem contribuir para o desenvolvimento de uma função de avaliação mais completa. Uma hipótese para a elaboração desta função é a consideração do cálculo da variância do sinal de saída do controlador, para aumentar a sensibilidade ao nível de oscilações e introduzir informações da energia consumida com essa resposta de controle. Isso também possivelmente contribuiria para superar a possibilidade de adoção de um controlador que não satisfaça os requisitos de projeto.

CAPÍTULO 7

CONCLUSÕES E FUTUROS DESENVOLVIMENTOS

Neste trabalho utilizou-se um algoritmo genético nas tarefas de otimização de um controlador PD e de projeto de dois controladores nebulosos do tipo Mamdani para a fase de apontamento de um satélite artificial.

Uma interface gráfica foi implementada com a finalidade de facilitar a interação entre o usuário e a plataforma de desenvolvimento do projeto. A interface dispõe de vários módulos, cada um direcionado à introdução e configuração de um subsistema.

Esta interface constitui uma nova ferramenta que contribui para o desenvolvimento e análise de outros projetos. Suas características principais são a versatilidade para comportar alterações, a facilidade para introdução de novos módulos e a portabilidade.

Os controladores desenvolvidos com o auxílio de algoritmos genéticos foram testados, tanto em uma situação normal de funcionamento quanto em condições críticas, e nos dois casos tiveram desempenho superior a um controlador do tipo PD desenvolvido originalmente para o mesmo modelo de satélite, utilizando o método de alocação dos pólos [9].

Os algoritmos genéticos são considerados como uma técnica de busca simples e ao mesmo tempo eficiente, produzindo resultados de qualidade comparáveis aos métodos de busca exaustiva, nos quais todas as possibilidades são exploradas. Sua maior deficiência porém, está geralmente relacionada ao tempo de processamento necessário na obtenção da convergência para uma solução satisfatória.

No problema específico estudado nesta dissertação, o grande tempo dispendido corresponde à abordagem utilizada, onde foi necessária a simulação do satélite em regime de operação nominal para que se pudesse avaliar os erros produzidos sob as condições de órbita. Estas simulações só foram possíveis de ser realizadas, graças

às equações de dinâmica do satélite desenvolvidas em [9]. Por si só o modelo do satélite carece de certo tempo intrínseco à propagação da órbita, o qual está vinculado à taxa de amostragem dos sinais dos sensores e ao respectivo processamento. Esse comportamento se estende ao uso do algoritmo genético por ele herdar esta característica.

O controlador PD original proporcionou resultados significativamente melhores após a otimização com o algoritmo genético, que conseguiu encontrar valores adequados para os ganhos em 23 gerações. Esta pode ser caracterizada como uma tarefa de média dificuldade, já que trata-se da busca de seis parâmetros sensíveis a pequenas variações.

Contudo, os dois controladores nebulosos projetados destacam-se sobre os dois modelos PD avaliados. Inicialmente, porque um número menor de parâmetros tiveram que ser aprendidos na construção dos dois modelos nebulosos. Isso teve consequências em termos de um tempo menor de desenvolvimento, tendo-se obtido a convergência do algoritmo genético com poucas gerações. Além disso, os controladores nebulosos se destacaram devido à própria robustez que é característica desse tipo de controle, levando a resultados sensivelmente melhores que os obtidos pelos controladores PD.

Em ambos os tipos de avaliação realizados, de desempenho e de robustez, ficam evidentes as diferenças entre os controladores PD e os nebulosos. Estes últimos apresentam resultados de erros inferiores aos primeiros, garantindo precisão no apontamento juntamente com a robustez necessária a operações desta natureza. Destaca-se o primeiro controlador nebuloso, o qual utiliza 3 conjuntos para mapear a variável de saída, pois este conduz a erros menores, com comportamento mais suave e menos oscilatório que os outros controladores estudados.

Os resultados obtidos podem ainda ser certamente melhorados. Uma primeira possibilidade envolveria um estudo mais aprofundado na determinação de uma função de avaliação mais adaptada ao problema do que a atual.

O AG adotado neste estudo utiliza operadores genéticos que operam sobre cromossomas nos quais cada parâmetro é representado por um número fixo de bits, codificado segundo especificações do utilizador. Essa representação tem desvantagens quando aplicada em cromossomas multidimensionais e problemas numéricos de alta precisão, pois o tamanho necessário ao cromossoma para a representação dos vários parâmetros com suas respectivas precisões pode ser impraticável devido ao tempo de processamento associado. Eventualmente, pode também ocorrer situações de convergência prematura para um ponto de ótimo não global ou a inability para a sintonização local fina dos parâmetros. Isso porque, dependendo da representação adotada, dois pontos próximos no espaço de representação do AG não necessariamente estão próximos no espaço do problema ¹.

Uma forma alternativa de evoluir o processo de busca é realizar as operações em cromossomas cujos genes estão codificados diretamente em ponto flutuante. Desta maneira, as operações de conversão de base são efetuadas diretamente pelo processador. Além disso, garante-se que dois pontos próximos um do outro no espaço de representação do AG também estão próximos no espaço do problema (o inverso também é válido). Essa abordagem é recomendada para problemas de otimização de parâmetros com variação no domínio do contínuo e representa portanto, uma abordagem interessante para desenvolvimentos futuros no contexto deste trabalho.

Uma outra possibilidade de estudo seria a de incrementar o número de termos nebulosos das variáveis envolvidas no controle nebuloso, desta maneira refinando as respostas de controle e propiciando uma precisão maior no apontamento. Por outro lado, tal incremento confere um grau maior de dificuldade para o algoritmo genético, que deve aprender mais parâmetros, e torna o próprio controlador nebuloso mais lento, por agregar uma carga computacional maior.

Deve-se lembrar também que as leis de controle PD de cada eixo foram substituídas, neste trabalho, por um único controlador nebuloso, responsável pelos três eixos

¹Isto pode eventualmente ser atenuado usando-se a codificação de Gray [88].

simultaneamente. Uma abordagem alternativa que pode ser utilizada no futuro é o projeto de controladores nebulosos específicos para cada eixo. Assim, deve-se obter controladores especializados, com desempenho e robustez melhorados. Todavia, deve-se esperar em contrapartida uma demanda computacional mais elevada.

Além disso, seria interessante investigar o uso de outras arquiteturas de controladores nebulosos, notadamente o modelo de Sugeno, que, embora requeira um número maior de parâmetros a serem aprendidos do que o de Mamdani, guarda uma semelhança com os controladores convencionais. Também outras arquiteturas convencionais podem ser investigadas, como por exemplo um controlador PID, que em geral apresenta maiores dificuldades de projeto, mas que podem ser atenuadas com o auxílio do AG.

Para problemas de ordem prática, como o objeto de estudo deste trabalho, a eficiência está em parte vinculada ao tempo intrínseco do ambiente de desenvolvimento na execução da tarefa de se encontrar respostas a um problema. Uma das características mais atraentes do ambiente Matlab, adotado neste trabalho, é o de dispor de várias ferramentas para o desenvolvimento de trabalhos de pesquisa, contendo tanto funções específicas quanto interfaces. No entanto, o código gerado é interpretado, dificultando a implementação de algoritmos para modelos complexos que sejam capazes de fornecer respostas em tempo hábil.

Os experimentos deste trabalho foram executados em um PC Pentium, com “clock” de 200 MHz, 128 Mbytes de RAM e rodando sob NT Server como sistema operacional. Neste equipamento, o tempo necessário para obtenção da convergência do AG variou entre 10 horas e 3 dias, conforme o número de gerações necessárias à convergência do AG. Uma maneira de diminuir o tempo de processamento, seria a conversão do código gerado pelo Matlab para uma linguagem compilável, como por exemplo Fortran ou C. Também seria possível melhorar o desempenho da aplicação utilizando-se um “hardware” de processamento mais veloz.

Também, para acelerar a criação de controladores no ambiente desenvolvido, ou

mesmo como forma de focalizar e refinar a busca dos parâmetros, seria interessante acrescentar na interface gráfica a opção de introduzir uma população inicial como “semente”, ou seja, uma população composta por candidatos qualificados, já aptos. Esta tarefa é relativamente simples, uma vez que o AG utilizado está preparado para trabalhar sob esta condição.

A forma como são obtidas as informações pela interface também poderia ser melhorada, diminuindo a complexidade do uso da ferramenta. No seu estado atual, a ferramenta apresenta características que podem representar uma restrição ao usuário, por exigir dele conhecimento sobre as técnicas em uso e a forma de representação das variáveis. Uma melhoria neste sentido seria também a amplificação do número de conjuntos nebulosos utilizados no mapeamento das variáveis envolvidas em um dado problema.

Seria também interessante aumentar o número de recursos da ferramenta atual, como por exemplo abordar outras formas de proceder às otimizações, acrescentar outros módulos e interfaces para flexibilizar a substituição do modelo de satélite, de modo a possibilitar a utilização de outras estruturas como por exemplo painéis flexíveis, flexibilizar o incremento e a inserção de novas perturbações, etc. Melhorias neste sentido podem, então, auxiliar o desenvolvimento de modelos de controle para outras tarefas.

Uma perspectiva mais ambiciosa seria a de direcionar os trabalhos de pesquisa futuros no sentido de desenvolver uma ferramenta capaz de projetar e otimizar controladores automaticamente. A literatura atualmente apresenta bons relatos e alguns sucessos nesta linha, por exemplo aquele encontrado em [86] (tese de doutorado em desenvolvimento na Alemanha), onde se emprega uma metodologia semelhante àquela utilizada no sistema “Adaptative Neuro-Fuzzy Inference System” (ANFIS) proposta em [87] e [27].

Finalmente, é importante lembrar que o AG tem as características necessárias para operar em processamento paralelo, uma das mais recentes e promissoras linhas de

pesquisa da área. Isso possibilitaria se elevar a capacidade de processamento e acelerar a execução de tarefas computacionais. Em nosso caso, o tempo de criação de um controlador poderia então ser drasticamente reduzido, pois a simulação, e consequentemente a avaliação, de cada um dos controladores descritos pelos cromossomos de uma dada geração poderia ser realizada de maneira independente.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Conway, D.; Sperling, R.; Folta, D.; Richon, K.; De'azio, R. Automated maneuver planning using a fuzzy logic algorithm. In: GSFC Flight Mechanics/Estimation Theory Symposium, 1994. **Anais.** NASA, 1994. p. 49-61.
- [2] Petermann, B.; Misra, A. K. Attitude control of small satellites using fuzzy logic. **Spacecraft Dynamics - AAS**, pages 1039-1057, 1996.
- [3] Hemerly, E. M.; Rodriguez, C. C. Guiagem de veículos autônomos utilizando sensor de visão. In: Congresso Brasileiro de Automática, 10., Congresso Latino Americano de Controle Automático, 6., Rio de Janeiro, 1994. **Anais.** Set. 1994. p. 873-878.
- [4] Saffiotti, A. **Autonomous robot navigation: a fuzzy logic approach.** Bruxelles. 224 p. Tese (Thèse présentée en vue de l'obtention du grade académique de Docteur en Sciences Appliquées) - Université Libre de Bruxelles, Faculté de Sciences Appliquées, IRIDIA - Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle, 1998.
- [5] Guerra, R.; Sandri, S. A.; Souza, M. L. O. Dynamics and design of autonomous attitude control of a satellite using fuzzy logic. In: Congresso Brasileiro de Engenharia Mecânica, (COBEM'97 - COB 1338). Bauru, 1997. **Anais.** Dec. 1997.
- [6] Silva, S. A. **Estudo de manóbras de atitude para satélites rígido/flexíveis.** São José dos Campos, 1998. Proposta de Dissertação de Mestrado do Curso de Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle (Instituto Nacional de Pesquisas Espaciais - INPE), abr. 1998.

- [7] Kawakami, R. **Modelamento neural de ações de controle manual em tarefas de rastreamento.** São José dos Campos. Dissertação (Mestrado em Computação) - Instituto Tecnológico da Aeronáutica, 1997.
- [8] Fonseca, I. M.; Souza, P. N.; Neri, J. A. F.C.; Guedes, U. T. V. Attitude dynamics and control of the Brazilian Scientific Application Satellite - SACI-1. **ESA International Conference On Spacecraft Guidance Navigation And Control Systems**, 3. Netherlands, Noordwijk. Nov. 1996.
- [9] Silva, A. R. da **Estudo de sistema de controle de um satélite artificial durante a fase de transferência orbital e apontamento.** São José dos Campos. 86 p. (INPE-6397-TDI/613). Dissertação (Mestrado em Engenharia Espacial e Tecnologias Espaciais/Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais, 1997.
- [10] Silva, A. R. da; Souza, L. C. G. de Interação do sistema de controle e a estrutura flexível de um satélite durante transferência orbital. In: Congresso Brasileiro de Engenharia Mecânica, (COBEM'97 - COB 1332). Bauru, 1997. **Anais.** dez. 1997.
- [11] Phillips, C. L.; Harbon, R. D. **Feedback control systems.** London: Prentice Hall, 1991. 664 p.
- [12] Zapata, G. O. A. **Contribuições à análise de desempenho e projeto de controladores lógicos nebulosos.** São José dos Campos. 143 p. Dissertação (Mestrado em Ciência, Área de Sistemas e Controle do Curso de Engenharia Eletrônica e Computação) - Instituto Tecnológico de Aeronáutica, 1997.
- [13] Zapata, G. O.; Kienitz, K. H.; Yoneyama, T. Análise de desempenho de um controlador nebuloso com quantização de sinais. In: Simpósio Brasileiro de Automação Inteligente, 3., (SBAI'97). Vitória, 1997. **Anais.** Vitória, set. 1997. p. 301-306.

- [14] Driankov, D.; Hellendoorn, H.; Reinfrank, M. **An Introduction to fuzzy control**. Springer-Verlag TELOS. 1996. 316 p.
- [15] Lee, C. C. Fuzzy logic in control systems: Fuzzy logic controller (part I). **IEEE Transactions on Systems, Man and Cybernetics**, v. 20, n. 2, p. 404-418, Mar./Apr. 1990.
- [16] Mendel, J. M. Fuzzy logic systems for engineering: a tutorial. **Proceedings of the IEEE**, v. 83, n. 3, p. 345-377, Mar. 1995.
- [17] Castro, J. L. Fuzzy logic controllers are universal approximators. **IEEE Transactions on Systems, Man and Cybernetics**, v. 25, n. 4, p. 629-635, Apr. 1995.
- [18] Dutta, S. Fuzzy logic applications: Technological and strategic issues. **IEEE Transactions on Engineering Management**, v. 40, n. 3, p. 237-254, Aug. 1993.
- [19] Woodward, M. A. Fuzzy open-loop attitude control for the fast spacecraft. In: NASA AIAA, Guidance, Navigation and Control Conference, San Diego, July 1996. **Proceedings**. San Diego: NASA, 1996.
- [20] Conway, D.; Sperling, R.; Folta, D.; Richon, K.; DeFazio, R. Automated maneuver planning using a fuzzy logic controller. In: Guidance, Navigation and Control Conference, San Diego-CA, 1994. **Proceeding**. San Diego: NASA, July 1996.
- [21] Guerra, R.; Sandri, S. A.; Souza, M. L. O. Controle de atitude autônomo de satélites usando lógica nebulosa. In: Simpósio Brasileiro de Automação Inteligente, 3., (SBAI'97). Vitória, 1997. **Anais**. Set. 1997. p. 337-342.
- [22] Chiu, S.; Chand, S. Adaptive traffic signal control using fuzzy logic. In: **Fuzzy logic technology and applications**. New York: The Institute of Electrical and Electronics Engineers, 1994. p. 101-106.

- [23] Karr, C. L.; Gentry, E. J. Fuzzy control of pH using genetic algorithms.
IEEE Transactions on Fuzzy Systems, v. 1, n. 1, p. 46-53, Feb. 1993.
- [24] Weil, H. G.; Probst, G.; Graf, F. Fuzzy expert system for automatic
transmission control. In: Marks II, R. J., ed. **Fuzzy logic technology
and applications**. New York: IEEE, 1994. p. 88 - 93.
- [25] Klir, G. J.; Folger, T. A. **Fuzzy sets, uncertainty, and informations**.
Englewood Cliffs - NJ: Prentice Hall, 1988.
- [26] D'Amore, R.; Saotome, O.; Kienitz, K. H. Controlador nebuloso com
detecção de regras ativas. In: Simpósio Brasileiro de Automação
Inteligente, 3., (SBAI'97). Vitória, 1997. **Anais**. Vitória, 1997. p. 313-318.
- [27] Jang, J. S. R. ANFIS: Adaptive-network-based fuzzy inference system.
IEEE Transactions on Systems, Man and Cybernetics, v. 23, n. 3,
p. 665-685, 1993.
- [28] Lin, C. -T. A neural fuzzy control system with structure and parameter
learning. **Fuzzy Sets and Systems**, n. 70, p. 183-212, 1995.
- [29] Kosko, B. **Neural networks and fuzzy systems**. Englewood Cliffs, NJ:
Prentice Hall, 1992.
- [30] Wang, L.; Mendel, J. M. Generating fuzzy rules by learning from examples.
IEEE Transactions on Systems, Man and Cybernetics, v. 22, n. 6,
p. 1414-1427, Nov./Dec. 1992.
- [31] Correa, C; Sandri, S. A. e Souza, L. C. G.; The use of Generic Algorithms on
a Fuzzy Controller for a Satellite During the Pointing Phase. ABCM
International Symposium on Flight Dynamics, 14., Foz do Iguaçu, Feb.
1999. p. 8-12.
- [32] Pham, D. T.; Karaboga, D. Optimum design of fuzzy logic controllers using
genetic algorithms. **Jornal System Engineering**, v. 1, p. 114-118, 1991.

- [33] Pinheiro, F. A.; Moreira, A. N.; Cardoso, E. N.; Vasconcelos, J. A.
Algoritmo genético aplicado ao controle de tensão em sistemas elétricos de potência. In: Simpósio Brasileiro de Automação Inteligente, 3., (SBAI'97). Vitória, 1997. **Anais**, set. 1997. p. 289-294.
- [34] Kim, J.; Moon, Y.; Zeigler, B. P. Designing fuzzy net controllers using GA optimization. In: IEEE/IFAC Joint Symposium on Computer-Aided Control System Design. Tucson, Mar 7-9, 1994. **Proceedings**. Tucson: IEEE, 1994. p. 83-88.
- [35] Sanchez, E.; Shibata, T.; Zadeh, L. A. **Genetic algorithms and fuzzy logic systems: soft computing perspectives**. Singapore: World Scientific Publishing, 1997. 240 p.
- [36] Silva, J. C. A.; Kierner, C. Algoritmo genético aplicado a simulação. In: Simpósio Brasileiro de Automação Inteligente, 3., (SBAI'97). Vitória, 1997. **Anais**, set. 1997. p. 283- 288.
- [37] Stachowitz, M.; Yao, D.; Chen, T. **Tuning a PID controller based on a genetic algorithm**. Minnesota: Un. of Minnesota, 1995. (Dept. Elect. and Comp. Eng. - Technical Report Lab. for Int. Syst.).
- [38] Tang, K. S.; Man, K. F.; Kwong, S.; Ho, Q. Genetic algorithms and their applications. **IEEE Signal Processing Magazine**, v. 13, n. 6, p. 22-37, Nov. 1996.
- [39] Vasconcelos, J. A. D.; Neiba, A. M.; Cardoso, E. N.; Pinheiro, F. A.
Algoritmo genético aplicado ao controle de tensão em sistemas elétricos de potência. In: Simpósio Brasileiro de Automação Inteligente, 3., (SBAI'97). Vitória, 1997. **Anais**. 1997. p. 38-44.
- [40] Husbands, P. Genetic algorithms in optimization and adaptation. In: Kronsjo, L.; Shunsheruddin, D., ed. **Advances in parallel algorithms**. London: Blackwell Scientific Publications, 1992. p. 227-277.

- [41] Grefenstette, J. J. Optimization of control parameters for genetic algorithms. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 16, n. 1, p. 122-128, Jan./Feb 1986.
- [42] Yoneyama, T.; Cardozo, E. Randomized search methods for optimization of controller parameters. In: Congresso Brasileiro de Automática, 10., Congresso Latino Americano de Controle Automático, 6., Rio de Janeiro, set. 1994. p. 584-589.
- [43] Karr, C. Applying genetics to fuzzy logic. **AI Expert**, v. 6, n. 3, p. 38-43, Mar. 1991.
- [44] Karr, C. Adaptive control with fuzzy logic and genetic algorithms. In: Yager, R. R.; Zadeh, L. A., ed. **Fuzzy sets, neural networks and soft computing**. New York: Van Nostrand Reinhold, 1994. p. 345 - 367.
- [45] Kinzel, J.; Klawonn, F.; Kruse, R. Modifications of genetic algorithms for designing and optimizing fuzzy controllers. [online]. <ftp://ftp.et-inf.fho-emden.de/pub/FHO/Informatik/klawonn/eci3e94.ps.gz>. Oct. 1999.
- [46] Mohamadian, M.; Stonier, R. J. Generating fuzzy rules by genetic algorithms. In: IEEE International Workshop on Robot and Human Communication, 3., Nagoya, 1994. **Proceedings**. NJ: IEEE, 1994. p. 362-367.
- [47] Mohamadian, M.; Stonier, R. J. Tuning and optimization of membership of fuzzy logic controllers by genetic algorithms. In: IEEE International Workshop on Robot and Human Communication. 3., Nagoya, 1994. **Proceedings**. NJ: IEEE, 1994. p. 356-361.
- [48] Linkens, D. A.; Nyonegesa, H. O. Genetic algorithms for fuzzy control: offline system development an application. **IEE Proc.: Control Theory Appl.** v. 142, n. 3, p. 161-176, May 1995.

- [49] Homaifar, A.; McCormick, E. Simultaneous design of membership functions and rule sets for fuzzy controllers using genetics algoritms. **IEEE Transactions on Fuzzy Systems**, v. 3, n. 2, p. 46-53, May 1993.
- [50] Kienitz, K. H. A fuzzy control method based on possibility distributions. In: Simpósio Brasileiro de Automação Inteligente, 2., (SBAI'95). Curitiba, 1995. **Anais.** Set. 1995. p. 79-84.
- [51] Aya, J. C. C.; Costa, O. L. V. Otimização de controladores nebulosos usando algoritmos genéticos. In: Simpósio Brasileiro de Automação Inteligente, 3., (SBAI'97). Vitória, 1997. **Anais.** set. 1997. p 207-212.
- [52] Mamdani, E. H.; Assilian, S. Application on fuzzy algorithm for control of simple dynamic plant. **Proceedings of Institute Electrical of Engeneering**, v. 121, n. 12, p. 1585-1888, 1974. (IEEF).
- [53] Larsen, P. M. Industrial aplicacions of fuzzy logic control. In: Mamdani, E. H.; Gaines, B. R., ed. **Fuzzy reasoning and its aplicacions**. London: Academic, 1981. p. 335 - 342.
- [54] Mamdani, E. H. Advances in the linguistic syntesis of fuzzy controllers. **International Jornal Man-Machines Studies**, v. 8, p. 669-678, 1976.
- [55] Mamdani, E. H.; Assilian, S. An experiment in linguistic synthesis with a fuzzy logic controller. In: Mamdani, E. H.; Gaines, B. R., ed. **Fuzzy reasoning and its aplicacions**. London: Academic, 1981. p. 311-323.
- [56] Mamdani, E. H.; Procyk, T.; Baaklini, N. Application of fuzzy logic to controller design based on linguistic protocol. In: Mamdani, E. H.; Gaines, B. R., ed. **Discrete systems and fuzzy reasoning**. London: University of London. p. 125 - 149.
- [57] Bishop, R. H. **Modern control systems analysis and design using MATLAB[©]**. Reading, Massachusetts: Addison-Wesley, 1993. 161 p.

- [58] Shahian, B.; Hassul, M. **Control system design using Matlab[©]**. Engewood Cliffs - NJ: Prentice Hall, 1993.
- [59] The MathWorks Inc. **The student edition of MATLABTM**: Student user guide. Engewood Cliffs: Prentice Hall, 1992. 494 p.
- [60] Kuo, B. C.; Hanselman, D. C. **MATLAB[©] tools for control system analysis and design**. Engewood Cliffs: Prentice Hall, 1994. 227 p.
- [61] Leonard, N. E.; Levine, W. S. **Using MATLAB[©] to analyse and design control systems**. 2 ed. Menlo Park-Ca: Addison-Wesley Publishing Company, 1995. 212 p.
- [62] Ogata, K. **Designing linear control systems with MATLAB[©]**. Engewood Cliffs: Prentice Hall, 1994. 226 p.
- [63] Roger J., J. -S.; Gulley, N. **Fuzzy logic toolbox**: For use with MATLAB[©]. Natick: The MathWorks, 1995.
- [64] Potvin, A. **Genetic algorithm m-files**. [Programa de computador - [online]. <<ftp://ftp.mathworks.com/pub/contrib/v4/optim/genetic.zip>>. Mar. 1998. Modelo original desenvolvido em outubro de 1993. Copyright (c) 1993 by the MathWorks, Inc. (Última modificação em 24 de janeiro de 1994).
- [65] Silva, J. C. A.; Kirner, C. Genetic algorithms applied to the simulation of complex systems. In: Proceedings of the International Conference on Intelligent Systems, 5, (IS'96). Reno, june, 1996. **Proceedings**. Reno: ISCA, 1996. p. 135-139.
- [66] Palazzo, L. A. M. Algoritmos para computação evolutiva. **Caderno de Informática and Educação**, v. 1, n. 1, p. 53-71, dez. 1996.
- [67] Holland, J. H.; **Adaptation in natural and artificial systems**. Ann Arbor: The Univ. of Michigan Press, 1975.

- [68] Filho, E. F. M. **Algoritmos Genéticos**. [online]
<<http://www.icmsec.sc.usp.br/~prico/genel.html>>. May 1999.
- [69] Goldberg, D. E. **Genetic algorithms in search, optimization and machine learning**. Reading: Addison-Wesley, 1989. 412 p.
- [70] Hoffmann, F. **Genetic algorithm**. [online]
<<http://www.cs.berkeley.edu:80/~fhoffman/ga.html>>. May 1997.
- [71] Davis, L. **The handbook of genetic algorithms**. New York:Van Nostrand Reinhold, 1991.
- [72] Zadeh, L. A. Fuzzy sets. **Information and Control**, v. 8, n.3, p. 338-353, June 1965.
- [73] Sandri, S. A. **Introdução a lógica "fuzzy"**. [online].
<<http://jupiter.lac.inpe.br/~sandri/files/vitoria/ps>>. 1997.
- [74] Bauer, P.; Nouak, S.; Winkler, R. **A brief course in fuzzy logic and fuzzy control**. [online].
<<ftp://ftp.fll.uni-linz.ac.at/pub/info/fllorg.ps.gz>>. Apr. 1998.
- [75] Guerra, R. **Projeto e simulação do controle de atitude autônomo de satélites usando lógica nebulosa**. São José dos Campos. 190 p. (INPE-6714-TDI/630). Dissertação (Mestrado em Eng. Espacial e Tec. Espaciais/Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais, mar. 1998.
- [76] Dubois, D.; Hellendoorn, H.; Reinfrank, M. **Fuzzy sets and systems theory and applications**. New York: Academic Press, 1980.
- [77] Yager, R. R.; Filev, D. P. **Essentials of fuzzy modeling and control**. New York: John Willey & Sons Inc, 1994. 388 p.
- [78] Gomide, F. A. C.; Gudwin, R. R.; Tanscheit, R. Conceitos fundamentais da teoria de conjuntos fuzzy, lógica fuzzy e aplicações. In: International

- Fuzzy Systems Association World Congress, 6., (IFSA'95). Campinas, 1995. **Anais**. São Paulo: IFSA, 1995. p. 01-38.
- [79] Fardin, J. F.; Arruda, L. V. R.; Amaral, W. C. Identificação relevante para controle preditivo utilizando Algoritmos Genéticos. In: Simpósio Brasileiro de Automação Inteligente, 3., (SBAI'97). Vitória, 1997. **Anais**. Set. 1997. p. 295-300.
- [80] Nazzetta, R. M.; Oliveira, G. H. C.; Campello, R. J. G. B.; Amaral, W. C. Controle preditivo de processos não lineares utilizando modelos dinâmicos nebulosos. In: Congresso Brasileiro de Automática, 12., Uberlândia, 1998. **Anais**. p. 169-174.
- [81] Rani, K. Y.; Unbehauen, H. Study of predictive controller tuning methods. **Automática**, v. 33, n. 12, p. 2243-2248, 1997.
- [82] Lee, C. C. Fuzzy logic in control systems: Fuzzy logic controller (part II). **IEEE Transactions on Systems, Man and Cybernetics**, v. 20, n. 2, p. 419-430, Mar./Apr. 1990.
- [83] International Fuzzy Systems Association World Congress, 6., Campinas, 1995. **Tutorials**. Campinas: State University of Campinas - UNICAMP, July 1995. p. 180. (IFSA'95).
- [84] Duarte, L. S. **SOHO**: Sign conventions for rotations. [online]. <<http://sohowww.nascom.nasa.gov/gif/sohorotations.gif>>. Apr. 1998.
- [85] Ogata, K. **Engenharia de controle moderno**. 2. ed. Rio de Janeiro: Editora Prentice Hall do Brasil, 1993. 781 p.
- [86] Heider H., Tryba V., Mühlenfeld E. Automatic design of fuzzy systems by genetic algorithms. In: Bouchon-Meunier B., Yager R. R., Zadeh L. A., ed **Fuzzy logic and soft computing**. Singapore: World Scientific, 1995. p. 21-28.

- [87] Jang, J. S. R. Fuzzy modeling using generalized neural networks and Kalman filter algorithm. In: National Conference on Artificial Intelligence, 9., (AAAI'91). Anaheim, July 1991. **Proceedings**. California: AAAI Press, 1991. p. 762-767.
- [88] Michalewicz, Z. **Genetic algorithms + data structures = evolution programs**. 3. ed. New York: Springer-Verlag, 1996. 387 p.