



**UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS DE
COMUNICAÇÃO E AUTOMAÇÃO**

MISSILENE DA SILVA FARIAS

**PROJETO DE CONTROLADOR ADAPTATIVO VIA OTIMIZAÇÃO POR
ENXAME DE PARTÍCULAS APLICADO A SISTEMAS
MULTIVARIÁVEIS DE QUARTA ORDEM**

MOSSORÓ – RN

2015

MISSILENE DA SILVA FARIAS

**PROJETO DE CONTROLADOR ADAPTATIVO VIA OTIMIZAÇÃO POR
ENXAME DE PARTÍCULAS APLICADO A SISTEMAS
MULTIVARIÁVEIS DE QUARTA ORDEM**

Dissertação de mestrado acadêmico apresentada ao Programa de Pós-Graduação em Sistemas de Comunicação e Automação, como requisito para a obtenção do título de Mestre em Sistemas de Comunicação e Automação.

Orientador: Prof. Dr. Luiz Gonzaga de Queiroz Silveira Jr. - UFRN.

Co-orientador: Prof. Dr. Luciano Sales Barros - UFRN.

MOSSORÓ – RN

2015

FICHA CATALOGRÁFICA

Catálogo na Fonte

Catálogo de Publicação na Fonte. UFERSA - BIBLIOTECA CENTRAL ORLANDO TEIXEIRA - CAMPUS MOSSORÓ

Farias, Missilene da Silva.

Projeto de controlador adaptativo via otimização por enxame de partículas aplicado a sistemas multivariáveis de quarta ordem / Missilene da Silva Farias. - Mossoró, 2015.

112f: il.

1. Controle computadorizado de processo. 2. Automação. 3. Sistemas de multivariáveis. I. Título

RN/UFERSA/BCOT/437
F224p

CDD 629.895

MISSILENE DA SILVA FARIAS

**PROJETO DE CONTROLADOR ADAPTATIVO VIA OTIMIZAÇÃO POR
ENXAME DE PARTÍCULAS APLICADO A SISTEMAS
MULTIVARIÁVEIS DE QUARTA ORDEM**

Dissertação de mestrado acadêmico apresentada ao Programa de Pós-Graduação em Sistemas de Comunicação e Automação, como requisito para a obtenção do título de Mestre em Sistemas de Comunicação e Automação.

APROVADA EM: ___ / ___ / _____.

BANCA EXAMINADORA

Prof. Dr. Luiz Gonzaga de Queiroz Silveira Jr. – UFRN.
Presidente

Prof. Dr. Luciano Sales Barros. – UFRN.
Orientador

Prof. Dr. Vicente Angelo de Sousa Jr. – UFRN.
Primeiro Membro

Prof. Dr. George Acioli Jr. – UFCG.
Segundo Membro

Mossoró, 27 de fevereiro de 2015

AGRADECIMENTOS

Primeiramente ao meu Deus, Senhor de todas as coisas, pelo dom da vida e por seu amor infinito.

A Ronu e à minha família, pelo apoio, exemplo e por compreenderem minha ausência.

A todos os mestres ao longo de minha vida acadêmica, sobretudo a José Luiz da Silva, André Laurindo Maitelli, Clovis Bosco, Allan Aminadab, João Maria Camara, David Simonetti, Alberto Nicolau, José Tavares, José Patrocínio e Fabiana Karla Varella, os quais me servem de inspiração graças a qualidades inestimáveis como: ética, respeito, competência e profissionalismo.

Aos orientadores, Dr. Luiz Gonzaga e Dr. Luciano Sales, pela dedicação, paciência, apoio, incentivo e confiança. Jamais deixaram de acreditar em minha capacidade, mesmo quando eu, por vezes, não acreditei.

A todos os meus colegas de estudo: Ádler, Adelson, Tarcísio, Emmanuel, Jorge, Carlos, Sameque, Phelipe, Rodrigo. Muito obrigada!

À querida Lívia, secretária do PPGSCA, por toda ajuda e carinho.

Ao coordenador do PPGSCA, Idalmir Queiroz, por todo incentivo e auxílio.

A todos que de alguma maneira contribuíram direta ou indiretamente para a realização desse trabalho.

EPÍGRAFE

"Que a cada dia você tenha paciência para as dificuldades, sensatez para as escolhas, delicadeza para as palavras, coragem para as provas... Ame muitas coisas, porque em amar está a verdadeira força. Quem ama muito conquistará muito, e o que for feito com amor estará bem feito". (Vincent Van Gogh)

RESUMO

É proposta neste trabalho a obtenção de um controlador adaptativo por realimentação de estados para sistemas multivariáveis de quarta ordem variantes no tempo e com incertezas paramétricas. Em alternativa às dificuldades e limitações envolvendo os métodos usuais de síntese de controladores ótimos para esta classe de sistemas, por meio da investigação de um algoritmo baseado na otimização por enxame de partículas com boa sintonia dos parâmetros, busca-se a definição de uma estrutura flexível destinada a aplicações que demandem a tomada de decisão e estimação em tempo real dos parâmetros da matriz de realimentação. O controlador desenvolvido deverá fornecer, a cada execução, não somente estabilidade ao sistema, mas ao mesmo tempo, propiciar observância aos índices de desempenho fixados, sem a necessidade da realização de cálculos matemáticos complexos e dispendiosos, eliminando, dessa forma, as componentes subjetivas de análise e projeto.

Palavras-Chave: sistemas multivariáveis, otimização por enxame de partículas, controle adaptativo, realimentação de estados, estimação em tempo real.

ABSTRACT

The proposal for this study is to obtain an adaptive controller by feedback states for fourth order time-varying multivariable systems or with parametric uncertainty. Alternatively the difficulties and limitations involving the usual methods of synthesis of optimal controllers for this class of systems, through the investigation of an algorithm based on particles swarm optimization with good tuning of parameters, we seek to define a flexible structure for applications that require decision making and online estimation of the feedback matrix parameters. The controller developed should provide, for every execution, not only system stability, but at the same time, provides compliance to the performance indexes set, without the need to perform complex and expensive mathematical calculations, eliminating thus the subjective components of analysis and design.

Keywords: multivariable systems, particles swarm optimization, adaptive control, feedback states, online estimation.

LISTA DE TABELAS

Tabela 1.1 – Parâmetros utilizado pelo GA em OLIVEIRA et al (2012)	30
Tabela 1.2 – Parâmetros utilizado pelo PSO em OLIVEIRA et al. (2013)	31
Tabela 3.1 – Parâmetros de configuração do PSO desenvolvido	63
Tabela 4.1 – Tabela de especificação de testes	70
Tabela 4.2 – Tabela de média de ocorrência de variações do sistema ao longo de 10s e média de acionamentos do PSO para uma probabilidade <i>Prob_Var</i> de 1%	73
Tabela 4.3 – Tabela de média de ocorrência de variações do sistema ao longo de 10s e média de acionamentos do PSO para uma probabilidade <i>Prob_Var</i> de 5%	73
Tabela 4.4 – Tabela de média de ocorrência de variações do sistema ao longo de 10s e média de acionamentos do PSO para uma probabilidade <i>Prob_Var</i> de 10%	74
Tabela 4.5 – Tabela de média de ocorrência de variações do sistema ao longo de 10s e média de acionamentos do PSO para uma probabilidade <i>Prob_Var</i> de 20%	74
Tabela 4.6 – Tabela de média de ocorrência de variações do sistema ao longo de 10s e média de acionamentos do PSO para uma probabilidade <i>Prob_Var</i> de 40%.	74
Tabela 4.7 – Tabela de medidas de variância e tempos médios demandados pelo PSO para encontrar uma solução para uma probabilidade de variação <i>Prob_Var</i> de 1%	77
Tabela 4.8 – Tabela de medidas de variância e tempos médios demandados pelo PSO para encontrar uma solução para uma probabilidade de variação <i>Prob_Var</i> de 5%	78
Tabela 4.9 – Tabela de medidas de variância e tempos médios demandados pelo PSO para encontrar uma solução para uma probabilidade de variação <i>Prob_Var</i> de 10%	78
Tabela 4.10 – Tabela de medidas de variância e tempos médios demandados pelo PSO para encontrar uma solução para uma probabilidade de variação <i>Prob_Var</i> de 20%.	78

Tabela 4.11 – Tabela de medidas de variância e tempos médios demandados pelo PSO para encontrar uma solução para uma probabilidade de variação *Prob_Var* de 40%. 79

LISTA DE FIGURAS

Figura 1.1 - Componentes de um sistema de controle com realimentação de saída.....	19
Figura 1.2 – Diagrama de blocos vetorial de controle com realimentação de estados	19
Figura 2.1 – Mudança de direção de uma partícula	39
Figura 2.2 – Fluxograma para implementação básica de um algoritmo PSO	40
Figura 2.3 – Diagrama de blocos de um sistema adaptativo	42
Figura 2.4 – Diagrama de blocos de um sistema de controle por escalonamento de ganhos ..	43
Figura 2.5 – Diagrama de blocos de um controlador adaptativo por modelo de referência ...	44
Figura 2.6 – Diagrama de blocos de um Regulador auto-ajustável	46
Figura 2.7 – Diagrama de blocos de um controlador dual.	47
Figura 3.1 – Diagrama de blocos para o sistema de controle adaptativo proposto	48
Figura 3.2 – Etapas do controle adaptativo proposto	49
Figura 3.3 – Monitoramento das saídas e cálculo de erro	50
Figura 3.4 – Pseudocódigo do algoritmo para análise do erro máximo admitido entre as saídas da planta e das saídas do modelo	50
Figura 3.5 – Diagrama de blocos de um sistema em espaço de estados em malha aberta	52
Figura 3.6 – Diagrama de blocos de um sistema em espaço de estados em malha fechada ...	53
Figura 3.7 – Pseudocódigo do algoritmo utilizado para estimação da nova matriz de estados do sistema	56
Figura 3.8 – Caso II.a: diagrama de pólos de um sistema de quarta ordem com um par complexo conjugado próximo a origem.	59

Figura 3.9 – Caso III.a: diagrama de pólos de um sistema de quarta ordem com um par complexo conjugado próximo a origem.	60
Figura 3.10 – Caso III.b: diagrama de pólos de um sistema de quarta ordem com um par complexo conjugado situado entre os pólos reais.	61
Figura 3.11 – Caso III.c: diagrama de pólos de um sistema de quarta ordem com um par complexo conjugado próximo a origem.	61
Figura 3.12 – Pseudocódigo do algoritmo PSO utilizado.	65
Figura 3.13 – Posicionamento dos ganhos de realimentação da matriz de controle adaptados ao vetor posição em uma partícula.	66
Figura 4.1.a – Saídas do sistema real e saídas do modelo estimado.	71
Figura 4.1.b – Gráficos de desempenho do sistema de controle adaptativo.	72

NOMENCLATURA

Letras Latinas

K – Matriz de ganhos de realimentação;

A – Matriz de estados;

B – Matriz de entradas;

C – Matriz de saídas;

D – Matriz de transferência direta;

Q – Matriz de ponderação de estados;

R – Matriz de ponderação de controle;

Δt – Corresponde ao passo de iteração definido;

$U(k+1)$ – Sinal aplicado às entradas do sistema;

$f(x)$ – Função que descreve o comportamento de uma reta;

$x(t)$ – Variável de estados;

$u(t)$ – Sinal de entrada da planta;

$\vec{v}(t)$ – Vetor velocidade de uma partícula;

$R(t)$ – Referência;

t_s – Tempo de estabilização;

T – Período;

Y_{\max} – Máximo valor de saída;

Y_{Regime} – Valor de saída em regime permanente;

M_p – Máxima ultrapassagem;

c_1 – fator de atração individual;

c_2 – fator de atração social.

Letras Gregas

$\varphi(\Delta t)$ – Matriz de transição de estados do sistema em tempo discreto, e equivale a $e^{A\Delta t}$;

ξ – Fator de amortecimento;

ω_n – Frequência natural de oscilação amortecida;

ω_d – Frequência natural de oscilação não amortecida;

χ – Fator de constrição.

Abreviações

MIMO – *Multiple Input, Multiple Output*;

PID – Proporcional-Integral-Derivativo;

LQG – *Linear Quadratic Gaussian Regulator*;

LQR – *Linear Quadratic Regulator*;

LTI – *Linear Time-Invariant systems*;

SISO – *Single Input, Single Output*;

SMC – *Sliding Mode Control*;

NP – *Non-Deterministic Polynomial time*;

GA – *Genetic Algorithm*

ACO – *Ant Colony Optimization*;

PSO – *Particle Swarm Optimization*;

MMQ – Método Dos Mínimos Quadrados;

DPS – *Digital Signal Processor*.

Siglas

UFERSA – Universidade Federal Rural do Semi-Árido

PPGSCA – Programa de Pós-Graduação em Sistemas de Comunicação e Automação

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

NOMENCLATURA

1 INTRODUÇÃO	17
1.1 CONTROLE DE SISTEMAS MULTIVARIÁVEIS	18
1.2 OTIMIZAÇÃO COMPUTACIONAL	24
1.3 REVISÃO BIBLIOGRÁFICA	27
1.4 MOTIVAÇÃO	33
1.5 CONTRIBUIÇÕES	34
1.6 METODOLOGIA	34
1.7 ORGANIZAÇÃO DA DISSERTAÇÃO	35
2 FUNDAMENTAÇÃO TEÓRICA	36
2.1 ABORDAGENS EVOLUTIVAS	36
2.2 OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS	37
2.3 CONTROLE ADAPTATIVO	42
2.4 CONCLUSÕES DO CAPÍTULO	47
3 PROPOSTA DE CONTROLE	48
3.1 ESTRUTURA DO SISTEMA PROPOSTO	48
3.2 MONITORAMENTO DAS SAÍDAS DE INTERESSE	50
3.3 ESTIMAÇÃO DOS PARÂMETROS DO NOVO MODELO DA PLANTA	51
3.4 AVALIAÇÃO DO DESEMPENHO DO MODELO ESTIMADO	57
3.5 RECÁLCULO DA NOVA MATRIZ DE CONTROLE	62
3.6 CONCLUSÕES DO CAPÍTULO	67
4 RESULTADOS E DISCUSSÕES	68
4.1 RESULTADOS	70
4.2 ANÁLISE DOS RESULTADOS	72
4.3 CONCLUSÕES DO CAPÍTULO	79

5 CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS 80

REFERÊNCIAS BIBLIOGRÁFICAS 82

APÊNDICE

1 INTRODUÇÃO

A questão essencial envolvendo tanto a teoria como a prática da área de controle e automação diz respeito à obtenção de sistemas estáveis e robustos por meio de técnicas de realimentação (MARCHIORI, 1994, p. 3), e para isto existe a necessidade de se descrever matematicamente o processo a ser controlado.

Entre 1930 e 1940, surgiu a nomenclatura do *controle clássico*, na qual os sistemas são descritos através de modelos, cuja estrutura admite uma única entrada tendo influência significativa sobre o comportamento da variável a ser manipulada ao longo do tempo. Tais sistemas receberam a denominação de SISO (do inglês, *Single Input, Single Output*) (TAKAHASHI, 1998, p. 10). Para este tipo de representação é necessária uma descrição matemática que corresponda aos aspectos imprescindíveis na definição de um modelo matemático dinâmico geral, que represente adequadamente a evolução do fenômeno físico a ser controlado. Tais modelos devem representá-los a partir de dinâmicas simplificadas, porém perfeitamente eficazes no que tange à resolução de problemas (AGUIRRE, 2007, p. 52). E uma definição das características de saída em relação aos estímulos na entrada do processo, conhecida como *função de transferência*, torna-se necessária e satisfatória para se realizar a análise e, conseqüentemente, o projeto de um sistema de controle adequado à resolução de determinado problema.

Sendo assim, para os sistemas descritos por meio de funções de transferência, não é fornecida nenhuma informação referente à estrutura interna do sistema, e as características de sinais exógenos ao processo não apresentam influência sobre a resposta da planta, o que torna o conceito de função de transferência aplicável necessariamente aos sistemas lineares e invariantes no tempo (LTI – *Linear Time-Invariant systems*) (MAYA, 2011, p. 262).

A análise de sistemas SISO pode ser realizada a partir dos trabalhos desenvolvidos principalmente nos anos que se seguiram após 1930, perdurando até 1960. As idéias de cientistas como Nyquist, Bode, Evans, Wiener, Routh, Hurwitz, dentre outros, levaram ao desenvolvimento da teoria de controle clássico. E, segundo TAKAHASHI (1998, p. 10-11), durante muito tempo serviu de alicerce à evolução da indústria de base, fornecendo bons resultados às aplicações práticas para os sistemas de controle desenvolvidos até então, sobretudo no que diz respeito à síntese de controladores do tipo Proporcional, Integral e Derivativo (PID) (MARTINS, 2009, p. 5-6).

No entanto, com o passar do tempo, a exigência da melhoria na qualidade dos mais diferentes processos produtivos e os avanços verificados na ciência corroboraram para o surgimento em grande escala de sistemas mais modernos baseados em tecnologias predominantemente mais sofisticadas.

Estas novas tecnologias, contrariamente ao caso dos sistemas SISO, não raramente apresentavam uma ou mais saídas simultaneamente influenciadas por diferentes sinais de entrada. Estes sistemas receberam a denominação de sistemas MIMO (do inglês *Multiple Input, Multiple Output*) (OGATA, 2000, p. 576; AGUIRRE, 2007, p. 59).

E dessa forma, a teoria clássica de controle, até então considerada suficiente para os sistemas da época, tornou-se insatisfatória, pois a utilização de uma descrição externa da dinâmica do processo, poderia ocasionalmente tornar-se insuficiente para a definição de um modelo matemático satisfatório (OGATA, 2000, p. 576). Uma representação física deveria ser realizada de maneira adequada à evolução desses sistemas, fazendo, assim, uma conexão entre seu estado presente e os estados passados. Esse novo modelo recebeu a denominação de *modelo em espaço de estados* (DORF e BISHOP, 2009, p.105-107; MAYA, 2011, p. 237-238).

O surgimento desse novo conceito de representação de sistemas levou à transição da teoria clássica de controle para outra distinta, tal e qual uma nova série de problemas fossem doravante tratáveis sob a ótica de um novo paradigma, denominado de *controle moderno* (TAKAHASHI, 1998, p. 12-13).

1.1 CONTROLE DE SISTEMAS MULTIVARIÁVEIS

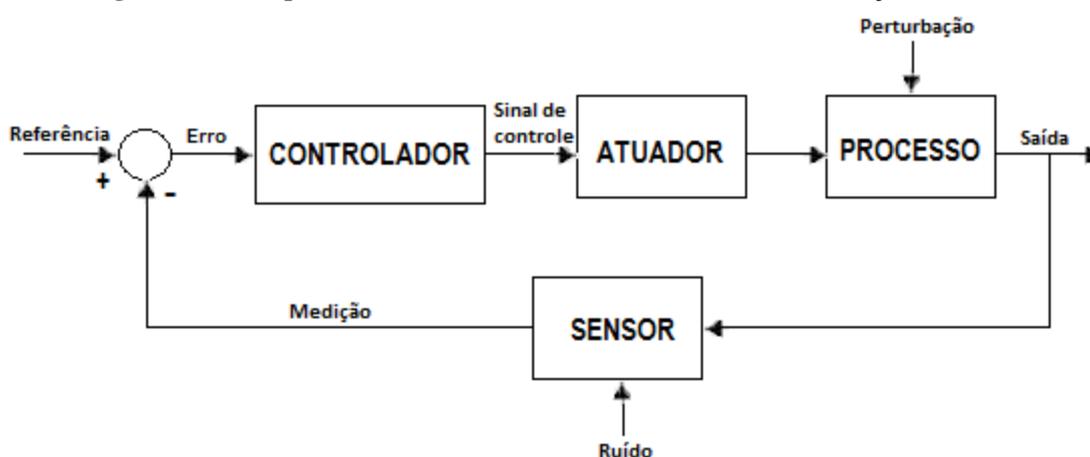
Para o caso de controle dos sistemas multivariáveis, um método proposto consiste no fato de, ao invés da utilização de um único sistema MIMO, realiza-se a fragmentação deste sistema em múltiplos subsistemas SISO menores (BERTACHI et al, 2013, p. 2). O principal objetivo, nesse caso, é dissociar a dinâmica do sistema e, desta forma, suprimir o acoplamento existente entre estas variáveis, com o intuito de se conceber sistemas com variáveis totalmente (ou ao menos aproximadamente) independentes, resultando em uma abordagem de controle executada de maneira descentralizada (ALBERTOS e SALA, 2003, p. 101, p. 127, p.136-146).

Em muitos casos, devido ao desacoplamento executado sobre as variáveis do sistema, tem-se como consequência a depreciação na influencia de certos parâmetros em relação aos

sinais de saída da planta, causando, dessa forma, dificuldade no ajuste do controlador e não raramente presença de instabilidade no sistema desacoplado. Como também, por utilizar cancelamento entre pólos e zeros do sistema, esse tipo de técnica não deve ser aplicado aos sistemas que apresentem incertezas paramétricas, ou termos de fase não mínima (DIAS, 2010, p. 17).

Já em sistemas cuja representação baseia-se na utilização de funções de transferência ou que apresentem incertezas nos parâmetros do modelo, o processo de realimentação é feito a partir da medição apenas da variável de saída, como representado na Figura 1.1.

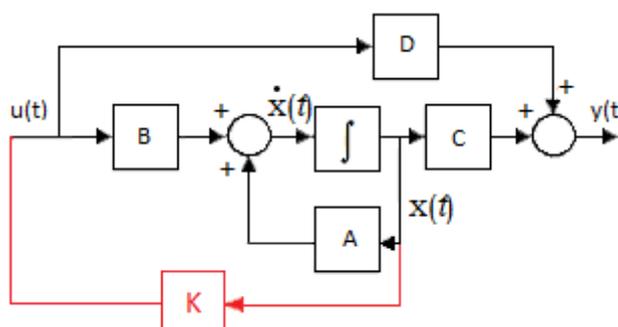
Figura 1.1 - Componentes de um sistema de controle com realimentação de saída.



Fonte: Autora.

Por meio da utilização do modelo em espaço de estados, existe a possibilidade de se fazer a realimentação de cada variável de estado do sistema, o que implica em um controle mais eficaz, com o mínimo de complexidade e que atenda aos requisitos de estabilidade e desempenho para o sistema (MAYA, 2011, p. 275). Na Figura 1.2 é apresentado um diagrama de blocos vetorial de um sistema de controle de realimentação de estados.

Figura 1.2 - Diagrama de blocos vetorial de controle com realimentação de estados.



Fonte: Adaptado de ARAÚJO (2007).

Neste tipo de abordagem, $u(t)$ é denotado como função das variáveis de estado e o procedimento de realimentação deverá corresponder ao controle de parâmetros fornecidos através de informações sobre estas variáveis em forma de medidas. Contudo, em aplicações práticas, freqüentemente existe a possibilidade de um ou mais estados não estarem disponíveis, assim como impossibilidade de se implementar recursos tecnológicos, dado que mesmo quando verificada a disponibilidade, por vezes, o acesso a estas variáveis exige o emprego de sensores de alto custo, inviabilizando economicamente sua aplicação (MAYA, 2011, p. 287).

Sendo assim, a concepção do projeto de um sistema de controle via realimentação de estados se resume à definição de uma matriz de ganhos de realimentação K (vide Figura 1.2) que posicione os pólos do sistema em malha fechada de forma a proporcionar ao processo o alcance do comportamento temporal desejado.

Nesse sentido, a partir da década de 60, os trabalhos de Rudolf E. Kalman e David G. Luenberger possibilitaram o desenvolvimento de métodos como: o *filtro de Kalman-Bucy* (KALMAN, 1960; KALMAN e BUCY, 1962) e o *observador de Luenberger* (LUENBERGER, 1964), respectivamente. Estes métodos são responsáveis por permitir estimações dos estados do sistema a despeito da presença de sinais ruidosos nos sensores, imprecisões nas equações que definem o modelo matemático e supressão da influência de elementos externos ao processo, porém, à custa do aumento considerável da complexidade do sistema em malha fechada, diminuição da robustez em relação à utilização de sensores físicos, especialmente quando os parâmetros da planta variam durante a operação (ELLIS, 2004, p. 192).

Em 1972, os estudos publicados por Jürgen Ackermann em (ACKERMANN et al., 2002, p. 37-44) levaram ao desenvolvimento de uma fórmula batizada em sua homenagem. A *fórmula de Ackermann* proporciona maior flexibilidade para o projeto de controladores em espaço de estados no que diz respeito aos índices de desempenho estabelecidos para o sistema em malha fechada, relacionando a alocação dos pólos desejados às variáveis internas do sistema (MAYA, 2011, p. 286). No caso de sistemas MIMO, uma análise baseada no posicionamento de pólos e zeros é substancialmente mais complicada (TROFINO e COUTINHO e BARBOSA, 2003, p. 12), o que faz com que o projeto via alocação de pólos pelo projetista torne-se, por vezes, não-trivial. Além do mais, para utilizar a fórmula de Ackermann, é necessário conhecer a matriz de controlabilidade do sistema e, segundo OGATA (2000, p. 599), apesar da maioria dos sistemas físicos serem controláveis, os

modelos matemáticos correspondentes podem não apresentar as propriedades de controlabilidade.

Outro problema verificado com relação aos métodos analíticos de projeto baseados em alocação de pólos é com relação à necessidade de observação da robustez em malha fechada a fim de caracterizar o desempenho dos sistemas de controle com realimentação, ou seja, o projeto deve levar em consideração o comprometimento da sensibilidade do sistema em relação a erros na modelagem como também eventuais alterações dos parâmetros da planta, rejeição de perturbações e atenuação dos ruídos de medida (DORF e BISHOP, 2009, p. 153-168).

Entretanto, na prática o que frequentemente se verifica são diferenças entre a planta e sua representação matemática que podem ser devidas: a não observação dos limites de acurácia na operação dos dispositivos de medidas e atuadores, o fato da estrutura e ordem do modelo não serem conhecidas em altas frequências, não linearidades e/ou mudança das condições de operação (MARCHIORI, 1994, p. 7).

Porém, uma caracterização satisfatória deve levar em consideração estas diferenças de modo a referenciar na construção teórica seus possíveis erros associados e muitas destas representações correspondem à descrição de sistemas lineares invariantes no tempo, o que significa que estes são regidos por dinâmicas com comportamento linear cujos parâmetros de controle permanecem imutáveis ao longo do tempo.

A classe de sistemas que apesar de considerados lineares, apresentam variação em seus parâmetros com o passar do tempo, caracterizam-se pela elevada complexidade necessária à sua análise e projeto, em relação aos sistemas lineares invariantes no tempo (GOLNARAGHI e KUO, 2012, p. 11, AGUIRRE, 2007, p. 52-54). Ainda segundo GOLNARAGHI e KUO (2012, p. 11) todos os sistemas físicos existentes são não lineares de alguma maneira, uma vez que os modelos matemáticos correspondem apenas às aproximações, visando apenas à simplicidade de análise e projeto de controle. Assim, para que um sistema seja considerado linear, as magnitudes dos sinais de controle devem ser limitadas às faixas nas quais os componentes do sistema possuam características lineares. Caso contrário, se os sinais de controle distendem além das zonas de operação lineares, o sistema não apresentará dinâmicas aproximadamente lineares, e assim como os demais sistemas considerados não lineares, passarão a ser considerados de difícil tratamento matemático, não dispondo de métodos gerais para se resolver uma ampla classe desses sistemas.

De acordo com OLIVEIRA (2006, p. 1-4), existem abordagens de controle que envolvem não linearidades intrínsecas, como o controle neural, o controle *fuzzy*, sistemas

chaveados, controle à estrutura variável por modos deslizantes (*Sliding Mode Control – SMC*), etc. Porém, uma parcela considerável dessas teorias considera o pressuposto do total conhecimento acerca do sistema de controle e de suas respectivas variáveis de estado. Hipótese esta que na prática nem sempre pode ser considerada razoável, uma vez que, como mencionado, em sistemas práticos, parte das variáveis encontra-se indisponível devido às limitações referentes à medição por parte dos sensores.

Essas variáveis de estados de difícil acesso podem ser aproximadas a partir da utilização de estimadores de estado (MAYA, 2011, p. 287-289, DORF e BISHOP, 2009, p. 551-554), mas ainda assim, torna-se necessário um conhecimento satisfatório acerca do modelo do processo, o que não é possível em se tratando de sistemas sujeitos a incertezas em seus parâmetros, com dinâmicas não modeladas ou a distúrbios externos.

Por essa razão, a partir da década de 70 até os dias de hoje, a evolução da já citada teoria de Controle Moderno se deu paralelamente ao desenvolvimento da teoria de *Controle Robusto*, a partir de preocupações com relação ao tratamento de incertezas paramétricas (TAKAHASHI, 1998, p. 14-17).

E nesse sentido, com relação à tentativa de se contornar as deficiências citadas no tocante aos controladores de realimentação a partir de ganhos fixos destinados a aplicações as quais o desempenho esteja submetido às mudanças bruscas das variáveis de processo, possíveis variações nos parâmetros das plantas, características dinâmicas não-lineares e essencialmente variantes no tempo, foram desenvolvidas as técnicas de *controle adaptativo*. Nas estratégias de controle adaptativo busca-se promover adaptações ao meio através da sintonização adequada do controlador, uma vez que um controlador que apresente comportamento dinâmico linear com parâmetros estáticos por vezes torna-se incapaz de fornecer flexibilidade suficiente para atender todas as especificações de desempenho, mediante estas variações e incertezas apresentadas pela planta (OLIVEIRA, 2013, p. 1-2).

Este tipo de estratégia tem a característica de permitir ao controlador ajustar-se automaticamente às mudanças detectadas, preservando, assim, o desempenho almejado para o sistema de controle mesmo na presença de não-linearidades, variações na dinâmica da planta assim como no âmbito do sistema (LUIZ et al, 1997, p. 1; OLIVEIRA, 2006, p. 1). Por essa razão, torna-se evidente a necessidade de se monitorar constantemente as saídas de interesse e assim, ajustar adequadamente os parâmetros de controle à medida que alterações, representando não cumprimento dos critérios de desempenho, sejam detectadas. Não obstante, propõe-se a busca por ferramentas que possibilitem adaptação em tempo real dos parâmetros do controlador.

Sendo assim, em virtude das limitações apresentadas nos métodos analíticos tradicionais de alocação de pólos, torna-se relevante a busca por técnicas capazes de minimizar total ou parcialmente estas desvantagens, admitindo a possibilidade de construção de sistemas reais, levando em consideração uma série de restrições.

A partir da observação de tais restrições se pode priorizar a definição de elementos passíveis de construção em um mundo físico em razão do cumprimento de certos requisitos de operacionalidade como, limite de potência dos atuadores, custo financeiro, confiabilidade, robustez, e assim por diante (BARÃO, 1997, p. 9).

Para isso, é necessária a aplicação de uma teoria de controle que leve em consideração não somente a aplicação de procedimentos matemáticos aliada ao princípio conceitual da teoria de controle para a elaboração de um controlador, mas também o conhecimento suposto a respeito das circunstâncias que delimitam o sistema a ser controlado. A partir destas técnicas, além dos requisitos de estabilidade, deverá ser possível fornecer um parecer em relação ao comportamento dinâmico desejado para o sistema, a fim de melhorar a resposta transitória e em regime permanente, ainda na fase de projeto.

Assim, os *sistemas de controle inteligentes* devem possuir considerável grau de independência de modo a detectar alterações na planta ou no ambiente, serem hábeis no tratamento de incertezas, da análise qualitativa da informação recebida, de estruturas de dados complexos e informação dos sensores. Devendo ainda possuir metodologia que possibilite tomadas de decisões mesmo na presença de informações insuficientes, gerar modelos aproximados do sistema e, finalmente, executar ações de forma a modificar, caso seja necessário, o desempenho do sistema na ocorrência de situações atípicas.

Nesse sentido, esses sistemas são aplicáveis não somente a projetos de controle, como também abordam técnicas capazes de simular características inerentes a determinados arranjos biológicos ou retratar um processo de adaptação ou aprendizado através da simulação da aquisição de conhecimento empírico humano, por meio de dados numéricos, atendendo às necessidades de controle às quais o sistema estará submetido. (COELHO e MARIANI, 2006, p. 2 e MENDONÇA, 2011, p. 12-13).

1.2 OTIMIZAÇÃO COMPUTACIONAL

Paralelamente ao avanço da teoria de controle moderno, o desenvolvimento tecnológico e inerente aumento da dinâmica observada no campo da automação implicaram na necessidade de otimização e controle dos processos industriais.

Tais necessidades acarretaram em grande interesse por parte da comunidade científica pela busca de técnicas capazes de definir um controlador que otimize o sistema a ser controlado, uma vez que, em situações práticas, segundo DA LUZ (2009, p. 1), o projetista de controle defronta-se com um dos maiores desafios referentes às áreas de ciência e tecnologia, no qual predomina o conflito existente entre a necessidade de melhoria na produtividade e às restrições orçamentárias, aliadas a requisitos de segurança.

Dessa forma, os sistemas de controle estão sujeitos a limitações de ordem física, e para tal, torna-se necessário o estabelecimento de uma relação adequada entre o controle ideal e aquele considerado aceitável dentro das restrições inerentes a cada aplicação (GEROMEL, 1986, p. 1-4). E de acordo com MAYA (2011, p. 280), em se tratando das técnicas analíticas usuais de projeto por alocação de pólos, o esforço de controle é frequentemente ignorado, levando à saturação dos dispositivos de controle.

A evolução dos computadores digitais e minimização da complexidade computacional relacionada à matemática do *cálculo variacional* impulsionou ainda mais o conceito de otimização em sistemas de controle (TAKAHASHI, 1998, p. 25). Nesse sentido, houve a dedução de fórmulas específicas que permitem o projeto de controladores otimizados para sistemas no espaço de estados, baseadas na resolução da equação algébrica de Ricatti, consistindo em um problema computacionalmente mais simples que o cálculo variacional (TAKAHASHI, 1998, p. 10).

Atualmente existe o favorecimento pela escolha de procedimentos fundamentados em métodos de projeto para plantas linearizadas, como o Regulador Linear Quadrático (em inglês, *Linear Quadratic Regulator – LQR*) e o Regulador Linear Quadrático Gaussiano (ou em inglês, *Linear Quadratic Gaussian Regulator – LQG*).

A partir de FIKAR e KOSTUR (2012, p. 7), o regulador linear quadrático destina-se aos casos onde todas as matrizes envolvidas são constantes, o tempo inicial é arbitrariamente definido como zero, e horizonte infinito de controle. Além do mais, não há um processo sistemático para definição das matrizes de ponderação Q e R (responsáveis por determinar a importância relativa do erro e o dispêndio de energia, respectivamente), considerando que a simples observação dessas matrizes não permite deduzir uma relação direta com a

observância dos índices de desempenho do sistema de controle. Assim, considera-se que estas são obtidas de forma empírica, a partir de um método de tentativa e erro, em contrapartida ao bom senso necessário por parte do projetista de controle relativo às premissas e formulações do problema.

O que se verifica é a continuação dos esforços da comunidade científica em se buscar técnicas com aplicações de relevante praticidade (BRANDINI, 2007, p. 29-33, p.35, FONTES, 2006, p. 1-3, p. 14). E o já citado aperfeiçoamento dos computadores digitais possibilitou o avanço de técnicas de otimização na síntese de controladores, por meio de métodos factíveis de projeto, permitindo, dessa forma, maior noção dos limites de desempenho esperado do controlador a ser projetado (ACKERMANN et al, 2002, p. 36; MEDEIROS, 2005, p. 2; NETO, 2000, p. 2-3).

Assim, surge o conceito de otimização computacional que, como o próprio nome sugere, baseia-se na tentativa de se ajustar determinados parâmetros de modo a satisfazerem às restrições iniciais e finais para determinado problema, a fim de se otimizar (maximizar ou minimizar) certos aspectos do processo, mantendo os demais dentro de suas limitações, envolvendo o desempenho do sistema em malha fechada como um todo (ADAMI, 1996, p. 1; FIKAR e KOSTUR, 2012, p. 1).

Os critérios a serem minimizados devem ser aqueles vinculados aos custos e descritos por meio de uma função denominada função de custo. Da mesma maneira, quando se necessita otimizar um critério visando sua maximização, busca-se definir uma função relacionada à sua transferência e/ou eficiência. Ambos os tipos de função representam equações matemáticas do processo a ser otimizado, conhecidas como funções objetivo ou, não raramente, como funções de *fitness* (FIKAR e KOSTÚR, 2012, p. 1).

Ainda, dentro do contexto de otimização computacional, existe o conceito de computação evolutiva, ou simplesmente EVOP (do inglês, *Evolutionary Optimization Approach*), a qual pode ser aplicável eficientemente a processos lineares, definidos ou indefinidos (FIKAR e KOSTÚR, 2012, p. 10).

Por meio da utilização de EVOP, as variáveis de controle podem ser sucessivamente ajustadas através de um procedimento iterativo de modo a convergirem em direção a um ponto de ótimo da função objetivo. E a existência de um modelo matemático para simulação do sistema torna o procedimento de otimização flexível, sendo aplicável a uma extensa gama de processos por meio de algoritmos fundamentados em técnicas convenientes de otimização combinatória ou através da aplicação de mecanismos heurísticos, nos quais as restrições devem ser observadas no decorrer de cada ciclo do algoritmo.

Uma maneira de se proceder consiste em priorizar a escolha da melhor solução em meio a um conjunto de alternativas possíveis, obtidas a partir da minimização ou maximização da função que descreva o propósito a ser alcançado. Assim, pode-se afirmar que o processo de otimização implica em buscar valores satisfatórios para todas as funções objetivo (FIKAR e KOSTÚR, 2012, p. 10), e por GLOVER e KOCHENBERGER (2003, p. 23-24, p. 91-94) e HSIAO e CHUANG e CHIEN (2004, p. 1-4), no caso da síntese de controladores, se resume à definição de um compensador que atenda aos objetivos de estabilidade, desempenho e viabilidade prática.

Para questões, cujas soluções exigem elevado grau de exatidão, demanda-se a aplicação de métodos exatos e eficientes, no entanto, torna-se inviável a aplicação de tais métodos à resolução de problemas complexos de otimização combinatória. Essas instâncias são do tipo NP, cujo tempo de processamento cresce exponencialmente à medida que se busca aumentar a precisão e eficiência em sua solução (FONTES, 2006, p. 3). Estes problemas, devido à complexidade presente e à conseqüente atenção necessária ao tempo de processamento, necessitam de métodos não-exatos baseados em heurísticas evolutivas fundamentadas em comportamentos de sistemas naturais e/ou sociais.

Sendo assim, o problema de controle para sistemas MIMO, previamente exposto, também pode ser considerado um problema NP (ver APENDICE A), visto que a expectativa de satisfação do desempenho é demasiadamente dificultada pela quantidade superior de entradas e saídas presentes no processo, elevando por vezes a ordem da matriz de realimentação necessária ao controle destes sistemas. Desta forma, merecem, portanto, serem otimizados através da utilização de métodos heurísticos de solução.

Vale salientar que nenhum método é capaz de resolver satisfatoriamente todo e qualquer problema de otimização, havendo, portanto, a necessidade de comparação entre as estratégias disponíveis e a escolha deverá ser concebida com base nas características gerais inerentes a cada uma, como simplicidade de implementação, velocidade de processamento e precisão nas soluções, além dos demais critérios e diretrizes que justifiquem a viabilidade da aplicação de determinada técnica, em detrimento às outras, para a resolução de um problema específico. Inclusive, caso seja necessário, existe a possibilidade de implementação de um algoritmo híbrido de otimização que proporcione o aproveitamento das vantagens de duas ou mais técnicas, compostos de uma fase de busca global e de uma fase de busca local (LI e XU e SHI, 2008, p. 4; CAVASOS e MOSS e O'BOYLE, s.d., p. 1-15, FONTES, 2006, p. 27).

1.3 REVISÃO BIBLIOGRÁFICA

Para as resoluções de diferentes problemas na área de engenharia de controle, algumas técnicas têm se destacado, dentre as quais pode-se citar: o arrefecimento simulado, o algoritmo genético (GA), otimização por enxame de partículas (PSO), busca tabu, otimização por colônia de formigas (ACO), dentre outras (FONTES, 2006, p. 3).

Em HSIAO e CHUANG e CHIEN (2004, p. 1-6), é proposto uma algoritmo baseado na técnica de Otimização por Colônia de Formigas para determinar os parâmetros de controladores PID, a fim de se obter com desempenho para determinada planta. Em HE e HAU (2011, p. 1-3), ainda com o objetivo de ajuste de parâmetros de um controlador PID, é utilizado um algoritmo adaptativo baseado na estratégia ACO, denominada pelos autores como AACA (*Adaptive Ant Colony Algorithm*), além de destacar características e vantagens a partir da aplicação do sistema de formigas para solução de diversos problemas, chamando a atenção para as desvantagens observadas para esse tipo de escolha, como o fato de possuir pobres limitações de convergência e cair facilmente e rapidamente em ótimos locais (soluções subótimas).

Em SIM e SUN (2002, p. 1-5) é discutida a aplicação do ACO no roteamento de redes (formigas correspondem aos pacotes viajando ao longo de uma rede de comutação), sobretudo em tratando-se da desvantagem observada deste possuir grande propensão à estagnação. E a fim de diminuir esse efeito (estagnação), é proposta uma nova abordagem, a *Multiple Ant-Colony Optimization* (MACO), através da adoção de várias colônias com diferentes tabelas de feromônio, cujo objetivo seria a busca de boas rotas, aumentando a adaptabilidade do sistema, reduzindo, assim, a possibilidade de estagnação.

Nos trabalhos de SWARANKAR e GUPTA e NIAZI, é apresentado um método para a reconfiguração de sistemas de distribuição radiais, tal que minimize a perda de potência real a partir de um algoritmo baseado em ACO adaptado de modo a sempre gerar topologias radiais viáveis, a partir da aplicação de teoria dos grafos.

Já em 2002, Algoritmos Genéticos foram aplicados à otimização e conformação da superfície do refletor de uma antena refletora *offset*, em ÁVILA (2002, p. 1-85), de maneira que, usando codificação real para os operadores genéticos, com o objetivo de melhorar a varredura do espaço de busca, se possa obter uma antena de satélite que produza um diagrama de radiação que cubra uniformemente todo o território brasileiro.

Em GUIMARÃES (2013, p. 1-72) é proposto um algoritmo fundamentado em Algoritmos Genéticos para sintonia de um controlador PID, em conjunto com a técnica de *Ziegler-Nichols*, para controlar a posição do rotor do motor CC, com rejeição a distúrbios.

Em 1995, a partir dos trabalhos de DIMEO e LEE (1995, p. 1-8), foi desenvolvido um Algoritmo Genético voltado a busca de um controlador Proporcional-Integral (PI) e um controlador por realimentação de estados para a planta de um sistema MIMO não linearizada de uma turbina a vapor. Em BELARBI e BELHANI e FUJIMOTO (2006, p. 1-11), é proposto um algoritmo genético com o objetivo de determinar os parâmetros de controladores *fuzzy* e compensadores *fuzzy* para sistemas multivariáveis de modelos de referência de uma coluna de destilação, assim como de robôs manipuladores com dois graus de liberdade. O controlador *fuzzy* é usado nos subsistemas monovariáveis desacoplados provenientes das plantas MIMO e um compensador *fuzzy* para as interações verificadas entre as variáveis, de modo que o compensador atue como um controlador *feedforward*, tendo como entradas as saídas dos controladores monovariáveis, consideradas como perturbações conhecidas, produzindo sinais de compensação adicionados aos sinais de controle, a fim de minimizar o efeito destas interações.

Em BAKIRTZIS (2002, p. 1-8) é utilizado um algoritmo genético para obtenção de soluções otimizadas do fluxo de potência com variáveis de controle contínuas (potência ativa e tensão de barramento de geradores) e discretas (configurações de dispositivos de derivação em um transformador).

Ainda no contexto de análise de desempenho em sistemas de controle, a otimização por enxame de partículas vem sendo frequentemente explorada como estratégia de obtenção de controladores aplicados a diversas finalidades práticas, devido sua facilidade de implementação e velocidade de convergência (ZIELINSKI et al., 2008, p. 1).

No trabalho de CHENG e ZHOU e YANG (2010, p. 1-6), é aplicado um algoritmo PSO para controle de MPPT (*Maximum Power Point Tracking*) em duas estruturas fotovoltaicas diferentes, em condições de sombreamento parcial, uma vez que, nestas situações, as curvas características dos painéis fotovoltaicos terão atributos tais que dificultam o rastreamento do ponto de máxima potência.

Em BIRLA e SWARUP (2009, p. 1-6), é proposta a aplicação de um algoritmo baseado em PSO para obtenção de controladores objetivando a melhoria do desempenho a partir da minimização da função objetivo da Integral do erro absoluto. Os resultados mostram que as soluções usando a técnica baseada em PSO são melhores em termos das características da

resposta transitória do sistema, controle de estabilidade e também em termos da função objetivo Integral de erro absoluto.

Em ZIELINSKI et al. (2008, p. 1-8), buscou-se a utilização do PSO para obtenção de um controlador PI (Proporcional-Integral) em cascata.

Em WAI e CHUANG e LEE (2010, p. 1-15), é proposto a análise da aplicação do PSO à busca de um controlador em tempo real de propulsão para um sistema de transporte para levitação magnética. Em KARANKI e MISHRA e KUMAR (2010, p. 1-11) foi verificado que um controlador baseado em PSO supera em qualidade de respostas um controlador obtido por meio do método LQR, quando aplicado a um controlador de qualidade energética.

Em AFROZI et al. (2011, p. 1-6) foram realizadas simulações de um motor assíncrono cujo controlador PI foi otimizado por meio de algoritmos inteligentes, dentre eles o PSO. Os resultados obtidos apontam para soluções melhores em relação aos controladores PID clássicos e *fuzzy*.

Por essa razão, na tentativa de se definir a melhor estratégia para resolver de maneira satisfatória o problema de controle aqui proposto, foram analisadas em trabalhos que antecederam à dissertação três metaheurísticas bio-inspiradas aplicadas à resolução de questões relacionadas à área de controle de processos. Sendo elas: o algoritmo genético, a otimização por colônia de formigas e a otimização por enxame de partículas. Nesse contexto, estas metaheurísticas foram aplicadas para fins de análise em (OLIVEIRA et al., 2010; BARROS et al., 2012; OLIVEIRA et al., 2013 e FARIAS, 2014).

Em 2010, no trabalho (OLIVEIRA et al, 2010, p. 1-6), foi feita a adaptação de um algoritmo genético para obtenção dos parâmetros de um controlador via realimentação de estados, destinado ao controle de um sistema com múltiplas entradas e múltiplas saídas, como alternativa aos métodos de tentativa e erro presentes nas técnicas analíticas de projeto de controle para sistemas multivariáveis, como por exemplo, os já mencionados métodos LQR e LQG. A partir de então, um GA sugere uma matriz de realimentação de estados responsável por estabilizar o sistema, com performance equivalente àquela obtida a partir das técnicas convencionais, além de atender aos requisitos de desempenho (de um tempo de estabilização igual ou inferior a seis segundos e máximo sobressinal limitado), sem a necessidade posterior de ajustes do controlador a fim de obter o comportamento ótimo para o sistema.

Posteriormente, em (BARROS et al, 2012, p. 1-8), os autores realizaram testes exaustivos para identificação de um conjunto de parâmetros (tais como, tamanho de população, diferentes valores de faixa de ganho dos elementos da matriz de realimentação e métodos de seleção de indivíduos, formando um total de 80 configurações possíveis para o

algoritmo) capazes de levar um sistema de quarta ordem com duas entradas e quatro saídas a soluções de boa qualidade.

No entanto, após vários ajustes relativos ao tamanho de população, tipo de seleção e inicialização do GA, o melhor conjunto de parâmetros sugerido forneceu boas soluções sob o ponto de vista de projeto. No entanto, o algoritmo demorava em média cerca de 8 segundos para definir uma matriz de controle composta por 8 parâmetros de ganhos de realimentação.

Dessa forma, pôde-se concluir que a complexidade temporal inerente a este tipo de abordagem impossibilita a utilização desta ferramenta à estimação em tempo real dos parâmetros do controlador.

Tabela 1.1 – Parâmetros utilizado pelo GA em BARROS et al. (2012).

Parâmetro AGD	Valor
Faixa valores	[-8, 8]
Tipo de seleção	Torneio
Taxa de cruzamento	60%
Taxa de mutação	2%
Tempo de acomodação	6 seg.
Máximo sobressinal	10%
Quantidade de indivíduos	80

Fonte: Adaptado de BARROS et al. (2012).

Em (OLIVEIRA et al; 2013, p. 1-6), com o intuito de se buscar uma nova alternativa que proporcionasse redução da complexidade temporal verificada em relação a utilização do GA proposto, um algoritmo PSO em sua versão simplificada foi empregado para a aquisição dos ganhos da matriz de realimentação do mesmo sistema MIMO de quarta ordem utilizado por BARROS et al. (2012) e seus resultados foram comparados àqueles obtidos por meio do algoritmo genético anteriormente proposto. A partir da utilização do PSO, verificou-se que foram obtidas saídas ligeiramente melhores em relação às saídas conseguidas a partir da utilização do GA para o mesmo sistema avaliado, apesar do PSO desenvolvido não ter passado por nenhuma campanha de testes para ajuste de seus parâmetros, a exemplo de peso inercial e coeficientes de aceleração que podem influenciar diretamente na convergência do algoritmo para uma solução.

Com base nestes resultados, ficou evidente que o algoritmo PSO conseguiu solucionar o problema de forma satisfatória evidenciando uma possível alternativa para o problema da complexidade temporal e, embora a utilização de algoritmos genéticos seja mais apropriada

em problemas cujo espaço de busca seja amplo, os GAs não são indicados para aplicações com elevadas restrições de tempo, na busca por soluções sub-ótimas.

Convém, dessa forma, a aplicação de testes para ajuste do algoritmo PSO, adaptação de parâmetros durante a execução, ou ainda o emprego de técnicas de divisão do enxame, inserindo certo nível de paralelismo a fim de tentar elevar este algoritmo para aplicação de ajuste em tempo real dos parâmetros do controlador.

A partir do desempenho satisfatório obtido pelo PSO, em comparação ao GA já conhecido e com configuração adequadamente definida, constatou-se que a estratégia de cooperação utilizada pelo PSO aliada à simplicidade no cálculo que define a evolução dos indivíduos da população, se sobrepõe à estratégia de competição proposta pelos GAs.

Além do fato de que a introdução dos mecanismos de cruzamento e mutação nos GAs aumentam a população na criação de novos indivíduos, e a sobrevivência dos indivíduos mais aptos implica necessariamente no descarte dos demais ao final de cada geração a partir das ferramentas de seleção. Este descarte pode ser caracterizado como um desperdício de recursos computacionais importantes, capacidade de processamento, energia e tempo em relação às estratégias nas quais as soluções apenas evoluem em direção a um ponto de ótimo. Nas aplicações em que seja necessário manter a população com uma grande quantidade de indivíduos, a fim de melhorar a diversidade da população, o desperdício de tempo de recursos computacionais aumentam.

Com isso, afirma-se que existe a possibilidade de se minimizar a complexidade temporal de modo a aplicá-lo à estimação de parâmetros em tempo real, a partir de:

- Mais testes e conseqüentemente melhor ajuste nos parâmetros do PSO;
- Busca de adaptação dos parâmetros durante sua execução;
- Uso de técnicas de paralelismo.

Tabela 1.2 – Parâmetros utilizado pelo PSO em OLIVEIRA et al. (2013).

Parâmetro PSO	Valor
Faixa de valores possíveis	[-8, 8]
Peso inercial w	0.6
Coefficiente aceleração c_1	0.8
Coefficiente aceleração c_2	0.8
Tempo de acomodação	6 seg.
Máximo sobressinal	10%
Quantidade de partículas	80

Fonte: Adaptado de OLIVEIRA et al. (2013).

Em 2014, foi proposto em (FARIAS et al; 2014, p. 1-6), a utilização de um algoritmo que faz uso de otimização por colônia de formigas (ACO) para a definição do mesmo tipo de matriz de controle. Foi constituído um conjunto composto por todos os controladores estabilizantes contidos no espaço de parâmetros, e em seguida, busca-se um subconjunto destes que proporcionam o atendimento das restrições impostas pelo projetista. No entanto, um procedimento analítico que possibilite tal caracterização não é uma tarefa fácil.

Com relação às respostas temporais, observou-se um comportamento transitório semelhante em comparação ao GA.

No caso do ACO, os autores chegaram à conclusão que um número menor de indivíduos (nesse caso, formigas) era suficiente para a definição de uma matriz capaz de fornecer soluções de qualidade equivalente àquelas por meio da aplicação do algoritmo genético de modo a sugerir complexidade temporal e espacial menores, em contraste ao GA. Este fato oportuniza sua aplicação em sistemas de ordens superiores ou em condições de rígidas restrições de tempo, como no caso de sistemas com adaptação em tempo real.

Assim, por meio dos resultados obtidos, no que diz respeito às condições relacionadas à menor complexidade temporal (algoritmo PSO) e redução do número de agentes (pelo ACO), evidenciam a necessidade de maiores estudos que motivem o emprego dessas estratégias para o controle de sistemas multivariáveis e com isso aplicá-las à estimação em tempo real dos parâmetros do controlador.

Tanto em OLIVEIRA et al (2013) quanto em FARIAS et al (2014), os testes realizados evidenciam o modo como a estratégia de cooperação adotada pelo ACO e PSO influencia o tempo de convergência do algoritmo. Em outras palavras, foi possível avaliar até que ponto a característica de cooperação existente entre as partículas do enxame e entre as formigas de uma mesma colônia (fato que a diferencia da estratégia de competitividade adotada pelos GAs), pode trazer benefícios, e assim, resultar em menor complexidade temporal para a aquisição de boas soluções.

Vale salientar que em nenhum dos trabalhos realizados até então, foi considerado sistemas que apresentam incertezas paramétricas, entradas com perturbações, ou que estes sejam variantes no tempo, o que merece ser analisado quando se intenciona ajuste em tempo real de parâmetros.

1.4 MOTIVAÇÃO

A partir dos conceitos que envolvem o projeto de controladores via realimentação de estados, assim como as principais dificuldades relacionadas às técnicas analíticas usuais de alocação de pólos, como também os fatores que dificultam suas aplicações em sistemas lineares variantes no tempo, com imprecisão nos parâmetros que regem suas dinâmicas, ou sujeitos a perturbações, objetiva-se a utilização de controle adaptativo para a resolução de tais sistemas.

Levando em consideração o problema de controle para sistemas MIMO, também considerado um problema da classe NP, e a necessidade de otimização para os processos atuais (aliada ao desenvolvimento do conceito de otimização computacional e as principais características que viabilizam seu emprego em sistemas de controle), busca-se a otimização de tais sistemas por meio de métodos heurísticos de solução.

Devido à constatação de que o controle para um mesmo sistema, porém obtidos a partir do uso das metaheurísticas ACO e PSO, asseguraram respostas dinâmicas com qualidades semelhantes aqueles obtidos a partir do emprego do algoritmo genético otimizado (cujos parâmetros foram definidos a partir de testes exaustivos). Os resultados alcançados até então, por meio da utilização das técnicas baseadas em estratégias que fazem uso de cooperação entre os indivíduos de sua população, indicaram vantagens como simplicidade e rápida convergência (no caso do PSO), e diminuição do número de agentes/formigas (no caso do ACO).

Verificou-se por meio da aplicação dessas metaheurísticas evolutivas que, a estratégia de cooperação na qual são baseadas as otimizações por enxame de partículas e por colônia de formigas, representa uma alternativa promissora por meio de resultados qualitativamente superiores em relação à estratégia de competição imposta pelos algoritmos genéticos no que diz respeito à diminuição da complexidade temporal, de modo a impulsionar estudos dessas técnicas voltadas a aplicações que envolvam adaptação em tempo real dos controladores destinados à sistemas multivariáveis com parâmetros variantes no tempo.

No caso da otimização por enxame de partículas, o algoritmo foi definido rapidamente, negligenciando-se a realização de uma quantidade de testes adequados à definição de um conjunto apropriado de parâmetros ótimos para sua aplicação de forma mais satisfatória ao problema de controle em questão, desconhecendo-se, dessa forma, a melhor configuração deste algoritmo para esse tipo de aplicação.

1.5 CONTRIBUIÇÕES

Em razão das dificuldades e limitações envolvendo os métodos usuais de síntese de controladores em problemas de otimização em controle, assim como os resultados obtidos em trabalhos que precederam esta dissertação, espera-se com o presente trabalho:

- a. Uma investigação da utilização de um algoritmo baseado na metaheurística otimização por enxame de partículas a fim de se obter controladores via realimentação de estados a serem aplicados a sistemas multivariáveis de quarta ordem, garantindo não somente estabilidade em malha fechada, mas também atendendo aos requisitos de desempenho desejados.
- b. Redução da complexidade temporal na busca por soluções satisfatórias, de modo a destiná-lo a aplicações que envolvam tomada de decisão e estimação em tempo real da matriz de controle por realimentação para sistemas sujeitos a incertezas paramétricas, variantes no tempo.

1.6 METODOLOGIA

Neste trabalho será abordada a prática envolvendo sistemas de controle adaptativos, buscando a estimação adequada do sistema a partir da medição das variáveis de estado do sistema, de modo a garantir que este mantenha um desempenho satisfatório, a despeito de incertezas e variações nos parâmetros de modelo.

O desenvolvimento da estrutura proposta baseia-se na tentativa de se definir um sistema flexível. E a partir da realização de testes, buscar-se-á definir um conjunto de parâmetros que representem boa sintonia do algoritmo PSO de modo a conduzi-lo às aplicações que permitam a tomada de decisão e estimação em tempo real da matriz de controle.

Vale ressaltar que, a fim de se testar a solução proposta, os métodos comuns de tentativa de se modelar incertezas de parâmetros são suprimidos a fim de se produzir modelos com características de variação mais severas em relação àqueles comumente utilizados.

Espera-se definir, por meio de simulações, um limite máximo da taxa de amostragem para os sistemas genéricos sujeitos às características de variações aqui avaliadas de modo que o algoritmo proposto possua comportamento satisfatório em relação à aquisição de um

controlador ótimo durante o processo de estimação e adaptação em tempo real do sistema de controle .

1.7 ORGANIZAÇÃO DA DISSERTAÇÃO

No capítulo 1 é feita uma explanação e análise dos principais métodos usuais de projeto de controle multivariável além de destacar as principais desvantagens limitantes ao emprego destas técnicas em sistemas MIMO e/ou variantes no tempo. Ainda neste capítulo é abordado o problema de projeto de controladores para processos que apresentam incertezas nos parâmetros, ou com perturbações em suas entradas. Busca-se também introduzir o conceito de controle adaptativo e sua aplicação em plantas incertas, finalizando-o por meio da motivação para a realização deste trabalho, a metodologia proposta, assim como a expectativa de contribuições relativas ao presente estudo.

No capítulo 2 é apresentada uma breve descrição das principais técnicas de controle adaptativo e da metaheurística populacional utilizada neste trabalho, denominada: Otimização por Enxame de Partículas, de modo a introduzindo informações importantes à estrutura adaptativa e construção do algoritmo PSO desenvolvidos.

O capítulo 3 diz respeito ao método de adaptação do algoritmo PSO ao cálculo da matriz de realimentação de estados de um sistema de quarta ordem genérico, assim como a maneira pela qual o vetor correspondente a uma solução candidata à resolução de determinado problema de controle pode ser avaliado de modo a se verificar o comportamento do sistema em malha fechada, assim como, maior detalhamento das principais etapas, a partir das quais se fundamenta o controle adaptativo proposto pelo presente trabalho de dissertação, de modo a definir a sequência de passos necessários à reprodução da estratégia aqui sugerida.

O capítulo 4 é reservado ao estudo de análise dos principais resultados obtidos a partir da utilização do algoritmo proposto, por meio do qual se busca estabelecer uma lista de dados a fim de se fixar uma média aproximada de tempos para sistemas com diferentes perfis de variação de parâmetros.

Por fim, no capítulo 5, finaliza-se a presente dissertação a partir das conclusões e sugestões para possíveis trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

A seguir, será apresentado um breve relato sobre a metaheurística bio-inspirada PSO, comprovadamente aplicada à área de controle de processos (capítulo 1, seção 1.3), assim como da introdução à teoria de controladores adaptativos, de modo a familiarizar o leitor com as principais particularidades inerentes a estas abordagens.

2.1 ABORDAGENS EVOLUTIVAS

Os métodos heurísticos evolutivos são baseados na dinâmica de agentes responsáveis pela obtenção de soluções, a princípio escolhidas aleatoriamente em meio a uma população inicial e que convergem gradualmente em direção a resultados satisfatórios.

O modo como se dá a evolução dos indivíduos no algoritmo genético é baseado em competição. Neste tipo de abordagem, a principal diferença em relação à estratégia de cooperação reside no modo como se dá a evolução da população ao longo das iterações do algoritmo. Segundo BJORKOY (2011, p. 2-4), para os sistemas que fazem uso desse tipo de estratégia, cada componente se esforça para alcançar um objetivo ao derrotar (ou superar) os demais, a partir de uma métrica comum para todos os participantes que define o sucesso a ser obtido. Através da competição, as soluções na população tornam-se melhores ao longo do tempo antes que alcancem uma solução considerada aceitável para a tarefa atual.

Em um sistema impulsionado pela concorrência, ao permitir que cada componente realize melhorias em suas soluções por meio da competição, o desempenho de cada componente melhora individualmente, resultando em uma série de soluções aplicáveis ao objetivo final (BJORKOY, 2011, p. 2) e a eliminação dos indivíduos com baixa aptidão com o intuito de fugir de regiões que apresentam expectativa inferior no que diz respeito à exploração, implica, ao mesmo tempo, em perda de informação relevante na busca por novos mínimos locais que conduzissem o algoritmo a soluções de melhor qualidade em relação às já analisadas (FARIAS et al, 2013, p. 2; MOEDINGER, 2004, p. 45).

Em contraste com a competição imposta pelos algoritmos genéticos, a otimização por colônia de formigas apresenta como estratégia de evolução um comportamento cooperativo, no qual os indivíduos atuam em conjunto para um objetivo final comum.

Assim como os algoritmos que fazem uso de competição como método de evolução, na estratégia de cooperação, a população inicial é gerada aleatoriamente. Da mesma forma, se

mantém o procedimento de avaliação de indivíduos. A principal diferença entre elas reside no fato de que a componente de seleção, no caso de cooperação entre agentes, é removida (BJORKOY, 2011, p. 2-3).

Nesses sistemas devem possuir agentes aptos a perceber o ambiente em redor e modificá-lo localmente. E é essa habilidade que possibilita a comunicação entre os indivíduos do sistema, uma vez que de alguma maneira captam as mudanças ocorridas no ambiente devido às decisões tomadas por seus semelhantes, e conseqüentemente conduzem a população como um todo em direção a um comportamento global correspondente a uma solução possível para determinado problema.

A partir do exposto, é possível constatar que a abordagem cooperativa favorece rápida convergência do algoritmo, no entanto é mais propensa a manter-se presa em ótimos locais, minimizando, assim, a qualidade das respostas para a resolução do problema. Ao longo do processo de evolução populacional, um método baseado na competitividade entre os agentes, tende a produzir soluções mais próximas do ótimo global, mas devido à diversidade e aleatoriedade presentes na massa de indivíduos, a velocidade de convergência por vezes pode ser reduzida.

2.2 OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

A metaheurística conhecida como otimização por enxame de partículas (do inglês, *Particle Swarm Optimization*) começou a ser concebida a partir de 1995, quando o psicólogo social James Kennedy e um engenheiro eletricitista chamado Russel Eberhart propuseram em (KENNEDY e EBERHART, 1995, p. 1-7) um algoritmo cuja origem fundamenta-se na observação do comportamento social de nuvens de insetos, cardumes e revoadas de pássaros. Estes indivíduos possuem em comum a característica de fazerem uso de sinergia para alimentarem-se, o que sugere a existência de uma possível troca de informação entre os indivíduos do grupo.

No caso de um bando de pássaros, existem indivíduos que podem estar distribuídos ao longo da formação síncrona ou encontrarem-se separados do grupo a fim de encontrar alimento. Ao passo que uma fonte de alimento é identificada por um dos integrantes, este é imediatamente seguido pelos demais componentes do bando que mudam rapidamente de direção, reagrupando-se em uma nova configuração de voo, e assim, o grupo inteiro migra para a região onde há a presença de comida.

Um modelo de algoritmo apropriado à tarefa de otimização via enxame de partículas visa à reprodução desse comportamento psicossocial, no qual existe a tendência de se seguir determinadas condutas comprovadamente eficazes em busca do sucesso adquirido por alguns indivíduos de determinado grupo (KENNEDY e EBERHART, 1995, p. 2).

2.2.1 O Algoritmo PSO

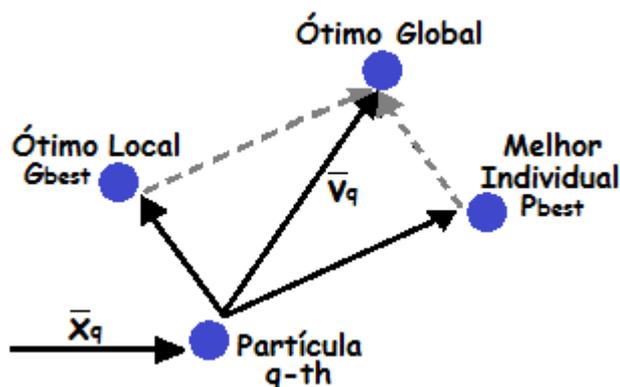
No PSO, a massa de soluções representa o bando de pássaros, o espaço de busca corresponde à região sobrevoada, a movimentação das aves em busca de alimento descreve o desenvolvimento e evolução do conjunto de soluções em direção a respostas que satisfaçam aos requisitos pré-estabelecidos pelo projetista, e por fim, a descoberta de comida constitui o alcance da solução ótima (EBERHART e SHI, 1998, p.2).

Assim, em um algoritmo básico de otimização por enxame de partículas, é definido inicialmente uma quantidade n de partículas, cujas posições em um espaço D -dimensional representam uma solução em potencial devendo ser ajustada com base na observação de sua vizinhança, a partir de quesitos como, passado recente desse indivíduo, e a sugestão de uma pontuação dada ao movimento desenvolvido por ela, em relação à melhor solução já encontrada por qualquer indivíduo da população (inclusive ela mesma) de modo que convirja, juntamente com as outras partículas, para um determinado ponto do espaço (KENNEDY e EBERHART, 1995, p. 4).

A velocidade de uma partícula, por sua vez, será definida como a mudança de direção de sua posição no espaço, criando um vetor-velocidade que sofrerá variações no decorrer do processo de otimização, representando a troca de informação dentro do enxame (SHI e EBERHART, 1999, p. 1; SERAPIÃO, 2009, p. 6).

A Figura 2.1 representa uma partícula que muda sua direção com base na melhor solução encontrada anteriormente por ela mesma, como também em relação a melhor solução obtida pela vizinhança. A interação entre estes dois vetores velocidades resulta em um deslocamento em direção a um novo ótimo local, o qual convergirá a cada iteração ao ponto considerado como ótimo global para o sistema.

Figura 2.1 – Mudança de direção de uma partícula.



Fonte: Autora.

Portanto, a velocidade \vec{v}_q de cada indivíduo dentro do enxame muda a cada iteração do algoritmo, e pode ser descrita matematicamente a partir da equação (2.1):

$$\vec{v}_q(t+1) = w\vec{v}_q(t) + c_1r_1(p_{best} - \vec{x}_q(t)) + c_2r_2(g_{best} - \vec{x}_q(t)) \quad (2.1)$$

De modo que:

- w → representa o peso que caracteriza um equilíbrio entre a exploração local e global, ou seja, define a inércia da partícula, a qual controla a maneira como os valores de velocidade em iterações anteriores têm influencia sobre a velocidade a ser desenvolvida atualmente, propiciando assim, a escolha do modo como o espaço de busca deve ser explorado. Portanto, valores altos para o peso inercial indicam privilégio da exploração global – evidenciada pelo aumento do vetor-velocidade em direção a novos ótimos –, à medida que valores menores favorecem a pesquisa em direção a regiões menores no espaço. O valor da inércia pode variar a cada iteração de modo a permitir sua redução ao longo do tempo, aumentando o desempenho alcançado pelo algoritmo (SHI e EBERHART, 1998, p. 1-3);

- c_1 e c_2 → devem ser, ambas, constantes maiores que zero, e têm por objetivo definir a importância relativa entre o mérito individual de cada partícula (melhor solução individual) e a interação entre os indivíduos do grupo (a partir do ótimo local), respectivamente;

• r_1 e r_2 → correspondem a números aleatórios gerados no intervalo entre [0,1], os quais proporcionam um caráter estocástico ao método (SHI e EBERHART, 1999, p. 1);

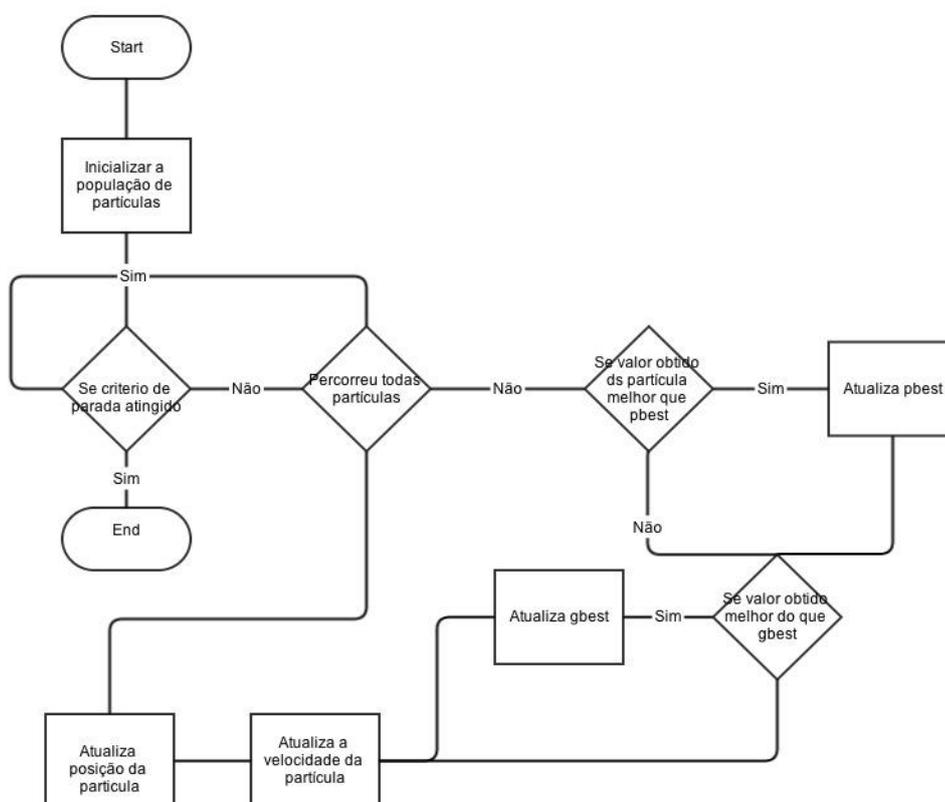
• p_{best} e g_{best} → correspondem respectivamente a melhor solução individual da q -th partícula (memória cognitiva) e a melhor solução encontrada pelos demais indivíduos do enxame, também conhecido como memória social (KENNEDY e EBERHART, 1995, p. 3).

E a nova posição \bar{x}_q da partícula é definida pela soma da posição atual e pela nova velocidade obtida em (2.1), de acordo com a equação (2.2):

$$\bar{x}_q(t+1) = \bar{x}_q(t) + \vec{v}_q(t+1) \quad (2.2)$$

O fluxograma representado na Figura 2.2 descreve os passos a serem seguidos para a implementação de um algoritmo PSO básico:

Figura 2.2 – Fluxograma para implementação básica de um algoritmo PSO.



Fonte: Autora.

Sendo assim, em comparação aos demais tipos de algoritmos, é possível verificar que o cálculo no PSO é mais simples (poucas linhas de código e apenas duas equações lineares regem o cálculo de atualização de cada indivíduo), e a pesquisa é realizada de tal maneira que apenas a partícula com melhor aptidão transmite informação às demais, elevando consideravelmente a velocidade de pesquisa (EBERHART e KENNEDY, 1995, p.1).

O algoritmo PSO faz uso tanto de cooperação quanto de competição para evolução das soluções e, dessa forma, as soluções candidatas inicialmente encontram-se distribuídas ao longo do espaço de busca, mas com o passar das iterações, estas tendem a convergir para uma região mais propensa à existência de um “ponto de ótimo”, o que acaba comprometendo a diversidade de indivíduos do algoritmo (ROLDÁN, 2005/2006, p. 6; EBERHART e SHI, 1998, p. 4-5).

Como não é possível se comprovar que a região para a qual a população está sendo direcionada corresponde efetivamente a um ótimo global para o problema, além da crescente perda de diversidade que os impossibilita explorar as diferentes áreas existentes no espaço, faz com que estes tipos de algoritmos acabem frequentemente convergindo para ótimos locais (LOPES, 2012, p. 10, p.12; MOEDINGER, 2005, p. 16; ARBOLEDA, 2012, p.44).

Uma maneira de se melhorar a exploração do espaço de busca, seria aumentar os “saltos” dos indivíduos a cada iteração. Isso faria com que a exploração local fosse afetada, beneficiando, assim, a busca por regiões relativamente distantes da posição atual de cada membro existente. Como consequência, a população continuaria a evoluir em direção a uma região mais apropriada, porém agora mantendo uma maior heterogeneidade entre os indivíduos do grupo (MOEDINGER, 2005, p. 26-27).

No entanto, à medida que se aumenta a discrepância entre as soluções que representam os estados atual e futuro de cada um dos agentes, se pode verificar a ocorrência de uma maior divergência entre os elementos da população. E ao passo em que a direção seguida leva a regiões distantes entre si e em relação ao ótimo global, a retomada do curso que levaria à solução que represente um ponto de máximo (ou mínimo) global torna-se, por vezes, impraticável.

Por essa razão, alguns trabalhos sugerem a utilização de uma componente responsável por proporcionar um balanço entre os procedimentos de busca local e global, de modo a ajustar a distância de salto da população. Assim, nas primeiras iterações do algoritmo, os agentes variam sua localização de maneira mais acentuada (com “saltos” maiores) possibilitando uma maior exploração do espaço, e ao passo em que se aumenta o número de iterações, essas distâncias são reduzidas de modo a fazer com que os indivíduos passem a

procurar em sua própria vizinhança, buscando aperfeiçoar as soluções obtidas até o momento (as quais se supõem já suficientemente próximas a uma região de ótimo).

A partir da aplicação deste tipo de prática, torna-se possível o aproveitamento conjunto das características de diversificação e intensificação, aumentando, assim, as chances de se alcançar uma zona de ótimo global. O PSO configura um exemplo da aplicação desse tipo de estratégia por meio de SHI e EBERHART (1998, p. 1-3) na qual o peso inercial pondera um equilíbrio entre a exploração local e global.

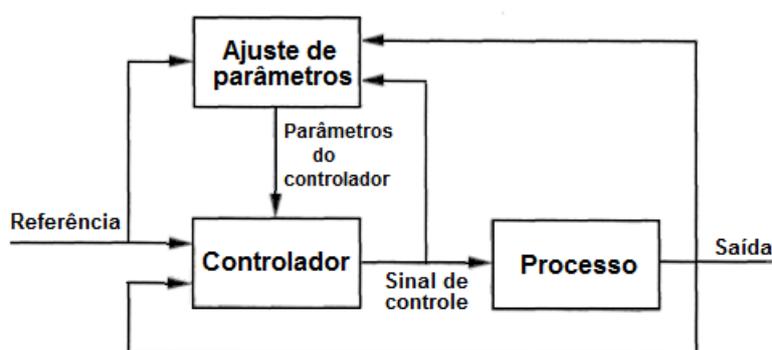
Dessa forma, a velocidade em um algoritmo que faz uso de estratégia cooperativa, pode ser comedida a fim de se manter a execução com velocidade de convergência relativamente rápida, no entanto evitando que esta aconteça de forma prematura.

Estas análises são importantes uma vez que, considerando-se o comportamento dos indivíduos, também se avalia métricas como tempo de execução, eficiência e tempo de desenvolvimento (BJORKOY, 2011, p. 10).

2.3 CONTROLE ADAPTATIVO

Entende-se por sistemas adaptativos, aqueles que possuem a capacidade de “adaptarem-se”, de modo a mudarem seu comportamento para tratar novas circunstâncias. Sendo assim, entende-se por controlador adaptativo, um controlador capaz de modificar seu comportamento em respostas a mudanças identificadas na dinâmica do processo e presença de distúrbios (ASTROM E WITTENMARK, 2008, p. 1). Em outras palavras, um controlador adaptativo é um controlador que possui parâmetros ajustáveis e apresenta um mecanismo voltado para ajuste de parâmetros. A Figura 2.3 representa um diagrama de blocos que descreve um sistema de controle adaptativo.

Figura 2.3 – Diagrama de blocos de um sistema adaptativo.



Fonte: Adaptado de ASTROM E WITTENMARK (2008).

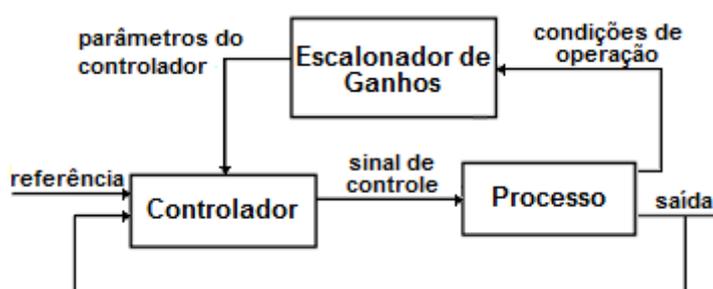
A seguir, serão apresentados quatro tipos de esquemas de sistemas adaptativos, sendo eles: controle adaptativo por escalonamento de ganhos, controle adaptativo por modelo de referência, controle dual e o regulador auto-ajustável.

2.3.1 Controle Por Escalonamento De Ganhos

Em muitos casos, é possível encontrar variáveis mensuráveis que se relacionam bem com as mudanças na dinâmica do processo. Essas variáveis podem então ser utilizadas para obtenção dos novos parâmetros do controlador.

Nesse caso, o regime é inicialmente utilizado para medir o ganho e, posteriormente, programar (mudar) o controlador para compensar as alterações nos ganhos do processo (ASTROM E WITTENMARK, 2008, p. 19). Em outras palavras, a principal idéia no controle por escalonamento de ganhos (do inglês, *Gain Scheduling*) é compensar as variações no processo mudando os parâmetros do controlador em função das condições de operação. A principal vantagem desse tipo de controlador é que esse proporciona uma mudança rápida dos parâmetros do controlador, uma vez que não há necessidade de estimação dos mesmos. No entanto, para este tipo de estratégia, a compensação é feita em malha aberta.

Figura 2.4 – Diagrama de blocos de um sistema de controle por escalonamento de ganhos.



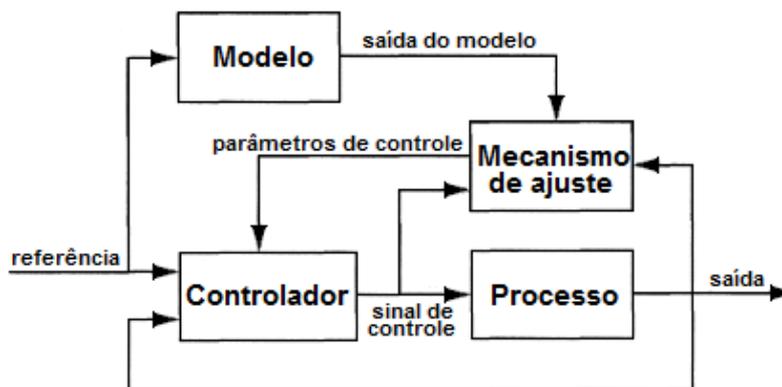
Fonte: Adaptado de ASTROM E WITTENMARK (2008).

2.3.2 Sistema Adaptativo Por Modelo De Referência

No sistema adaptativo por modelo de referência (do inglês, *Model-Reference Adaptive System - MRAS*) as especificações de desempenho são dadas por um modelo de referência, de modo a especificar como o sinal de saída ideal deve responder ao sinal de comando, tal que o controlador pode ser pensado como sendo de dois *loops*. O *loop* interno consiste de uma

malha de realimentação comum, constituída pelo controlador e o processo, ao passo que o *loop* externo ajusta os parâmetros do controlador em função de uma medida de erro, dado pela diferença entre a saída do modelo de referência e a saída da planta, de modo que este permaneça pequeno (ASTROM E WITTENMARK, 2008, p. 20).

Figura 2.5 – Diagrama de blocos de um controlador adaptativo por modelo de referência.



Fonte: ASTROM E WITTENMARK (2008).

O principal problema envolvendo o *MRAS* consiste em definir o mecanismo de ajuste de modo a buscar um sistema estável capaz de minimizar o sinal de erro (ASTROM E WITTENMARK, 2008, p. 20). O ajuste dos parâmetros do controlador é dado em função da derivada da sensibilidade do erro em função dos parâmetros de controle.

2.3.3 Regulador Auto-Ajustável

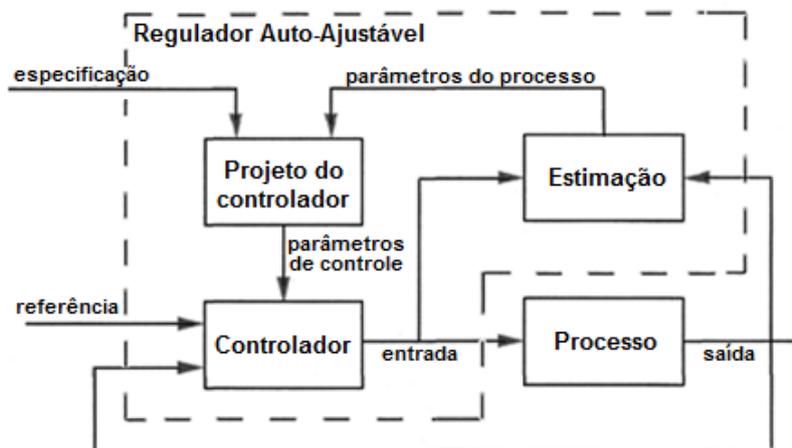
No caso do Regulador auto-ajustável (*Self-Tuning Regulator – STR*), A idéia básica é separar o procedimento de estimação dos parâmetros da etapa do projeto do controlador, de modo que os parâmetros desconhecidos são estimados em tempo real, assumindo que o processo tem parâmetros constantes, porém desconhecidos.

Assim como no caso do *MRAS*, o sistema de controle adaptativo *STR* pode ser entendido como constituído por dois *loops*. Um *loop* interno constituído por uma malha de realimentação comum, ao passo que os parâmetros do controlador são ajustados no *loop* externo, constituído por um bloco estimador e outro responsável pelo projeto do controlador (ASTROM E WITTENMARK, 2008, p. 22).

Os parâmetros estimados são assumidos como verdadeiros, através do princípio da equivalência à certeza, ao passo que o bloco denominado “projeto do controlador” fornece

uma solução em tempo real do problema de controle para um sistema de controle a parâmetros conhecidos.

Figura 2.6 – Diagrama de blocos de um Regulador auto-ajustável.



Fonte: Adaptado de ASTROM E WITTENMARK (2008).

De acordo com LUIZ et al (1997. p.1), os sistemas adaptativos *STR* podem ser obtidos segundo dois procedimentos: direto (implícito) e indireto (explícito). Para o caso direto é utilizado um estimador recursivo para obter diretamente os parâmetros da lei de controle de realimentação a partir de medidas de entrada e saída do processo. Já no caso indireto um estimador recursivo é utilizado para obtenção dos parâmetros do modelo da planta aliado a um algoritmo responsável pelo procedimento de projeto do controlador, no qual, os parâmetros do controlador são ajustados por um estimador de parâmetros recursivo e um cálculo de projeto.

O modelo do processo e o projeto de controle são atualizados a cada período de amostragem, a partir do qual se torna evidente a necessidade de se monitorar constantemente as saídas de interesse e assim, ajustar adequadamente os parâmetros de controle à medida que as alterações, representando não cumprimento dos critérios de desempenho, sejam detectadas. Não obstante, é necessária a busca por ferramentas que possibilitem adaptação em tempo real dos parâmetros do controlador, assim como diferentes combinações de métodos de estimação e métodos de projeto levam a reguladores com diferentes propriedades.

2.3.4 Controle Dual

O controle dual corresponde a um tipo de estratégia de controle estocástico não-linear que objetiva levar a saída para determinado valor desejado, no entanto, introduz perturbações quando as estimativas são incertas (ASTROM E WITTENMARK, 2008, p. 22).

Deste modo, um controlador com características duais estabelece um balanço entre manter um bom controle e os erros de estimação pequenos. Uma consequência principal é que as incertezas nos parâmetros estimados são levadas em conta no projeto do controlador. Esta abordagem leva ações especiais devido ao conhecimento pobre sobre o processo. No entanto, é de projeto complexo, o que inibe sua aplicação em problemas práticos (ASTROM E WITTENMARK, 2008, p. 22).

O modelo de um processo com incertezas nos parâmetros é obtido pela ampliação das variáveis de estado do processo e o ambiente pelo vetor de parâmetros cuja dinâmica é descrita por (2.3) com uma distribuição inicial que reflete a incerteza nos parâmetros.

$$\frac{d\theta}{dt} = 0 \quad (2.3)$$

Neste tipo de descrição não existe distinção entre estes parâmetros e as outras variáveis de estado. Isto significa que o controlador resultante pode manipular mais rapidamente variações de parâmetros.

O objetivo de controle é formulado de modo a minimizar uma função de custo dada por:

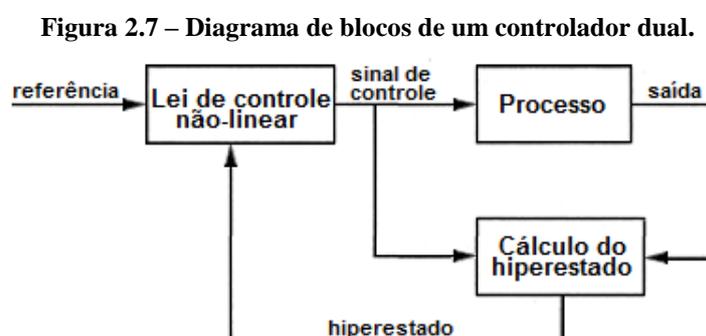
$$V = E(G(z(T), u(T)) + \int_0^T g(z, u) dt) \quad (2.4)$$

Sendo:

- $E \rightarrow$ A esperança matemática, que considera todos os valores iniciais e distúrbios que aparecem no modelo;
- $u \rightarrow$ A variável de controle;
- G e $g \rightarrow$ funções escalares de z e u .
- $z \rightarrow$ Consiste em um estado aumentado do estado do processo e os parâmetros, tal que $z = (x^T \theta^T)^T$.

O problema de encontrar um controlador que minimiza a função custo (2.4) é não-trivial. Uma solução pode ser obtida, através de programação dinâmica, dada em termos de uma equação funcional, denominada equação de Bellman, uma extensão da equação de Hamilton-Jacobi (ASTROM E WITTENMARK, 2008, p. 23).

O controlador é composto de duas partes: um estimador não-linear e uma realimentação de controle. O estimador gera uma distribuição de probabilidade condicional do estado para as medidas, $p(z|y,u)$, denominada hiperestado do problema. E o controlador realimentado é uma função não-linear que mapeia o hiperestado dentro do espaço das variáveis de controle (ASTROM E WITTENMARK, 2008, p.23).



Fonte: Adaptado de ASTROM E WITTENMARK (2008).

2.4 CONCLUSÕES DO CAPÍTULO

No capítulo 2, a metaheurística populacional PSO, assim como uma introdução à teoria de controladores adaptativos, utilizadas no presente trabalho, foram discutidas de maneira sucinta, uma vez que tratam-se de técnicas já devidamente consolidadas, cujas bibliografias encontram-se facilmente disponíveis para consulta.

Ainda neste capítulo, é feita uma análise entre a estratégia de competição (proposta por alguns tipos de metaheurísticas, como os GAs) e àquela que dispõe de uma abordagem cooperativa (adotada pelo ACO) para proporcionar a evolução da população em busca de uma solução satisfatória para determinado problema.

3 PROPOSTA DE CONTROLE

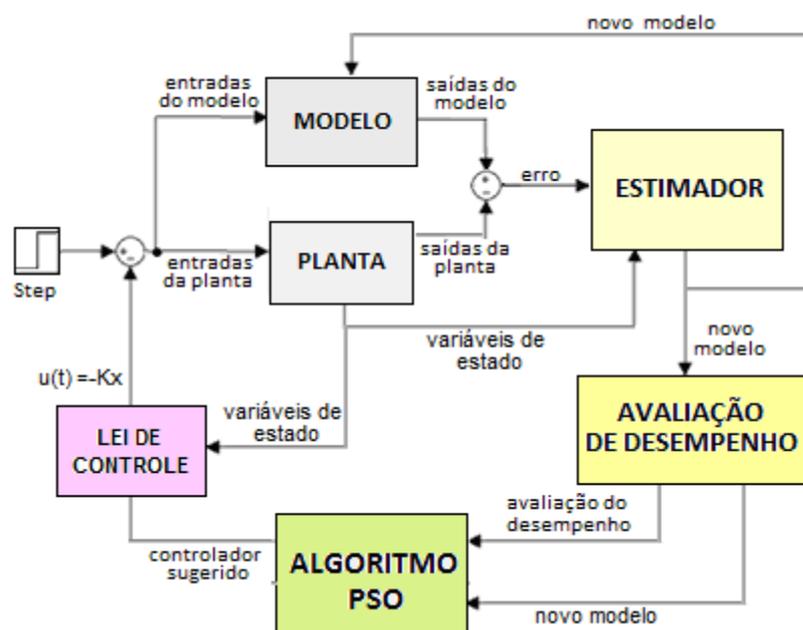
Nesse capítulo, é abordado o emprego da metaheurística PSO voltada a fins de controle, assim como as estratégias de avaliação e estimação aqui utilizadas.

3.1 ESTRUTURA DO SISTEMA PROPOSTO

O capítulo 3 aborda o emprego da metaheurística PSO voltada a fins de controle, assim como das estratégias de avaliação e estimação utilizadas.

Para a validação da metodologia proposta, o algoritmo de controle foi encapsulado a partir da utilização do pacote computacional MATLAB/Simulink. Neste contexto, a implementação do sistema de controle em *software* visa à simulação do comportamento da referida estrutura em tempo real. A Figura 3.1 representa o diagrama de blocos simplificado para o sistema de controle pretendido.

Figura 3.1 – Diagrama de blocos para o sistema de controle adaptativo proposto.



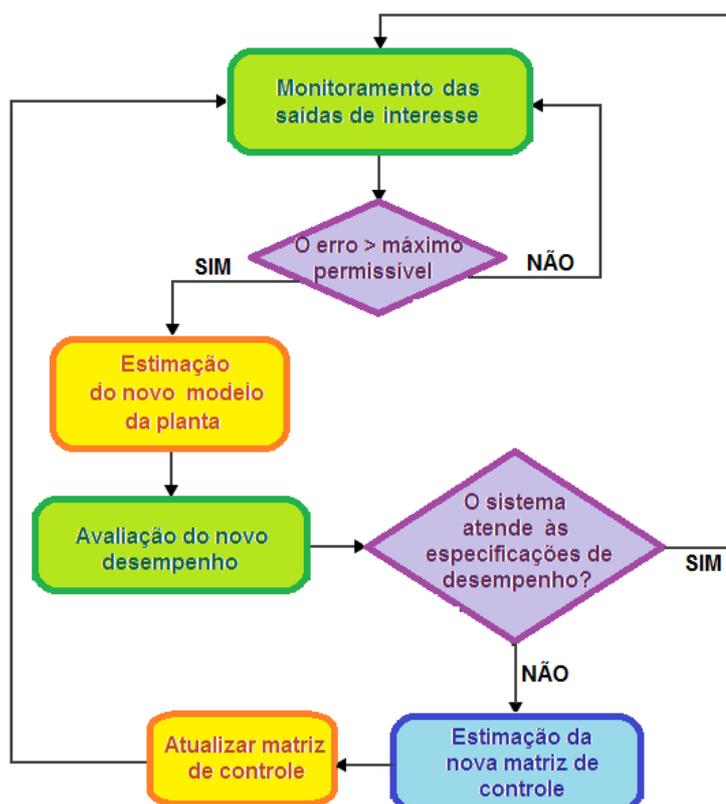
Fonte: Autora.

Por meio da Figura 3.1 é possível observar que o sistema adaptativo é composto por um bloco contendo o modelo vigente e outro correspondente à planta. Quando a diferença entre as

saídas da planta e do modelo ultrapassam determinado valor de erro máximo permitido verifica-se a ocorrência de variação e um novo modelo deverá então ser estimado, a partir de medidas das variáveis de estado do sistema. De posse do novo modelo, é feita a análise do novo desempenho de modo a se averiguar se, mesmo após constatação da ocorrência de variação, segue atendendo aos requisitos pretendidos para o desempenho. Caso seja verificado que o novo sistema não atende a algum dos requisitos impostos, um novo controlador deverá então ser calculado para o novo modelo.

Desse modo, o sistema de controle será então obtido através do cumprimento das etapas descritas pelo fluxograma da Figura 3.2, as quais serão abordadas detalhadamente na sequência.

Figura 3.2 - Etapas do controle adaptativo proposto.



Fonte: Autora.

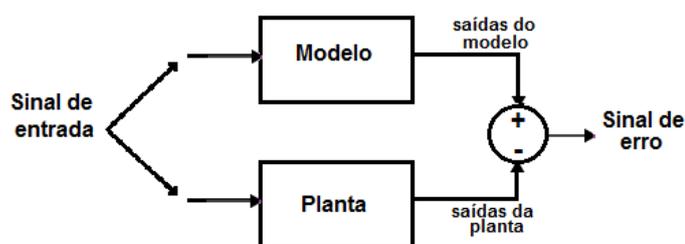
3.2 MONITORAMENTO DAS SAÍDAS DE INTERESSE

O objetivo do controle adaptativo é fornecer uma lei de adaptação de modo a garantir ao sistema seguir um desempenho satisfatório, independente das incertezas e/ou variações de parâmetros presentes.

Para isso, é feito o monitoramento das saídas de interesse a cada instante de tempo. Desse modo, um sinal de erro quantifica a diferença entre as saídas do sistema físico e as saídas provenientes do bloco contendo o suposto modelo vigente para o sistema.

Ao se verificar que a diferença entre a saída da planta e do modelo ultrapassa o limite permitido para o erro máximo, admite-se uma ocorrência de variação dos parâmetros da planta.

Figura 3.3 - Monitoramento das saídas e cálculo de erro.



Fonte: Autora.

O pseudocódigo do algoritmo utilizado para análise do erro é apresentado na Figura 3.4.

Figura 3.4 - Pseudocódigo do algoritmo para análise do erro máximo permitido entre as saídas da planta e das saídas do modelo.

Pseudocódigo: Algoritmo para análise do erro máximo

- 1 Iniciar a variável *variação* igual a 0;
 - 2 Atribua um valor para o *erro máximo permitido*
 - 3 Para cada instante de tempo faça
 - 4 Atualize as saídas do modelo;
 - 5 Atualize as saídas da planta;
 - 6 Se as saídas do modelo $> (1 + \text{erro máximo permitido}) * \text{saídas da planta}$
 - 7 | Faça *variação* igual a 1;
 - 8 Se as saídas do modelo $< (1 - \text{erro máximo permitido}) * \text{saídas da planta}$
 - 9 | Faça *variação* igual a 1;
 - 10 Retorne a variável *variação*
-

Fonte: Autora.

Nota-se que o erro é dado em função da proporção entre os sinais de saída da planta, das saídas do modelo e um valor de tolerância para o erro admissível, uma vez que existem situações nas quais o sinal de referência é considerado baixo e situações onde a referência consiste de valores relativamente elevados, o que torna difícil quantificar um nível único de erro suficientemente aceitável para diferentes situações.

3.3 ESTIMAÇÃO DOS PARÂMETROS DO NOVO MODELO DA PLANTA

Para o processo de estimação de parâmetros do modelo, foi desenvolvido um algoritmo fundamentado na minimização da soma dos quadrados dos resíduos existentes entre os valores estimados e aqueles medidos.

A fim de se validar a estratégia aqui proposta, buscar-se-á modificar todos os parâmetros da matriz de realimentação de estados do sistema, de modo a obter um sistema com características distintas daquele supostamente conhecido.

No seguinte trabalho, toma-se como pressuposto o acesso a todas as variáveis de estados e, a partir delas, será calculada a nova matriz de estados para o sistema, a partir da equação de transição de estados em (3.1).

$$x(t) = e^{At} x(0) + e^{At} \int_0^t e^{-\tau} BU(\tau) d(\tau) \quad (3.1)$$

Como as informações referentes aos valores das variáveis de estado serão colhidas a cada instante de tempo posterior à detecção de variação ocorrida no sistema, utiliza-se a equação de estados para um sistema discretizado (3.2a) e de saídas (3.2b):

$$X(k) = \varphi(\Delta t) X(k-1) + \Delta t BU(k) \quad (3.2.a)$$

$$y(k) = CX(k) + DR(k) \quad (3.2.b)$$

Sendo:

- $\Delta t \rightarrow$ Corresponde ao passo de iteração definido;
- $\varphi(\Delta t) \rightarrow$ Matriz de transição de estados do sistema em tempo discreto, e equivale a $e^{A \cdot \Delta t}$;
- $B \rightarrow$ Matriz de entrada do sistema MIMO;
- $U(k) \rightarrow$ Sinal aplicado às entradas do sistema;

- $C \rightarrow$ Matriz de saídas do sistema MIMO;
- $D \rightarrow$ Matriz de transmissão direta do sistema MIMO;
- $R(k) \rightarrow$ Referência para o sistema.

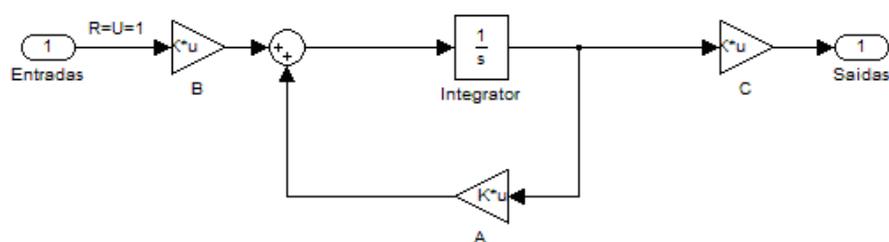
Dessa forma, a variável de estado atual $X(k)$ depende do sinal de entrada U também nesse mesmo instante de tempo, ao passo que $U(k)$ é função dos valores adquiridos por Δt , B , assim como das variáveis de estado no instante imediatamente anterior, as quais constam as parcelas referentes ao somatório dos sinais aplicados às entradas do sistema nos instantes de tempo anteriores. Em outras palavras, os valores aplicados às entradas variam, uma vez que receberão incrementos ao longo do tempo, e $U(\tau)$ em (3.2.a) não corresponde necessariamente ao valor do sinal de controle aplicado ao sistema, mas sim ao tipo (comportamento) do sinal de entrada. Em malha aberta $U(\tau)$ corresponde ao sinal de referência utilizado. Dado que em um sistema em malha aberta em espaço de estados as equações são:

$$\dot{x} = Ax + Bu(t) \quad (3.3)$$

$$y = Cx + DR(t) \quad (3.4)$$

Desse modo, considerando a matriz de transferência D igual à zero, em uma apresentação em blocos de um sistema genérico em espaço de estados, tem-se o seguinte diagrama de blocos:

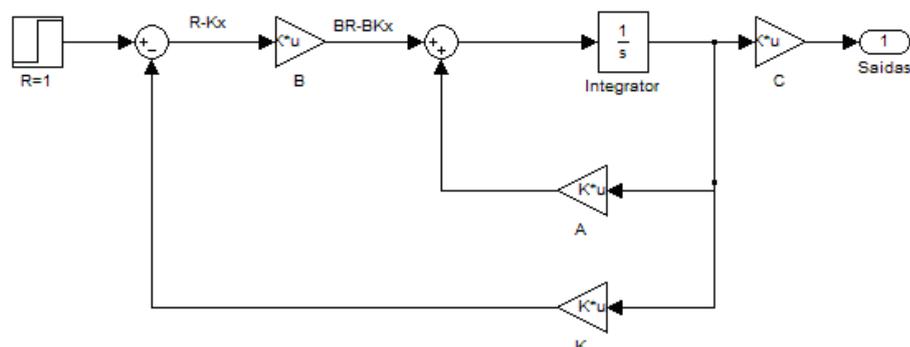
Figura 3.5 – Diagrama de blocos de um sistema em espaço de estados em malha aberta.



Fonte: Autora.

Assim, para um sistema em malha fechada, o sinal de entrada $u(t)$, exemplificado pela Figura 3.6, será então quantificado a partir das contribuições dos valores correspondentes ao sinal de referência R e das variáveis de estado x devidamente realimentadas a partir da matriz K de ganhos de realimentação.

Figura 3.6 – Diagrama de blocos de um sistema em espaço de estados em malha fechada.



Fonte: Autora.

Em malha fechada, a variável de estado \dot{x} é então definida como:

$$\dot{x} = Ax - BKx + BR(t)$$

Após alguns ajustes, pode-se reescrever a equação de estados para um sistema com realimentação de estados, como segue:

$$\dot{x} = (A - BK)x + BR(t) \quad (3.5)$$

Caso a referência para o sistema seja igual à zero, as variáveis de estado tenderão a zero em regime permanente ($t \rightarrow \infty$), e as saídas de interesse tenderão a infinito com o passar do tempo, porém, a partir da aplicação de um sinal de referência limitado, as respostas do sistema em regime permanente tenderão a um valor também limitado (e constante se $R(t)$ for igualmente constante) ao longo do tempo.

Dessa forma, a equação de estados para um sistema discretizado (3.2.a) será utilizada no algoritmo do estimador a partir de uma análise estatística da relação existente entre uma variável dependente e outra com características independentes, com o intuito de se obter o modelo que explique como se dá a variação de $x(k)$ em função da variável independente $x(k-1)$.

O algoritmo proposto utiliza o método dos mínimos quadrados (MMQ) a fim de buscar a minimização da soma dos quadrados dos resíduos provenientes da diferença entre os valores estimados e aqueles efetivamente medidos, de forma a se maximizar o grau de ajuste do modelo em relação a todos os dados observados. Ou seja, a partir da aplicação do método dos

mínimos quadrados, o valor mais razoável para a grandeza Y a partir de n medições y_i será aquele que minimiza a soma dos quadrados das diferenças entre Y e y_i , tal que:

$$\sigma = \sum_{i=1}^n (y_i - Y_i)^2 \quad (3.6)$$

Dessa forma, admitindo-se linearidade e tomando-se a equação responsável por descrever o comportamento de uma reta (3.7), nota-se que a equação linear de estados (3.2.a) poderá ser equiparada aos estimadores cujos valores são aqueles assumidos para as matrizes de estados e de entrada do sistema em espaço de estados.

$$f(x) = a + bx \quad (3.7)$$

Estes valores de $\phi(\Delta t)$ e $\Delta t.B.R(t)$ correspondem a “ b ” e “ a ” em $f(x)$, respectivamente. Assim, aplicando o método dos mínimos quadrados à tentativa de obtenção dos coeficientes “ b ” e “ a ”, equivale à estimação dos parâmetros $\phi(\Delta t)$ a partir da obtenção de “ b ” e de $\Delta t.B.R(t)$ a partir de “ a ”, que especificam (3.2.a), tal que:

$$\sigma = \sum_{i=1}^n (y_i - f(x_i))^2 = \sum_{i=1}^n (y_i - a - bx_i)^2$$

No presente trabalho, considera-se que a matriz de entradas B não sofre variações significativas ao longo do tempo, de modo que será aqui considerada constante e precisamente conhecida.

Logo, o processo de estimação corresponde à tentativa de se obter aproximadamente o valor correspondente da variável “ b ” na equação $f(x)$ citada.

Para isso, assume-se que “ b ” corresponde à região na qual a função objetivo (3.6) é mínima. Esta região é definida como aquela onde o ponto $(b, a = \Delta t.U.R(t))$ minimiza o quadrado do erro entre os valores medidos e aqueles estimados, em outras palavras, toma-se a derivada de primeira ordem da equação (3.6) em função de “ b ”, da forma:

$$\frac{\partial \sigma}{\partial b} = -2 \sum_{i=1}^n (y_i - a - bx_i).x_i \quad (3.7)$$

Igualando a expressão (3.7) à zero, especifica-se o valor de b que aproxima (3.6) de seu valor mínimo. Para isto, a expressão adquirida será definida como segue:

$$0 = \sum y_i x_i - \sum a x_i - \sum b x_i^2$$

Fazendo-se os devidos arranjos, chega-se à expressão (3.8) utilizada pelo estimador MMQ desenvolvido:

$$b = \frac{\sum x_i y_i - a \sum x_i}{\sum x_i^2} \quad (3.8)$$

Deste modo, a partir das aplicações das correspondências necessárias aos termos das (3.2.a) e (3.8) chegam-se às seguintes relações:

- $x_i \rightarrow$ Indica os valores adquiridos pelas variáveis de estado no instante anterior;
- $y_i \rightarrow$ Representa as variáveis de estado na iteração atual;
- $b \rightarrow$ Determina a matriz $\phi(\Delta t)$ de transição de estados do sistema discretizado;
- $a \rightarrow$ Corresponde à porção acrescida a cada instante de tempo Δt das quantidades $B.R(t)$ aplicadas às entradas.

Uma vez de posse do valor equivalente a $\phi(\Delta t)$, o padrão compatível com a matriz de estados do sistema em malha fechada, $A - BK$, pode ser facilmente adquirido a partir da multiplicação do inverso da quantidade equivalente a uma amostra de tempo pela matriz logarítmica do $\phi(\Delta t)$ obtida em (3.8).

A Figura 3.7 representa o pseudocódigo do algoritmo utilizado para estimação do novo modelo para sistema a partir da ocorrência da variação descrita no subitem 3.2.

No pseudocódigo do bloco estimador, tem-se:

- ***E_AmenosBK*** \rightarrow Matriz de transição de estados $\phi(\Delta t)$ do modelo estimado;
- ***B*** \rightarrow Matriz de entrada do sistema em espaço de estados;
- ***C*** \rightarrow Matriz de saídas do sistema em espaço de estados;
- ***K*** \rightarrow Matriz de controle vigente;
- ***novo_A*** \rightarrow Matriz encontrada para o novo modelo em espaço de estados;
- ***Tolerancia_Erro*** \rightarrow Precisão exigida na aproximação do modelo;
- ***U*** \rightarrow Entradas do sistema;
- ***passo*** \rightarrow Passo de iteração Δt definido.
- ***achou_modelo*** \rightarrow Variável lógica que indica se os critérios de parada foram estabelecidos;

- *Soma de X* → Somatório do vetor de variáveis de estado nos instantes anteriores ao longo das iterações, correspondendo à $\sum x_i$.
- *Soma de X ao quadrado* → Somatório do vetor de variáveis de estado nos instantes anteriores multiplicado pelo vetor de variáveis de estado na iteração atual transposto, de modo que seja equivalente à parcela $\sum x_i^2$.
- *Soma de Y vezes X* → Somatório do vetor de variáveis de estado multiplicado por seu vetor transposto de modo que este seja equivalente à $\sum x_i y_i$.

Figura 3.7 - Pseudocódigo do algoritmo utilizado para estimação da nova matriz de estados do sistema.

Pseudocódigo: Algoritmo para estimação da nova matriz de estados do sistema

```

1  Iniciar variáveis
2  Enquanto critério de parada não for alcançado faça
3      Se for detectada variação faça
4          Para cada iteração k faça
5              Se k = 1 então
6                  Faça X = 0;
7              Se k > 1 então
8                  Faça Y = Variáveis de estado atuais;
9                  Faça X = Y anterior;
10                 Faça Soma de X = Soma de X + X;
11                 Faça Soma de X ao quadrado = Soma de X ao quadrado + X * X;
12                 Faça Soma de X vezes Y = Soma de X vezes Y + Y * X;
13                 Se k > número mínimo de leituras desejadas
14                     Faça E_AmenosBK = (Soma de X vezes Y - passo * B * U * Soma de X) / (Soma de X
                        ao quadrado);
15                     Faça saída real = C * Y;
16                     Faça saída com novo modelo = C * (E_AmenosBK * X + passo * B * U);
17                     Se o erro entre a saída real e a saída com o novo modelo > Tolerancia_Erro
18                         Extraia AmenosBK a partir de E_AmenosBK
19                         Extraia novo_A sabendo que A = (A - B * K) + B * K;
20                         Faça achou modelo = 1;
21                     Senão
22                         incremente K e refaça os passo de 8 a 20 até o novo modelo estar com erro
                        dentro da tolerancia exigida para precisão.
23                 Senão
24                     Faça novo_A = A anterior;
25                     Faça achou modelo = 0;
26  Retorne achou modelo e novo_A

```

3.4 AVALIAÇÃO DO DESEMPENHO DO MODELO ESTIMADO

Depois de estimada a nova matriz de estados do modelo é feita uma nova avaliação das características do novo sistema de controle a fim de se evitar desnecessariamente o recálculo da nova matriz de realimentação de estados.

Primeiramente, a estabilidade é verificada observando-se o posicionamento dos pólos do sistema em malha fechada. Caso seja constatada a translação de algum destes para o semiplano direito do eixo de coordenadas (ou sobre o eixo imaginário), o sistema é instável, o que evidencia a necessidade imediata de recálculo do controlador.

Caso o sistema permaneça estável, é feita então a análise de desempenho, de modo a verificar se este ainda atende aos requisitos previamente definidos pelo projetista.

O conjunto de critérios de desempenho estabelecidos como forma de avaliação das respostas transitórias no corrente trabalho constitui-se dos máximos valores permitidos para a porcentagem de sobressinal e do tempo necessário para que os sinais de saída permaneçam dentro da faixa de dois por cento do valor em regime.

Em sistemas de primeira e segunda ordem, esses índices de desempenho podem ser facilmente obtidos a partir da utilização de fórmulas simples, frequentemente citadas na literatura, como exemplo cita-se a equação para o cálculo do sobressinal¹. No entanto em sistemas de ordem superior a dois, estes apresentam comportamento descrito por meio do somatório de dinâmicas de primeira e segunda ordem (MAYA, 2010, p. 91) de acordo com o posicionamento dos pólos no plano complexo.

Assim, os pólos, antes calculados para conferir a estabilidade do sistema, serão aproveitados para a obtenção do tempo de estabilização apresentado pelo sistema, de modo que:

$$t_s = \frac{4}{\xi\omega_n} \quad (3.9)$$

Sabendo-se que a parcela $\xi\omega_n$ coincide com a magnitude adquirida pela parte real do pólo responsável pela dinâmica mais lenta apresentada pelo sistema, utiliza-se o módulo da parte real do pólo mais próximo da origem em substituição à medida $\xi\omega_n$ para o cálculo do tempo de estabilização t_s em (3.9). Caso este apresente valor superior àquele previamente

¹ A equação para cálculo do máximo sobressinal em sistemas de segundo grau é: $Mp(\%) = e^{\frac{-\xi\pi}{\sqrt{1-\xi^2}}} \times 100$

estabelecido, o desempenho é considerado insatisfatório e uma nova matriz de realimentação é calculada sem a necessidade de se averiguar o sobressinal.

Uma vez atestado o cumprimento do requisito para o tempo máximo de estabilização, verifica-se finalmente a máxima porcentagem de sobressinal. Para se determinar o ponto de sobressinal, as saídas são calculadas fazendo uso das equações para o sistema discretizado (3.2.a) e (3.2.b).

Em geral, para a avaliação de máximo sobressinal demanda-se uma quantidade de tempo fundamentalmente importante ao processo de adaptação em tempo real. Por essa razão, em vez de se tomar um número dispendioso de leituras dos sinais de saída e a partir deles registrar o máximo valor obtido por cada curva de resposta, é preferível aqui uma estratégia que demande a menor quantidade possível de cálculos necessários à sua obtenção.

Assim, foi feito o mapeamento dos possíveis posicionamentos para os pólos já definidos, o que resultou em cinco diferentes configurações possíveis para este conjunto de singularidades.

3.4.1 Caso I

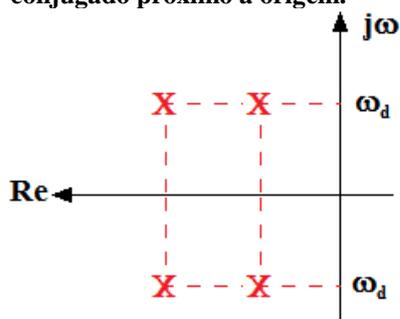
Corresponde à situação em que todos os pólos estão posicionados sobre o eixo real. Para estes casos o sistema é amortecido, caracterizado pela ausência de oscilações durante o regime transitório.

Assim, considerando que o sistema é livre de zeros de transmissão lentos, independente dos valores adquiridos por todos os quatro pólos do sistema, desde que possuam parte imaginária nula, o sobressinal será igual à zero.

3.4.2 Caso II

Outro arranjo possível corresponde à presença de dois pares de pólos complexos conjugados e nessa condição, o modo de obtenção do máximo sobressinal depende da distancia entre as partes reais dos pólos.

Figura 3.8 – Caso II.a: diagrama de pólos de um sistema de quarta ordem com um par complexo conjugado próximo a origem.



Fonte: Autora.

Se o par dominante está consideravelmente afastado do segundo par complexo, o cálculo do sobressinal é feito a partir da equação para sistemas de segunda ordem. Porém, se ambos estão relativamente próximos, as saídas são, então, calculadas até um tempo próximo ao tempo de estabilização.

Com o intuito de minimizar a quantidade de pontos a serem calculados, em todos os casos que demandam a plotagem das curvas para o cálculo do sobressinal, são estipulados passos de tempo Δt para cada solução analisada. A amplitude da maior frequência angular não amortecida ω_d equivale ao maior valor das partes imaginárias dos pares complexos, de modo que o período T das oscilações sofridas é inversamente proporcional à ω_d , tal que:

$$T = \frac{2\pi}{\omega_d} \quad (3.10)$$

Supõe-se que o máximo sobressinal ocorre em $T/4$. Desse modo, levando-se em consideração as possíveis defasagens entre os sinais de saída, são sugeridos passos de iteração iguais à oitava parte do período T esperado para as oscilações.

Desse modo, faz-se o cálculo das saídas a cada instante de tempo igual à oitava parte do período de oscilação até o total equivalente ao tempo de estabilização calculado. Em situações nas quais se busca apenas o máximo sobressinal Y_{\max} (e não a máxima ultrapassagem percentual), é necessário o cálculo até a ocorrência do pico máximo (não necessitando análise até o valor de regime Y_{Regime}), o que diminui sobremodo a quantidade de cálculos realizados, isto é:

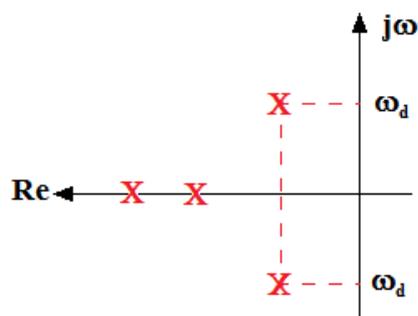
$$M_p (\%) = \frac{Y_{\max} - Y_{\text{Regime}}}{Y_{\text{Regime}}} \times 100 \quad (3.11)$$

3.4.3 Caso III

O terceiro caso é dado pela situação na qual se tem um par complexo conjugado e dois pólos sobre o eixo real. Este pode ainda ser subdividido em três outras situações possíveis:

3.4.3.1 Pólos complexos próximos à origem

Figura 3.9 – Caso III.a: diagrama de pólos de um sistema de quarta ordem com um par complexo conjugado próximo a origem.



Fonte: Autora.

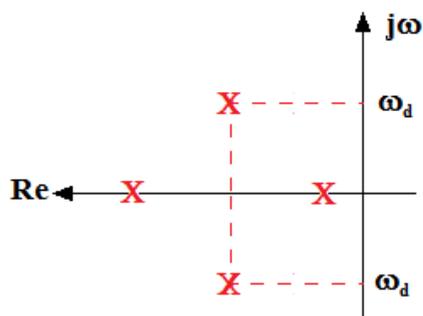
Nessa situação o par conjugado está mais próximo ao eixo complexo em relação aos dois pólos sobre o eixo real.

Sendo assim, verifica-se a posição da parte real do par dominante. Na circunstância em que estes se encontram posicionados de forma a possuírem dinâmica dominante em relação aos demais, utiliza-se a fórmula para o cálculo do sobressinal para sistemas de segundo grau. Caso contrário, as saídas do sistema são calculadas por (3.2) e o sobressinal poderá ser obtido por (3.11).

3.4.3.2 Pólos complexos entre os pólos reais

Quando o par complexo estiver de acordo com a Figura 3.10 verifica-se novamente o posicionamento do pólo mais próximo à origem.

Figura 3.10 – Caso III.b: diagrama de pólos de um sistema de quarta ordem com um par complexo conjugado situado entre os pólos reais.



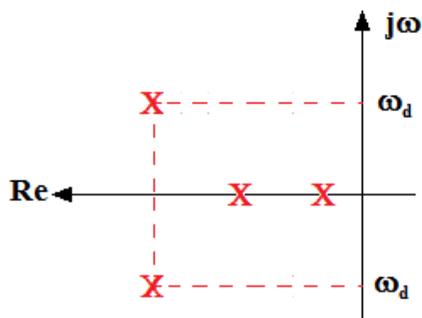
Fonte: Autora.

Caso este se encontre em uma posição dominante em relação ao par complexo, o sobressinal é considerado igual ou próximo à zero, caso contrário, as formulas (3.2) e (3.11) são utilizadas.

3.4.3.3 Pólos complexos afastados da origem em relação aos pólos reais

Finalmente, na ocorrência da quinta situação, na Figura 3.11, se o pólo complexo encontrar-se suficientemente próximo ao segundo (e conseqüentemente do primeiro) pólo real. Em caso afirmativo, segue-se o procedimento de cálculo do sobressinal por meio de (3.2) e posteriormente (3.11). Do contrário, a porcentagem de sobressinal será considerada igual à zero.

Figura 3.11 – Caso III.c: diagrama de pólos de um sistema de quarta ordem com um par complexo conjugado próximo a origem.



Fonte: Autora.

3.5 RECÁLCULO DA NOVA MATRIZ DE CONTROLE

Depois da avaliação em malha fechada do novo sistema estimado, e caso seja verificada degradação de algum requisito de desempenho, o PSO deverá ser executado e a nova matriz de ganhos do controlador deverá então ser recalculada.

3.5.1 Ajuste dos parâmetros do algoritmo de obtenção da matriz de controle

Com relação à calibração dos parâmetros do algoritmo desenvolvido, foram executados inúmeros testes a partir da análise de diversas configurações de parâmetros para o PSO. Para a realização de tais testes, foram realizadas simulações de plantas distintas utilizando diferentes tamanhos de população, assim como diversos valores para os coeficientes de atração social e individual (c_1 e c_2) e, variação do peso de inércia w , de modo a se buscar definir o conjunto destes parâmetros capaz de fornecer convergência da forma mais eficiente possível, tanto no que diz respeito à velocidade de convergência quanto em relação às medidas qualitativas das soluções sugeridas.

No entanto, os conjuntos analisados forneciam resultados satisfatórios quando utilizados para obtenção da matriz de controle para alguns sistemas, mas ao serem aplicados para outros sistemas, nem sempre proporcionavam bons resultados (seja em relação à qualidade das soluções obtidas, seja em relação ao tempo de convergência), de modo que o algoritmo necessitava ser novamente calibrado.

Dessa forma, chegou-se ao bom desempenho do algoritmo para todos os sistemas testados a partir da aplicação do fator de constrição, sugerido por SHI e EBERHART (2000, p.1-5), o qual modifica a forma como a velocidade das partículas é atualizada a cada iteração do algoritmo.

Estudos experimentais demonstram que a aplicação desse tipo especial de tratamento é similar à utilização do fator de inércia na equação (2.4) do capítulo 2, aliada à boa calibração dos fatores de atração (BAI, 2010, p. 3). Neste tipo de abordagem, a equação de velocidade é reescrita como:

$$v_p(t+1) = \chi[v_p(t) + c_1.r_1(Pbest - x_p(t)) + c_2.r_2(Gbest - x_p(t))] \quad (3.12)$$

Sendo χ o coeficiente de constrição, calculado a partir da equação (3.13), o qual corresponde a um fator de amortecimento responsável por limitar em norma a velocidade da

partícula, evitando o efeito de “explosão” de sua velocidade ocasionando, por sua vez, aumento significativo dos valores correspondentes ao vetor posição acima dos limites de factibilidade para o projeto.

Consequentemente, este tipo de adaptação altera o algoritmo original de forma a aumentar a capacidade do PSO para encontrar pontos de ótimo para as funções de testes às quais foi aplicado (CLERC e KENNEDY, 2002, p. 1-16).

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad (3.13)$$

O fator de convergência φ é dado pela soma dos fatores cognitivo e social respectivamente, de modo que:

- $\varphi = c_1 + c_2$
- $\varphi > 4$

No algoritmo aqui desenvolvido, foram definidos parâmetros de modo a determinar um valor de $\chi = 0,729$.

Assim, o sistema é inicializado com uma população de soluções aleatórias. Em seguida é feita a atualização da partícula (a cada geração) com base na melhor posição encontrada por ela e na melhor posição encontrada analisando-se todas as partículas presentes no enxame. A tabela 3.1 expõe as características do algoritmo PSO utilizado:

Tabela 3.1 – Parâmetros de configuração do PSO desenvolvido.

Parâmetro	Descrição	Valor
<i>Dim</i>	Dimensão de cada partícula	8
<i>Nmax</i>	Número máximo de iterações	100
<i>NumParticulas</i>	Número de indivíduos no enxame	80
<i>X</i>	Fator de constrição	0.72
<i>c1</i>	Coefficiente de atração individual	2
<i>c2</i>	Coefficiente de atração social	2.1
<i>FatorConv</i>	Fator de convergência	4.1

Fonte: Autora.

3.5.2 Avaliação das soluções

Após a inicialização, o algoritmo entra em um laço que deverá ser executado até que um dos critérios de parada (requisitos de: estabilização, tempo de acomodação e máximo sobressinal percentual) definidos seja atingido.

Cada indivíduo do enxame no algoritmo PSO representa uma solução candidata e consiste de um vetor n -dimensional, em que n representa a dimensão espacial do problema, de modo que, a cada iteração, as partículas atualizam seus vetores velocidade e posição de acordo com os conjuntos que representaram melhores soluções em iterações anteriores, avaliados por meio de uma função de *fitness* para o problema.

O projeto de controle ótimo resume-se a um problema de otimização multiobjectivo, uma vez que consiste em um problema no qual existem múltiplos objetivos conflitantes a serem alcançados, de modo que, o aprimoramento de dado objetivo influencia a qualidade dos demais itens. Como resultado, um conjunto solução correspondente àquele capaz de maximizar os objetivos a serem alcançados para o presente trabalho, deve levar em consideração o somatório dos requisitos de desempenho impostos, tal que:

$$f(\textit{fitness}) = \textit{Nota_estabilização} + \frac{1}{T_s} + \frac{1}{Mp(\%)} \quad (3.14)$$

Sendo assim, para a avaliação de cada solução, somam-se as parcelas correspondentes às pontuações correspondentes ao alcance dos requisitos de estabilidade, menor tempo de acomodação e menor sobressinal.

Nesse caso, uma solução para o problema de controle proposto corresponde à posição da partícula, sendo cada elemento do vetor equivalente a um ganho de realimentação do controlador que maximize os objetivos a serem alcançados, conhecidos com base nos valores dos polos de malha fechada para cada solução candidata, a partir das matrizes A (estados) e B (entrada) do sistema.

3.5.3 Pseudocódigo do algoritmo PSO utilizado

O modo como se deu a adaptação do algoritmo ao problema de controle exposto, pode ser entendido através de um pseudocódigo do algoritmo PSO utilizado.

Figura 3.12 – Pseudocódigo do algoritmo PSO utilizado.

Pseudocódigo: Algoritmo PSO proposto

```

1  Iniciar as partículas do enxame com posição e velocidade aleatórias;
2  Enquanto critério de parada não for alcançado faça
3    Para cada partícula faça
4      Atualize a velocidade da partícula;
5      Atualize a posição da partícula;
6      Para cada elemento do vetor posição
7        Enquanto o elemento do vetor > LimiteSuperior
8          Faça elemento do vetor = elemento do vetor - rand[0-1];
9        Enquanto o elemento do vetor < LimiteInferior
10         Faça elemento do vetor = elemento do vetor + rand[0-1];
11     Faça  $AmenosBK = A - B * (posição\ da\ partícula)$ 
12     Avalie AmenosBK
13     Atualize a nota da partícula com base na função de fitness
14     Se nova posição é melhor do que a melhor solução encontrada pelo enxame então
15       Atualize Gbest e maior nota conseguida pelo enxame;
16     Senão
17       Se nova posição é melhor do que a memória cognitiva então
18         Atualize Lbest e memória cognitiva;
19  Retorne a melhor memória cognitiva do enxame.

```

Fonte: Autora.

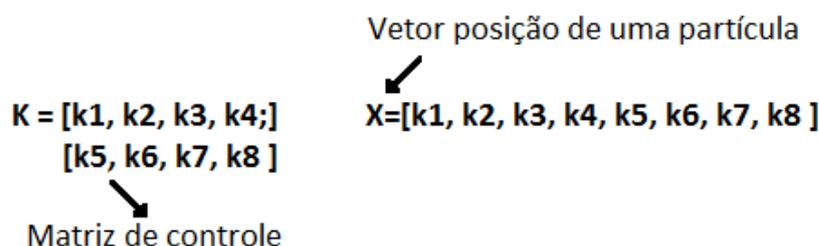
3.5.4 Elementos do algoritmo

- *Posição da partícula* → Vetor posição de cada partícula no espaço de soluções, constituindo uma solução candidata. Possui dimensão 1x8;
- *Memória cognitiva da partícula* → Maior nota recebida pela partícula;
- *Gbest* → Melhor solução encontrada pelo enxame;
- *Lbest* → Melhor solução obtida por cada partícula;
- *A* → Matriz de estados do sistema em malha aberta;
- *B* → Matriz de entrada do sistema;
- *AmenosBK* → Matriz de estados correspondente para o sistema em malha fechada a partir da solução obtida por cada partícula;
- *Nota da partícula* → Nota recebida pela iésima-partícula após a avaliação da matriz *AmenosBK*;

- **Elemento do vetor posição** → cada um dos oito elementos do vetor posição de cada partícula;
- **LimiteSuperior** → Limite superior do espaço de soluções. Constitui o maior valor de ganho admissível na matriz de ganhos do controlador;
- **LimiteInferior** → Limite inferior do espaço de soluções. Constitui o menor valor de ganho admissível na matriz de ganhos do controlador.

Os sistemas de quarta ordem testados apresentam duas entradas e duas saídas, sendo assim, suas respectivas matrizes de ganhos de realimentação possuem ordem 2x4. Porém, objetivando-se a simplificação, são gerados vetores velocidades e vetores posição correspondente a uma solução candidata para a matriz do controlador com dimensões 1x8 nos quais a quinta posição corresponde ao elemento da segunda linha e primeira coluna da matriz de controle K , tal que:

Figura 3.13 – Posicionamento dos ganhos de realimentação da matriz de controle adaptados ao vetor posição em uma partícula.



Fonte: Autora.

O espaço de soluções é limitado em valores pré-estabelecidos. Sendo assim, os ganhos máximos conferidos ao controlador deverão situar-se dentro de um intervalo de ganhos entre os limites inferior e superior, previamente estabelecidos pelo projetista.

Caso após a atualização do valor de uma partícula, algum elemento ultrapasse os limites admissíveis para a matriz de ganhos do controlador, faz-se esse valor ser acrescido com valores aleatórios até estarem posicionados dentro dos extremos pretendidos.

3.6 CONCLUSÕES DO CAPÍTULO

Este capítulo teve como objetivo instruir o leitor em relação aos métodos e abordagens concernentes à aplicação da estratégia de controle proposta.

Para isso, foi feita uma breve explanação acerca dos passos necessários ao processo de avaliação, estimação de parâmetros, avaliação e cálculo do controlador.

No que se refere à obtenção dos parâmetros do novo controlador, buscou-se definir um algoritmo que sugerisse uma nova matriz de controle responsável por proporcionar estabilidade e desempenho ao sistema em tempo hábil. Assim, foi desenvolvido um algoritmo PSO (APENDICE B) para o qual foram testados diferentes arranjos relacionados à especificação de parâmetros de relevância entre busca global e atração individual, utilização de diferentes pesos de inércia (variáveis ou não) e posteriormente a utilização do fator de constrição.

Os resultados dos testes para ajuste dos parâmetros do algoritmo demonstraram bom desempenho da utilização do fator de constrição em todos os sistemas testados, sendo esta a razão pela qual foi eleita como critério de ajuste de parâmetros para o algoritmo aqui desenvolvido, objetivando sua aplicação a diferentes sistemas de controle sem a necessidade de grandes ajustes.

4 RESULTADOS E DISCUSSÕES

A fim de se validar a solução proposta, foram testados trinta sistemas MIMO de quarta ordem apresentando duas entradas e duas saídas. Buscou-se eleger cada sistema a ser experimentado de modo que diferenciem-se dos demais em relação às características em malha aberta, tais como: sensibilidade, instabilidade, modos oscilantes, zeros lentos, presença ou não de zeros de transmissão, existência de pólos na origem, e outras peculiaridades.

Tendo em vista o caráter divergente entre estes sistemas em relação às dinâmicas apresentadas, o algoritmo PSO sofreu pequenas alterações no tocante aos limites extremos (máximos e mínimos) admissíveis para os ganhos da matriz K de realimentação, assim como às especificações de desempenho pretendidas.

Dessa forma, optou-se pela definição de índices que representassem considerável dificuldade à obtenção por parte do algoritmo de busca, como também pela demarcação de um *range* compatível com o problema proposto, de modo a proporcionar condições de factibilidade para o projeto de controle.

A fim de se testar os limites de eficácia da estratégia proposta, cada um dos trinta sistemas foi submetido a condições de variações em todos os elementos da sua respectiva matriz de estados, tanto em relação às margens de variação possíveis quanto em relação à probabilidade destas ocorrerem a cada amostra de tempo equivalente a 0,01 segundos.

Para isto foram impostos cinco diferentes limites de variações possíveis, aqui denotados como Lim_Var ($\pm 1\%$, $\pm 2\%$, $\pm 5\%$, $\pm 10\%$ e $\pm 20\%$). As probabilidades de ocorrência de tais variações em cada instante de tempo ($Prob_Var$) foram de: $\pm 1\%$, $\pm 5\%$, $\pm 10\%$, $\pm 20\%$ e $\pm 40\%$.

Em outras palavras, um sistema que esteja sujeito a mudanças nos parâmetros do modelo em torno de 1% será testado de modo que: a cada iteração haverá um por cento de probabilidade de todos os elementos da matriz de estados A sofrerem variações aleatórias em torno de um por cento de seu valor imediatamente anterior, em seguida essa configuração ($Lim_Var = 1\%$, $Prob_Var = 1\%$) é simulada uma quantidade n de vezes, e em cada simulação são recolhidas informações relevantes à análise pretendida. Posteriormente, para o mesmo sistema e mesmo $Lim_Var = 1\%$, aumenta-se a probabilidade desta ocorrer em cada intervalo de tempo para $Prob_Var = 2\%$, procede-se com as n simulações necessárias, e assim por diante.

Convém que o número n , equivalente à quantidade de amostras necessárias para cada configuração, seja suficientemente amplo de modo a possibilitar o preciso conhecimento das variáveis a serem pesquisadas.

Idealmente, para se ter conhecimento preciso destas variáveis, n tenderia a infinito, o que corresponderia a todos os indivíduos dentro da população de amostras possível. No entanto, nem sempre é possível obter informações a respeito de todos os elementos da população, e as vantagens das técnicas de estatísticas de inferência, nesses casos, justificam o uso de planos amostrais (MARTINS, 2002, p. 184 - 186).

Sendo assim, foi utilizada a fórmula (4.1) para definição do tamanho mínimo de n necessário para se ter uma quantidade de amostras significativas, de modo que se possa representar o melhor possível toda a população, tal que:

$$n = \left(\frac{Z^* \sigma}{d} \right)^2 \quad (4.1)$$

Sendo:

- $Z \rightarrow$ A abscissa da distribuição normal padrão, fixada para um nível de 95% de confiabilidade estatística, a qual corresponde, na tabela de distribuição normal padrão a um valor $Z = 1,96$ (MARTINS, 2002, p. 363).
- $\sigma \rightarrow$ Desvio padrão da população, expresso na unidade da variável;
- $d \rightarrow$ Erro amostral, também expresso na unidade da variável, o qual equivale à maior *diferença* que se admite suportar entre μ e M , tal que:
 - $|\mu - M| \leq d$;
 - $\mu \rightarrow$ é a verdadeira média populacional;
 - $M \rightarrow$ é a média a ser calculada com base na amostra;

Inicialmente, foram realizados testes fazendo uso de uma amostra de 200 simulações para cada configuração. No entanto, a partir da aplicação da fórmula em 4.1, foi constatado que o menor tamanho de n que fornece um $d = 0,001$ (uma vez que foi verificado que a média de tempos calculada era em torno de milissegundos) deve ser igual a pelo menos 237 simulações para cada uma das 25 configurações utilizadas que fornece a confiabilidade necessária para construção do intervalo de confiança em um nível de 95%.

Sendo assim, levando-se em consideração a quantidade de plantas testadas, o número de configurações para cada sistema relacionadas aos limites impostos, probabilidade de tais

variações ocorrerem, como também a definição mínima de 237 simulações necessárias para cada uma das vinte e cinco configurações possíveis, chega-se ao total de testes descritos por meio da tabela 4.1:

Tabela 4.1 – Tabela de especificação de testes.

Testes realizados	
Quantidade de sistemas testados	15
Quantidade de configurações possíveis	25
Quantidade de testes por configuração	237
Total de testes realizados	88.875

Fonte: Autora.

Deste modo, a partir de cada uma das vinte e cinco configurações possíveis para cada sistema, buscou-se definir uma taxa de amostragem pretendida para sistemas genéricos sujeitos às situações aqui avaliadas.

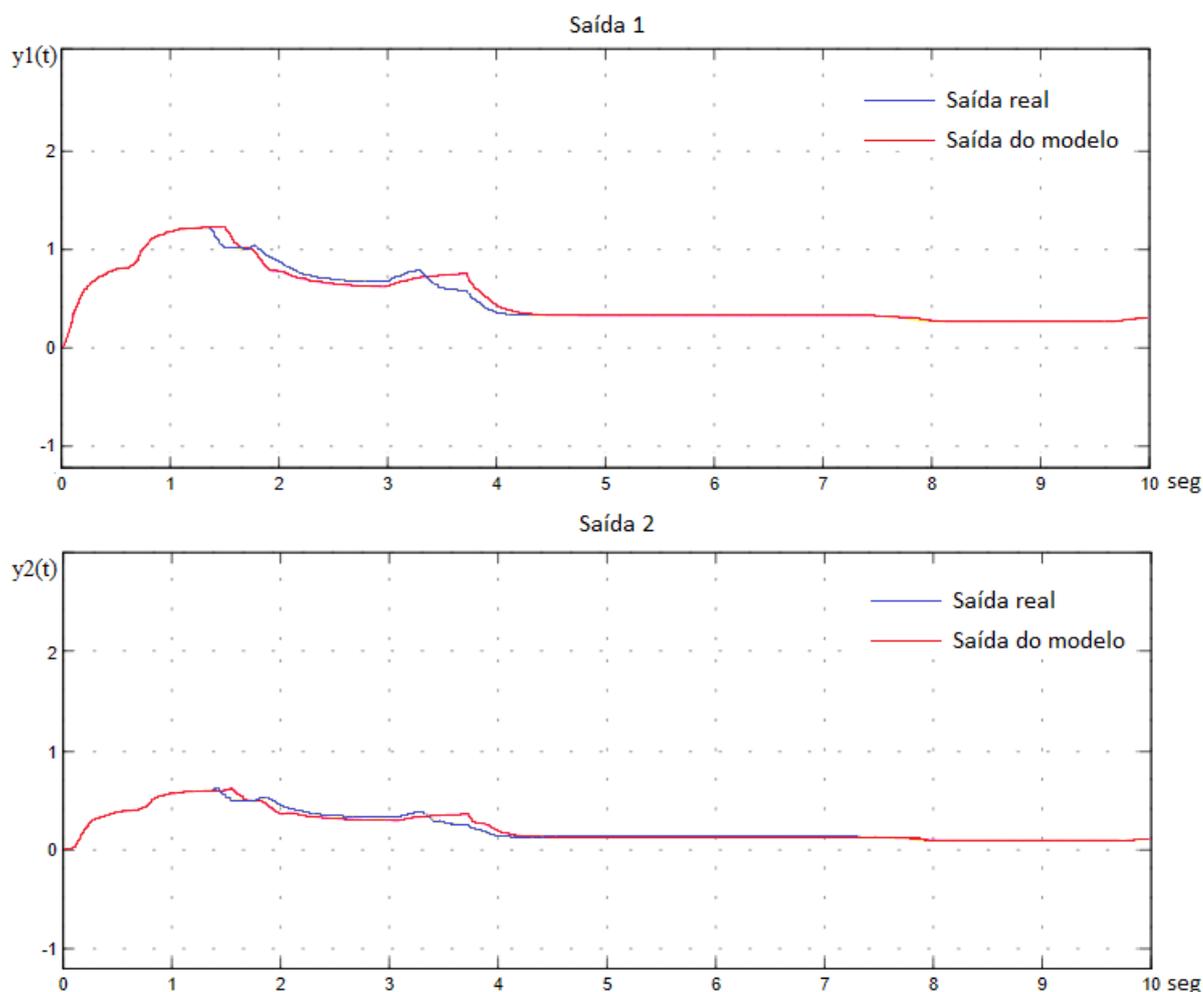
A cada simulação realizada, buscou-se medir as informações referentes aos tempos requeridos para se obter uma solução que atenda aos requisitos impostos ao algoritmo PSO no que concerne à obtenção de uma solução considerada satisfatória.

Posteriormente, mediu-se o tempo total transcorrido ao levar-se em conta o método de avaliação proposto e descrito no capítulo 3, assim como aquele consumido apenas pelo PSO.

A escolha de se levar ou não em consideração o tempo gasto com a avaliação deve-se ao fato de que o tempo demandado para se proceder com a verificação de cada solução candidata depende não somente da quantidade, mas também do tipo de índice de desempenho pretendido. Além do mais, o critério de avaliação pode ser realizado por meio de um DPS (*Digital Signal Processor*) dedicado a esse propósito.

4.1 RESULTADOS

As Figuras 4.1.a e 4.1.b exemplificam as respostas ao degrau unitário referentes às saídas 1 e 2 ao longo de 10 segundos de simulação para a estratégia de controle adaptativo proposta aplicada ao sistema de número 1 (escolhido aleatoriamente dentre os 15 sistemas testados) do APENDICE B.

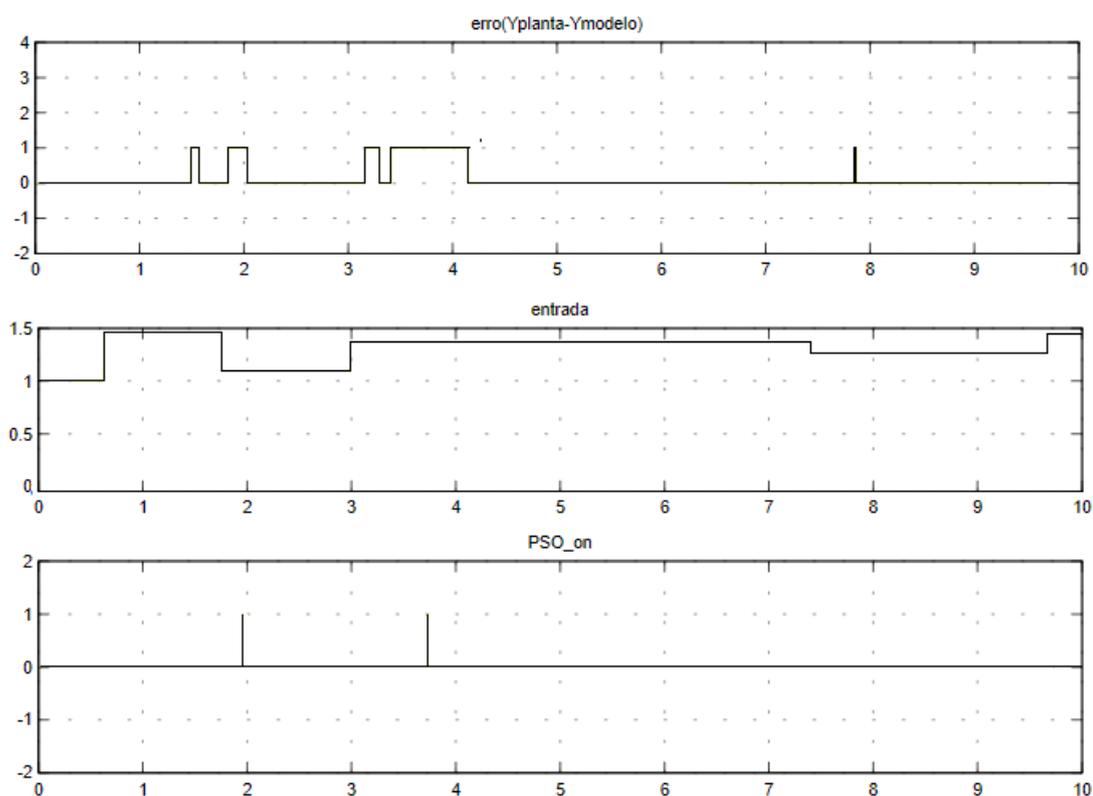
Figura 4.1.a – Saídas do sistema real e saídas do modelo estimado.

Fonte: Autora.

Observa-se na Figura 4.1.a que o modelo estimado para a matriz de estados do sistema fornece saídas aproximadas àquelas apresentadas pela planta. Essa característica foi também verificada para os demais sistemas testados.

Para o mesmo sistema, a partir da Figura 4.1.b, é possível verificar que o erro foi detectado entre as saídas da planta e do modelo cinco vezes ao longo dos dez segundos de simulação, ao passo que o algoritmo PSO foi acionado apenas duas vezes, em aproximadamente quatro segundos de simulação.

Figura 4.1.b – Gráficos de desempenho do sistema de controle adaptativo.



Fonte: Autora.

O sinal de entrada foi variado de maneira aleatória e com amplitudes também aleatórias, dentro de um limite que pode variar entre 0 e 80% do valor inicial definindo uma entrada do tipo degrau unitário. Apesar da presença de tais perturbações nas entradas, verifica-se que a partir da metodologia proposta o sistema em malha fechada apresenta comportamento dinâmico de modo a manter-se insensível no que diz respeito a variações consideráveis nos sinais de entrada.

4.2 ANÁLISE DOS RESULTADOS

Para fins de análise dos tempos demandados pelo algoritmo PSO, foi necessária a construção de um conjunto de dados relacionados aos períodos durante os quais este permaneceu ativo na busca por uma solução.

Sendo assim, primeiramente definiu-se um grupo de tabelas correspondentes aos valores médios obtidos a partir das medições de todos os sistemas medidos ao longo de um tempo de simulação equivalente a 10 segundos, fazendo uso da ferramenta *Simulink*[®]/*MATLAB*[®]

7.10.0 (versão R2010a), em um computador com processador Intel® Core™ i5-4200U 1,6 GHz, memória 6 GB DDR3 .

Cada segundo foi particionado em intervalos de 10 (dez) milissegundos, perfazendo um total de 1001 amostras ao longo do período de uma simulação.

Todos os sistemas testados encontram-se no APENDICE B.

4.2.1 Média de acionamentos

As tabelas 4.2 a 4.6 mostram como se deu a média de variações ocorridas na matriz de estados dos sistemas testados ao longo de 10 segundos de simulação, assim como a média de acionamentos do PSO, denotada como N_{ac} .

Tabela 4.2 – média de ocorrência de variações do sistema ao longo de 10s e média de acionamentos do PSO para uma probabilidade $Prob_Var$ de 1%.

$Prob_Var$	Lim_Var	Média de Variações em 10s	Média de Acionamentos do PSO (N_{ac})
1%	1%	5	1,0667
1%	2%	5	1,2
1%	5%	5	1,4
1%	10%	5	1,6
1%	20%	5	1,8667

Fonte: Autora.

Tabela 4.3 – média de ocorrência de variações do sistema ao longo de 10s e média de acionamentos do PSO para uma probabilidade $Prob_Var$ de 5%.

$Prob_Var$	Lim_Var	Média de Variacões em 10s	Média de Acionamentos do PSO (N_{ac})
5%	1%	26	2,6667
5%	2%	26,13333333	3
5%	5%	26,26666667	4,8
5%	10%	25,93333333	7,2667
5%	20%	26,26666667	11,667

Fonte: Autora.

Tabela 4.4 – média de ocorrência de variações do sistema ao longo de 10s e média de acionamentos do PSO para uma probabilidade *Prob_Var* de 10%.

<i>Prob_Var</i>	<i>Lim_Var</i>	Média de Variações em 10s	Média de Acionamentos do PSO (Nac)
10%	1%	78,86666667	3,2667
10%	2%	78,66666667	4,0667
10%	5%	79,4	8,3333
10%	10%	78,66666667	15,733
10%	20%	78,8	36,067

Fonte: Autora.

Tabela 4.5 – média de ocorrência de variações do sistema ao longo de 10s e média de acionamentos do PSO para uma probabilidade *Prob_Var* de 20%.

<i>Prob_Var</i>	<i>Lim_Var</i>	Média de Variações em 10s	Média de Acionamentos do PSO (Nac)
20%	1%	167	3,3333
20%	2%	166,33333333	4,2
20%	5%	167,33333333	10,8
20%	10%	166,66666667	25,2
20%	20%	167,33333333	71,733

Fonte: Autora.

Tabela 4.6 – de média de ocorrência de variações do sistema ao longo de 10s e média de acionamentos do PSO para uma probabilidade *Prob_Var* de 40%.

<i>Prob_Var</i>	<i>Lim_Var</i>	Média de Variações em 10s	Média de Acionamentos do PSO (Nac)
40%	1%	375,33333333	3,3333
40%	2%	375,73333333	4,6
40%	5%	375,4	14,067
40%	10%	375,46666667	44,733
40%	20%	375,26666667	155,1

Fonte: Autora.

Nota-se que limites de variações em torno de 1% implicam em uma média de acionamentos do algoritmo PSO relativamente baixa. Na verdade para alguns sistemas testados esse tipo de variação não implica em uma mudança significativa da dinâmica do sistema. Isto significa que na maioria dos casos, para limites baixos de variações nos

parâmetros do sistema, mesmo nas situações em que se impõe uma *Prob_Var* de 40%, correspondendo a uma média de pouco mais de 375 variações a cada 10 segundos, estes mantém dinâmica similar de modo a manter os índices de desempenho dentro das especificações do projetista.

Logicamente, esta é uma característica inerente à sensibilidade do processo, a partir da qual se mede a capacidade das saídas do sistema serem afetadas por pequenas flutuações paramétricas.

4.2.2 Tempo médio e taxa de amostragem

Nas tabelas 4.7 a 4.11 constam os valores médios de tempo nas quais o algoritmo PSO manteve-se atuante na busca por uma solução, ao longo de 10 segundos de simulação, desconsiderando-se o tempo consumido para se fazer a avaliação de estabilidade e desempenho.

A partir dos resultados verifica-se que, embora os tempos médios demandados para a obtenção de uma nova solução seja maior quando se impõe um limite maior para variação dos elementos da matriz *Lim_Var*, ao se levar em consideração o fato de que o algoritmo de busca é solicitado em uma quantidade de vezes maior nessas situações, nota-se que o tempo médio despendido para se chegar a uma matriz de realimentação satisfatória está contido dentro de um intervalo de tempo em torno de aproximadamente 5 milissegundos (0,004833 segundos).

Estes resultados devem-se ao conhecimento de que, uma vez que ao se implantar uma variação em torno de 1% nas matrizes, o PSO necessitará realizar apenas ajustes na velocidade das partículas, uma vez que a solução deverá encontrar-se na vizinhança. Ao passo que variações em torno de 20% representam uma mudança brusca do sistema de controle e geralmente a nova matriz de realimentação deve possuir características totalmente divergentes daquelas encontradas na vizinhança.

Sendo assim, para inicialização dos vetores velocidade e posição das partículas, metade destas recebe valores totalmente novos, gerados randomicamente dentro do *range* limite, de modo a estabelecer diversidade à população. A metade remanescente terá seus valores atualizados a partir dos valores de *Gbest* correspondendo ao valor vigente para a matriz de controle, a fim de se favorecer a busca local no caso de o sistema manter características aproximadas em relação ao modelo anterior, de modo que a solução será facilmente encontrada na vizinhança.

Ao se levar em consideração o número de acionamentos ocorridos tem-se a medida de tempo médio efetivo do algoritmo de busca.

Essas informações são necessárias quando se deseja analisar a relação entre a taxa de amostragem necessária para o processo e aquela demandada por controladores a serem aplicados a sistemas de controle sujeitos às faixas de variação analisadas.

Em outras palavras, ao se fazer a especificação dos elementos pertencentes à malha de controle, torna-se necessária à observação das dinâmicas envolvidas no processo, de modo que, o controlador digital possua taxa de amostragem adequada para manter um tempo de resposta suficiente para sistemas com dinâmicas rápidas.

Isto se deve ao fato de que, como o sistema digital executa as funções pertinentes ao controlador a partir de um algoritmo de controle, este por sua vez requer certo tempo de processamento necessário à efetivação de todas as tarefas a ele requisitadas, desde leitura das entradas até modificação do sinal de controle.

Assim, para que o controlador realize a aferição das informações provenientes dos elementos sensores, é feita a conversão destas grandezas de natureza analógica em variáveis digitalizadas a partir de uma taxa de amostragem, tal qual o sistema de controle somente tenha acesso a esses valores em intervalos de tempo equivalentes ao período predefinido.

No entanto, estes intervalos caracterizam o que se chama de “tempo morto”, durante o qual se houver a oscilação de uma variável controlada, este impedirá que as medidas de controle sejam tomadas imediatamente no sentido de evitar tais oscilações, sendo a variável controlada atualizada somente no início do intervalo seguinte. Com isso, observa-se que quanto maior a frequência de amostragem, mais rapidamente o sinal de controle poderá ser atualizado, aumentando-se, desta forma, sua eficiência.

Por essa razão, busca-se definir uma relação existente entre o tempo de amostragem do sistema de controle e as características dinâmicas do processo ao qual será implantado. E levando-se em conta os dados referentes aos tempos médios demandados para o PSO, sem levar em consideração o tempo necessário para a avaliação, verifica-se que ao longo dos dez segundos de duração da simulação, o sistema de controle gasta em média 5 milissegundos por intervalo de amostragem para se chegar a uma solução.

De acordo com o exposto, o algoritmo necessita que o sistema amostrado possua um intervalo de amostragem igual ou superior a 5 milissegundos, equivalente à frequência de amostragem obtida por meio da expressão em (4.2):

$$f = \frac{1}{5 * 10^{-3} \text{segundos}} \approx 200 \text{Hz} \quad (4.2)$$

Ou seja, o algoritmo de busca sugerido terá um desempenho satisfatório para sistemas que demandem uma frequência de amostragem de cerca de até 200 Hertz.

A tabela contendo os valores de tempo medidos, levando-se em conta o método para se proceder com a análise aqui utilizada encontra-se exposta no APENDICE C da presente dissertação.

4.2.3 Medidas de variância

Outra análise pretendida é relacionada às medidas de variância, contidas nas tabelas 4.7 a 4.11, a fim de se averiguar os desvios padrões observados nos tempos gastos pelo PSO, de modo a se ter informação a respeito da quantidade de pontos de dispersão fora do conjunto que apresenta comportamento similar à maioria, de outro modo, busca-se saber se existem grandes divergências de tempo entre os resultados obtidos, tal que o algoritmo proposto encontre frequentemente soluções em volumes de tempo fora da média de tempos padrão.

Medidas quantitativamente baixas de variância implicam que os tempos demandados para a obtenção de uma boa solução possuem valores próximos da média, de modo a indicar que o algoritmo proposto apresenta um desempenho significativamente regular no tocante à média de tempos necessários à obtenção de uma boa solução.

Tabela 4.7 – medidas de variância e tempos médios demandados pelo PSO para encontrar uma solução para uma probabilidade de variação *Prob_Var* de 1%.

<i>Prob_Var</i>	<i>Lim_Var</i>	Tempo médio gasto pelo PSO/Nac em 10s	Variância dos tempos do PSO/Nac
1%	1%	0,002773	6,1E-06
1%	2%	0,002417	4,67E-05
1%	5%	0,002489	0,000254
1%	10%	0,002509	0,000165
1%	20%	0,002919	0,000281

Fonte: Autora.

Tabela 4.8 – medidas de variância e tempos médios demandados pelo PSO para encontrar uma solução para uma probabilidade de variação *Prob_Var* de 5%.

<i>Prob_Var</i>	<i>Lim_Var</i>	Tempo médio gasto pelo PSO/Nac em 10s	Variância dos tempos do PSO/Nac
5%	1%	0,002722	6,95E-06
5%	2%	0,002401	0,000139
5%	5%	0,004544	0,000236
5%	10%	0,004531	0,000181
5%	20%	0,005301	0,000178

Fonte: Autora.

Tabela 4.9 – medidas de variância e tempos médios demandados pelo PSO para encontrar uma solução para uma probabilidade de variação *Prob_Var* de 10%.

<i>Prob_Var</i>	<i>Lim_Var</i>	Tempo médio gasto pelo PSO/Nac em 10s	Variância dos tempos do PSO/Nac
10%	1%	0,002737	7,79E-06
10%	2%	0,00385	0,000311
10%	5%	0,005056	0,000161
10%	10%	0,004914	0,000184
10%	20%	0,006149	0,000123

Fonte: Autora.

Tabela 4.10 – medidas de variância e tempos médios demandados pelo PSO para encontrar uma solução para uma probabilidade de variação *Prob_Var* de 20%.

<i>Prob_Var</i>	<i>Lim_Var</i>	Tempo médio gasto pelo PSO/Nac em 10s	Variância dos tempos do PSO/Nac
20%	1%	0,002565	1,42E-05
20%	2%	0,00423	0,000247
20%	5%	0,004833	0,000155
20%	10%	0,004973	0,000138
20%	20%	0,006652	8,48E-05

Fonte: Autora.

Tabela 4.11 – medidas de variância e tempos médios demandados pelo PSO para encontrar uma solução para uma probabilidade de variação *Prob_Var* de 40%.

<i>Prob_Var</i>	<i>Lim_Var</i>	Tempo médio gasto pelo PSO/Nac em 10s	Variância dos tempos do PSO/Nac
40%	1%	0,002651	4,32E-05
40%	2%	0,004217	0,000405
40%	5%	0,005092	0,000175
40%	10%	0,004994	8,63E-05
40%	20%	0,006399	0,000162

Fonte: Autora.

Desta forma, a partir dos dados das tabelas 4.7 a 4.11, pode-se afirmar que os baixos valores de variância obtidos indicam o desempenho regular do algoritmo proposto em relação ao tempo médio de obtenção de uma matriz correspondente a uma solução satisfatória a determinado problema de controle proposto.

4.3 CONCLUSÕES DO CAPÍTULO

Este capítulo foi reservado à análise dos resultados obtidos para a realização do presente trabalho.

Mediante a aplicabilidade do algoritmo proposto no tocante às diversas situações práticas exequíveis e, a fim de se atestar a eficácia do método proposto em relação a sua funcionalidade, foram testados 15 diferentes sistemas, a partir de vinte e cinco configurações possíveis dentro dos limites de variações analisados, assim como a quantidade de duzentos e trinta e sete testes para cada configuração, atingindo-se um total de oitenta e oito mil oitocentos e setenta e cinco (88.875) testes realizados.

A partir dos dados coletados, foram elaboradas tabelas com o intuito de determinar a média de tempo necessária à obtenção de uma solução, assim como se existem grandes variações nesses valores, de modo a analisar a regularidade do algoritmo proposto em relação ao período necessário para a aquisição de um controlador capaz de proporcionar cumprimento dos requisitos de desempenho para o sistema em malha fechada.

Também se definiu uma frequência limite de amostragem para a performance satisfatória do algoritmo aqui proposto.

5 CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

Este trabalho propõe uma alternativa de controle adaptativo a ser aplicado a sistemas lineares variantes no tempo ou que apresentem incertezas paramétricas.

A estratégia de controle aqui proposta buscou um meio de se obter rapidamente o controle ótimo que proporcione ao sistema atingir não somente estabilidade, mas também desempenho ao sistema de controle.

Devido às características de rápida convergência observada a partir da aplicação do PSO, o trabalho desenvolvido para esta dissertação teve como objetivo sugerir uma estratégia de controle adaptativo por meio da utilização de um algoritmo baseado neste tipo de metaheurística adaptado à busca de uma matriz de controle via realimentação de estados a ser aplicada a sistemas multivariáveis lineares de quarta ordem com incertezas nos parâmetros da matriz de estados do modelo.

O meio utilizado para se realizar a estimação da nova matriz de estados A para o modelo vigente foi fundamentado em regressão linear simples. Para fins de análise de desempenho, foi utilizada uma avaliação dos pólos do sistema em malha fechada. O método de avaliação deve ficar a critério do projetista, uma vez que este pode ser realizado em *hardware* dedicado a este propósito.

Para a obtenção da matriz de controle de forma rápida, chegou-se ao algoritmo desenvolvido cujos parâmetros foram calibrados a partir da inserção de um fator de constrição. O fator de constrição foi eleito como método de ajuste de parâmetros para o algoritmo uma vez que foi capaz de fornecer sempre boas soluções em um número superior de aplicações às quais foi implantado.

Com relação aos resultados obtidos, foram testados cerca de quinze sistemas com características distintas, onde se buscou emular o caráter mutável de diversos sistemas, para os quais se pôde enfim definir uma taxa de amostragem máxima designada para o sistema, a partir da qual se verificou que o algoritmo proposto garante um desempenho satisfatório nas situações onde o processo necessite de uma taxa máxima de cerca de aproximadamente 200 Hz. Em frequências superiores a esta, não se assegura a total ausência de perdas apreciáveis do sistema de controle digital, em comparação a um controlador contínuo.

Os resultados demonstraram, também, que o algoritmo de busca apresenta comportamento satisfatório com relação aos tempos médios de aquisição de uma solução, de modo a tornar o sistema de controle adaptativo proposto apto no que se refere à tomada de decisão, estimação de modelo e obtenção em tempo real dos parâmetros da matriz de

realimentação de estados para sistemas multivariáveis sujeitos a incertezas paramétricas ou variantes no tempo.

Como sugestão de trabalhos futuros aponta-se a realização de testes envolvendo sistemas de ordem inferiores (segunda e terceira ordens) e quantidades diversas de entradas e saídas de interesse, tal que se busque analisar o desempenho do sistema de controle proposto, assim como taxas máximas de amostragem para uma gama ainda maior de aplicações existentes, assim como a utilização e análise da metodologia aqui desenvolvida em plantas que representem um sistema físico real, necessitando índices de desempenho compatíveis com a aplicação pretendida.

6 REFERÊNCIAS BIBLIOGRÁFICAS

ACKERMANN, Jürgen; BLUE, Paul; BÜNTE, Tilman; GÜNVENC, Levent; KAESBAUER, Dieter; KORDT, Michael; MUHLER, Michael; ODENTHAL, Dirk. *RobustControl: The Parameter Space Approach*. 2nd Edition. Springer-Verlag, London. 2002.

ADAMI, Alberto. **Resolução de Problemas de Controle ótimo Utilizando o Algoritmo Box**. Dissertação de mestrado. Campinas, SP. 1996.

AGUIRRE, Luiz Antônio. **Introdução à Identificação de Sistemas– Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais**. 3^a Ed. Editora UFMG. 2007.

AFROSI, M. Nabipour; NAEBI, Ahmad; AGHDAM, M. Hassanpour, AGHDAM, S. Hassanpour. *Simulation and Optimization of asynchronous AC Motor Control by Particle Swarm Optimization (PSO) and Emperor Algorithm*. 5th European Symposium on Computer Modeling and Simulation, 2011.

ALBERTOS, P.; SALA, Antonio. *Multivariable Control Systems: Na Engineering Approach*. Springer. 2003.

ARAÚJO, Fábio Meneghetti Ugulino de. **Sistemas de Controle**. Apontamentos. UFRN. Departamento de Engenharia de Computação e Automação. Natal-RN. Fevereiro de 2007. 101p.

ARBOLEDA, Daniel Mauricio Munoz. **Otimização por Inteligência de Enxames Usando Arquiteturas Paralelas para Aplicações Embarcadas**. Tese de doutorado. Brasília, DF. 2012. 192p.

ASTRÖM, Karl J; WITTENMARK, Björn. *Adaptive Control*. 2nd Edition. Dover Publications. 2008.

ÁVILA, Sérgio Luciano. **Algoritmos Genéticos Aplicados na Otimização de Antenas Refletores**. Dissertação de Mestrado. Santa Catarina. RS. 2002. 85p.

BAI, Qinghai. *Analysis of Particle Swarm Optimization Algorithm*. College of Computer Science and Technology, Inner Mongolia University for Nationalities. China. Vol. 3, No 1. 2010.

BAKIRTZIS, Anastasios G.; BISKAS, Pandel N.; ZOUMAS, Christoforos E.; PETRIDIS, Vasilios. *Optimal Power Flow by Enhanced Genetic Algorithm*. *IEEE Transactions On Power Systems*, Vol. 17, NO. 2. 2002.

BARÃO, Renato Casali. **Algoritmos De Controle Ótimo Quadrático Com Restrições**. Dissertação de mestrado. São Paulo, SP. 1997. 81p.

BAZANELA, A.; SILVA JR., J. **Sistemas de Controle, Princípios e Métodos de Projeto**. Porto Alegre: UFRGS, 2005.

BELARBI, Khaled; BELHANI, Ahmed; FUJIMOTO, Kuniak. *Multivariable Fuzzy Logic Controller Based On A Compensator Of Interactions And Genetic Tuning*. *International Journal of Innovative Computing, Information and Control ICIC*. Vol 2. 2006.

BERTACHI, Arthur Hirata; BIAGI SILVA, Lyvia Regina; SUMAR, Rodrigo Rodrigues; ANGÉLICO, Bruno Augusto; GOEDEL, Alessandro. **Controle de um Processo Multivariável em uma Planta Didática Industrial Utilizando Redes Neurais**. XI Simpósio Brasileiro de Automação Inteligente – SBAI. Fortaleza, CE, 2013.

BIRLA, N.; SWARUP, A. *PSO approach to preview tracking control systems*. In *TENCON 2009-2009 IEEE Region 10 Conference*. Singapore. 2009.

BJORKKOY, Olav. *Competitive and Cooperative Behavior in Bio-Inspired AI*. Department of Computer and Information Science NTNU. January, 2011.

BRANDINI, Pedro Aleixo Ferreira. **Metaheurística Particle Swarm Utilizada para Alocação Ótima de Bancos de Capacitores em Sistemas de Distribuição Radial**. Dissertação de mestrado, Ilha Solteira, SP, 2007.

CAVAZOS, John; MOSS, J. Eliot B.; O'BOYLE, Michael F. P. *Hybrid Optimizations: Which Optimization Algorithm to Use?*. Edinburgh, UK, in: <http://www.anc.ed.ac.uk/machine-learning/colo/hybrid.pdf>. s.d, acesso em 15/12/2014.

CHENG, Ze; ZHOU, Hang; YANG, Hongzhi. *Research On MPPT Control Of Pv System Based On PSO Algorithm*. In *Control and Decision Conference (CCDC)*. 2010.

CLERC, M.; KENNEDY, J. *The Particle Swarm-explosion Stability, and Convergence in a Multidimensional Complex Space*. *IEEE Transactions on Evolutionary Computation*, 2002, 6(1):58-73.

COELHO, Leandro dos Santos; MARIANI, Viviana Cocco. **Sistema Híbrido Neuro-Evolutivo Aplicado ao Controle de um Processo Multivariável**. Revista Controle & Automação/Vol.17 no.1/Janeiro, Fevereiro e Março, 2006.

DA LUZ, André Ferreira. **Uma Metodologia Baseada em Algoritmo de Otimização por Enxame de Partículas para Manutenção Preventiva Focada em Confiabilidade e Custo**. Dissertação de mestrado. Rio de Janeiro, RJ, 2009. 66p.

DIAS, Samaerni Moraes. **Controle Adaptativo Robusto para um Modelo Desacoplado de um Robô Móvel**. Tese de doutorado. Natal, RN, 2010. 121p.

DIMEO, R., LEE, K. Y. *Boiler-Turbine Control System Design Using A Genetic Algorithm. Published in: Energy Conversion, IEEE Transactions*. Vol. 10. 1995.

DORF, R. C.; BISHOP, R. H., **Sistemas de Controle Modernos**. 11ª Edição. Editora LTC. 2009.

DORIGO, Marco; MANIEZZO, Vittorio; COLORNI, Alberto. *The Ant System: Optimization by a Colony of Cooperating Agents. Proc. of IEEE, Transactions on Systems, Man, and Cybernetics*, 1996.

EBERHART, Russell C.; SHI, Yuhui. *Comparison Between Genetic Algorithms and Particle Swarm Optimization. Evolutionary Programming VII, Lecture Notes in Computer Science Volume 1447*, pp 611-616, 1998.

EBERHART, Russel C.; SHI, Yuhui. *Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization*. In: congress on evolutionary computation, 2000, New York, USA. *Proceedings*. New York, 2000.p.84-88.

EBERHART, Russell C.; KENNEDY, James. *A New Optimizer Using Particle Swarm Theory*. in: *Proc. of the Sixth Int. Symp. on Micro Machine and Human Science (MHS'95)*, Nagoya, Japan (1995) 39–43.

ELLIS, George; *Control systems Design Guide: Using your computer to understand and diagnose feedback controllers*. 3th edition, Elsevier Academic Press, 2004.

FARIAS, Missilene S.; OLIVEIRA, Phelipe Sena; BARROS, Luciano Sales; SILVEIRA JÚNIOR, Luiz Gonzaga de Queiroz. **Projeto de Sistema de Controle Multivariável Baseado em Otimização por Colônia de Formigas**. V Simpósio Brasileiro de Sistemas

Elétricos –SBSE, Foz do Iguaçu, PR. 2014.

FIKAR, M., KOSTÚR, K. *Optimal Process Control*. Slovak University of Technology in Bratislava. 13th International Carpathian Control Conference (ICCC), 2012.

FONTES, Fábio Franciso da Costa. **Algoritmo Memético com Infecção Viral: Uma Aplicação ao Problema do Caixeiro Viajante Assimétrico**. Dissertação de mestrado, Natal, RN. 2006. 72p.

GEROMEL, J. C. **Controle Ótimo de Sistemas Dinâmicos**. Tese de livre docência. Campinas, SP, 1986. 121p.

GLOVER, Fred; KOCHENBERGER, Gary A. *Handbook Of Metaheuristics*. Kluwer Academic Publishers; New York, Boston, Dordrecht, London, Moscow, 2003.

GOLNARAGHI, Farid; KUO, Benjamim C. **Sistemas de Controle Automático**. 9ª Edição. Editora LTC. 2012.

GUIMARÃES, Ádller De Oliveira. **Algoritmo Genético Aplicado No Controle De Posição Do Rotor De Um Motor De Corrente Contínua Com Rejeição A Distúrbios Por Ação Feedforward**. Dissertação de Mestrado. Mossoró. RN. 2013. 72p.

HE, Jiajia; HAU, Zaien. *Adaptive ant colony algorithm and its application to parameters optimization of PID Controller*. In 3rd International Conference on Advanced Computer Control (ICACC 2011). 2011.

HSIAO, Ying-Tung.; CHUANG, Cheng-Long; CHIEN, Cheng-Chih. *Ant colony optimization for designing of PID controllers*. in *Proceedings of the IEEE International Symposium on Computer Aided Control Systems Design, Taipei, Taiwan, 2004*.

JALALI, Mohammad Reza; AFSHAR, Abbas; MARINO, Miguel A. *Ant Colony Optimization Algorithm (ACO): A new heuristic approach for engineering optimization*. *Proceedings of the 6th WSEAS Int. Conf. on evolutionary computing, Lisbon, Portugal, June 16-18, 2005* (pp188-192).

KALMAN, Rudolf E. *A new approach to linear filtering and prediction problems*. *Journal of Basic Engineering* 82 (1): 35–45. 1960.

KALMAN, Rudolf E.; BUCY, R. S. ***New Results in Linear Filtering and Prediction Theory.*** *Journal of Basic Engineering; Transactions of the AS ME.* 1961.

KARANKI, Srinivas Bhaskar; MISHRA, Mahesh K.; KUMAR, B. Kalyan. ***Particle Swarm Optimization-Based Feedback Controller for Unified Power-Quality Conditioner.*** *IEEE transactions on power delivery*, vol. 25, no. 4, 2010.

KENNEDY, James; EBERHART Russell. ***Particle swarm optimization.*** *Proc. IEEE International Conf. on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, 1995 (in press).*

LI, W. T.; XU, L.; SHI, X. W. ***A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Antenna Design.*** *Piers online*, vol.4, China, 2008.

LOPES, Elenice da Conceição Castro. ***Aplicação da Transformada Discreta Wavelet no Controle da Diversidade Populacional nos Algoritmos Genéticos em Problemas de Otimização.*** Dissertação de mestrado, São Paulo, SP, 2012. 70p.

LUENBERGER, David G. ***Observing the State of a Linear System.*** *IEEE Transactions on Military Electronics.* 74–80. 1964.

LUIZ, Cristiano C; SILVA, Allan C; COELHO, Antonio A. R.; BRUCIPAGLIA, August H. ***Controle adaptativo versus controle fuzzy: Um estudo de caso em um processo de nível.*** *SBA Contrate & Automação I*, Vol.8, Jun., Jul. e Agosto 1997.

MANDANI, Kurosh; CORREA, A. Dourado; ROSA, Agostinho; FILIPE, Joaquim. ***Computational Intelligence: Revised and Selected Papers of the International Joint conference.*** *Valencia, Spain*, 2010.

MARCHIORI, Gustavo Capovilla. ***Controle Robusto Multivariável Freqüencial com Modelo Interno Via Valor Singular Estruturado.*** Dissertação de mestrado, São Paulo; Universidade de Campinas – UNICAMP; 1994. 86p.

MARTINS, Alexandre Teixeira. ***Projeto de um Controlador PI-MIMO para uma Planta de Produção de Poliestrileno.*** Dissertação de mestrado, Porto Alegre, RS, 2009. 98p.

MARTINS, Gilberto de Andrade. ***Estatística Geral e Aplicada.*** 2ª Edição. Editora Atlas SA. 2002.

MAYA, P. A.; LEONARD, F. **Controle Essencial**. 1ª Edição, *Pearson Education* do Brasil, 2011.

MEDEIROS, José Antônio Carlos Canedo. **Enxame de Partículas como Ferramenta de Otimização em Problemas Complexos de Engenharia Nuclear**. Tese de doutorado, Rio de Janeiro, RJ. 2005. 108p.

MENDONÇA, Marcio. **Uma Contribuição ao Desenvolvimento de Sistemas Inteligentes utilizando Redes Cognitivas Dinâmicas**. Paraná; Tese de Doutorado; Universidade Tecnológica Federal do Paraná, 2011. 179p.

MOEDINGER, Luís Henrique. **Algoritmos Evolutivos e Inteligência Coletiva Aplicados a Problemas de Otimização Não-linear com Restrições: Fundamentos e Estudos Comparativos**. Dissertação de mestrado, Curitiba, PR. 2005. 74 p.

NETO, João Viana da Fonseca. **Alocação Computacional Inteligente de Autoestruturas para Controle Multivariável**. Tese de doutorado, Campinas, SP, 2000. 212p.

OGATA, K. **Engenharia de Controle Moderno**. *Prentice Hall*. 2003.

OLIVEIRA, Tiago Roux de. **Controle Por Modos Deslizantes de Sistemas Incertos Com Direção de Controle Desconhecida**. Dissertação de mestrado; Rio de Janeiro, RJ, 2006. 196p.

OLIVEIRA, Phelipe Sena; BARROS, Luciano Sales; SILVEIRA JÚNIOR, Luiz Gonzaga de Queiroz. **Genetic Algorithm Applied to State Feedback Control Design**. In: *IEEE Latin America Transmission and Distribution Conference and Exposition*, São Paulo, 2010.

OLIVEIRA, Phelipe Sena; BARROS, Luciano Sales; SILVEIRA JÚNIOR, Luiz Gonzaga de Queiroz. **Projeto de Sistema de Controle Multivariável Baseado em Algoritmo Genético Otimizado**. In: Congresso Brasileiro de Automática – CBA Campina Grande, 2012.

OLIVEIRA, Phelipe Sena; BARROS, Luciano Sales; SILVEIRA JÚNIOR, Luiz Gonzaga de Queiroz; FARIAS, Missilene da Silva. **Otimização por Enxame de Partículas Aplicada ao Projeto de Controle Multivariável**. In: *XI Simpósio Brasileiro de Automação Inteligente – SBAI*, Fortaleza, CE, 2013.

ROLDÁN, Marco A. Montes de Oca. **On the Performance of Particle Swarm Optimizers**. *par diplôme d'Etudes Approfondies, Bruxelles Faculté des Sciences Appliquées. Institut de Recherches Interdisciplinaires et de D'éveloppements en Intelligence Artificielle*, 2006.

SERAPIÃO, Adriane Beatriz de Souza. **Fundamentos de Otimização por Inteligência de Enxames: Uma Visão Geral**. Revista Controle & Automação/Vol.20 no.3/Julho, Agosto e Setembro, 2009.

SHI ,Yuhui; EBERHART, Russell. *A Modified Particle Swarm Optimizer*. in: *Proc. of the IEEE Congress on Evolutionary Computation*. IEEE Service Center, USA (1998) 69–73.

SHI, Yuhui, EBERHART, Russel C. *Empirical study of particle swarm optimization*. In: *Proceedings of the 1999 IEEE Congress on Evolutionary Computation*, Piscataway, NJ, IEEE Press (1999) 1945–1950.

SIM, Kwang Mong; SUN, Weng Hong. *Multiple Ant-Colony Optimization for Network Routing*. *Proceedings of the First International Symposium on Cyber Worlds (CW.02) IEEE* . 2002.

SOARES, Gustavo L, **Algoritmos Genéticos: Estudo, Novas Técnicas e Aplicações**. Dissertação de mestrado, Belo Horizonte, MG, 1997.

SUCUPIRA, I. R. **Métodos Heurísticos Genéricos: Meta-heurísticas e Hiperheurísticas**. São Paulo, SP; 2004. 36p.

SWARANKAR, A.; GUPTA, N.; NIAZI, K. R. *Efficient Reconfiguration of Distribution Systems Using Ant Colony Optimization Adapted by Graph Theory*. In *Proceedings of the IEEE Power and Energy Society General Meeting*, Detroit, MI, 2011.

TAKAHASHI, Ricardo Hiroshi Caldeiras. **Controle Singular de Sistemas Incertos**. Tese de Doutorado, Campinas, SP, 1998. 236 p.

TANOMORU, Júlio. **Motivação, Fundamentos e Aplicações de Algoritmos Genéticos**. II Congresso Brasileiro de Redes Neurais, 1995.

TROFINO, Alexandre; COUTINHO, Daniel; BARBOSA, Karina Acosta. **Sistemas Multivariáveis: Uma abordagem via LMIs**. Santa Catarina. 2003.

WAI, Rong-Jong; CHUANG, Kun-Lun; LEE, Jeng-Dao. *On-Line Supervisory Control Design for Maglev Transportation System via Total Sliding-Mode Approach and Particle Swarm Optimization*. *IEEE transactions on automatic control*, vol. 55, no. 7, july 2010.

ZIELINSKI , K; JOOST, M.; LAUR, R.; ORLIK, B. *Choosing Suitable Variants of Differential Evolution and Particle Swarm Optimization for the Optimization of a PI Cascade Control. Published in: Optimization of Electrical and Electronic Equipment, 2008.*

APENDICE A

Um problema computacional é entendido como uma tarefa que é, em princípio, possível de ser resolvida por meio de um computador, significando que o problema pode ser descrito por um conjunto de instruções matemáticas.

Informalmente, um problema computacional consiste de instâncias do problema e suas respectivas instâncias. E denota-se a classe de todos os problemas de busca como NP. A classe P é a classe de todos os problemas de busca que são resolvidos em tempo polinomial.

Assim, NP são os problemas cuja soluções não podem ser encontradas e verificadas em tempo polinomial por um algoritmo não determinístico.

No nosso problema, o número de combinações possíveis dependerá diretamente do espaço de soluções, de modo que:

- Em um espaço composto por m linhas e n colunas, serão possíveis m^n possibilidades:
 - Tomando-se como exemplo uma matriz de controle de um sistema de quarta ordem com duas entradas e duas saídas, será necessário obter 8 ganhos de realimentação.

- Em um espaço de soluções onde os ganhos variam entre -10 e 10, teremos:

$$20^8 = 2,56E10 \text{ combinações possíveis.}$$

Em uma plataforma que fosse capaz de um cálculo a cada 1 milhão de análises por segundo, para verificar todos as 2,56E10 possibilidades seriam necessários 25600 segundos.

Se os ganhos devessem agora está contido entre os limites -10 e 10, porém agora particionados em espaços de 0,5:

$$20^8 = 6553600 \text{ segundos.}$$

Aumentando novamente o particionamento do espaço em 0,1 teríamos:

$$100^8 = 1E16 \text{ possibilidades, analisadas em 1E10 segundos.}$$

Assim, verifica-se que o tempo de verificação de todas as soluções crescerá de forma exponencial à medida que se aumenta o espaço de busca. Mas por outro lado, a verificação de cada possibilidade isolada, não cresce de forma exponencial.

APENDICE B

```

clc
clear all
format short
tempoIniciol=tic;

% _____ SISTEMAS A SEREM CONTROLADOS _____
% -----
% ----- Sistema 1 -----
% Este sistema instavel possui zeros lentos, alem de ganhos em zero
% -----
A =[-0.9221 -0.2315  0.0002  0.0095;
     0.2315 -0.9221 -0.0003 -0.0007;
     3.6129 45.8243   0      0.3652;
     0      0      1      0];

B=[0.9767   0;
   0   0.9767;
   0   0;
   0   0];

C=[0.5908 -0.0747  0 -0.0025;
   1      0      0   0];

D=[0 0; 0 0];

% Gbest = [3.2741 4.2593 0.4982 5.7703 -6.4833 6.4644 2.1594 -3.8073];
% LimiteSuperior = 8;
% LimiteInferior = -8;
% -----
% ----- Sistema 2 -----
% Da dinamica de um satelite artificial, possui zeros na origem
% -----
% omega=1; %ZERO TRANSMISSAO EM 0 = PRIMEIRO POLO
% A=[0 1*omega 0 0;3*omega^2 0 0 2*omega;0 0 0 1*omega;0 -2*omega 0 0];
% B=[0 0;1 0;0 1;0 0];
% C=[1 0 0 0;0 0 1 0];
% D=[0 0;0 0];
% -----
% Gbest = [-1.6896 2.4244 -0.9279 0.0090 0.3438 -0.3666 0.2172 0.0457];
% LimiteSuperior = 30;
% LimiteInferior = -30;
% -----
% ----- Sistema 3 -----
% INSTAVEL e apresenta zeros rapidos
% -----
% A = [ 3.1775 -15.0899 -7.0566 5.9162;
%       16.0387 2.5743 6.5514 -5.3053;
%       -15.0539 -3.5528 -12.6944 6.2062;
%       -16.9560 -19.7730 -19.4170 -12.5238];
% -----
% B = [0 0;
%      1 2.1736;
%      0 1;
%      0 0];
% -----
% C = [-1.6248 0 1 0;

```

```

%      1      0 0 0];
%
% D =[0 0; 0 0];
%
% Gbest = [-9.9      9      -6.1      -9.8      -9.5      8.9      9      -9];
% LimiteSuperior = 15;
% LimiteInferior = -15;
%
% ----- Sistema 4-----
% ESTAVEL, gerado randomicamente, possui zeros rapidos, porem uma resposta
extremamente lenta caracterizada por polos proximos a origem
% -----
% A =[-14.2515 12.1599  -5.1498  -0.5065;
%      -18.1202 15.7356 -10.3147  -7.1159;
%      -13.3168 14.4082 -13.3720 -17.4135;
%      14.5627  2.8396  13.9186   0.7267];
%
% B =[ 0      0;
%      1      0;
%      -3.0935 1;
%      0      0];
%
% C =[0 -3.0955 1 0;
%      1 0      0 0];
%
% D =[0 0; 0 0];
%
% Gbest = [2.6760 -9.8471 -9.4479 -3.5678 -8.6713 -9.2108  9.8579  0.9894];
% LimiteSuperior = 10;
% LimiteInferior = -10;
%
% ----- Sistema 5 -----
% Sistema INSTAVEL, com presença de UNDERSHOOTS.
% -----
% A =[-1.2118  -11.5493  0.2651  -8.8681;
%      1.0000      0      0      3.2037;
%      0      1      0      0;
%      3.4967      0      1      0];
%
% B =[ 0      0;
%      1.0000  3.4859;
%      0      1.0000;
%      0      0];
%
% C =[0 0 1      0;
%      1 0 0 2.6530];
%
% D =[ 0 0; 0 0];
%
% Gbest =[-2.2714  5.0000  1.1506  4.7  2.5734 -5.0000 -5.0000 -4.8];
% LimiteSuperior = 12;
% LimiteInferior = -12;
%
% zeroTransm = -9.2769;
% polos =[0.7470+6.7823i 0.7470-6.7823i -2.6897-0.0161i -2.6897+0.0161i];
%
% ----- Sistema 6 -----
% Sistema apresenta UNDERSHOOT e ultrapassagem positiva c ganho negativo
% -----
% A=[-1.2118  -11.5493  0.2651  -8.8681;
%      1.0000      0      0      0];

```

```

%           0    1.8752    5.1455   -13.6493;
%           0         0    1.0000         0];
%
%
% B = [      0         0;
%       1.0000         0;
%         0    1.0000;
%         0   -4.4410];
%
%
% C = [2.7998    0    1.0000    0;
%       1.0000    0         0    0];
%
%
% D = [0  0; 0  0];
%
%
% zeroTransm = -67.0116
% polos = -0.7074+3.3858i -0.7074-3.3858i 2.6742+2.7229i 2.6742-2.7229i
%
% Gbest = [9.8539  9.2519 -9.1348 -5.2258 -9.7772  3.1239  1.1983  9.9633];
% LimiteSuperior = 12;
% LimiteInferior = -12;
%
% ----- sistema 7 -----
% sistema com dinâmica oscilante
% -----
% A = [  7.3953   14.7638  -11.1369   -8.8681;
%       -17.9934  -13.2141   11.8429    1.1862;
%         -4.1342  -17.5455   13.7849    9.3066;
%           0         0    3.9597         0];
%
% B = [      0    3.7772;
%       -1.4279  -2.7871;
%       -4.7720    3.0997;
%       -2.4825  -1.8521];
%
% C = [      0   -1.1092    3.3889   -1.5873;
%       4.7615    4.9052         0    3.0144];
%
% D = [0  0; 0  0];
%
%
% zeroTransm = -3.3705+19.1071i -3.3705-19.1071i
% polos = 2.8604+13.3139i 2.8604-13.3139i 1.1227+3.6145i 1.1227-3.6145i
%
% Gbest = [9.5066  8.0219  2.6773 -9.7838  0.9244  9.5783  3.6677  9.7404];
% LimiteSuperior = 12;
% LimiteInferior = -12;
%
% ----- sistema 8 -----
% Sistema bastante oscilante
% -----
% A = [-2.7547  -17.3727   -8.8599   -7.5200;
%       16.8598   -3.0874  -13.3772   11.3144;
%         5.0872   19.7710   12.4780  -14.7083;
%        -1.0004   16.9044   18.6187   -0.6612];
%
% B = [-1.0276  -0.3846;
%       2.8978  -3.7746;
%       -1.8544  -2.7268;
%       3.6906  -4.5815];
%
% C = [-0.9240  0.1676  -2.5382  -1.3528;
%       0.8600  2.8461  2.7851  4.2739];
%
%

```

```

% D =[0 0; 0 0];
%
% zeroTransm = -11.4145+15.7077i -11.4145-15.7077i
% polos = 1.3119+22.3304i 1.3119-22.3304i 1.6755+9.8370i 1.6755-9.8370i
%
% Gbest = [-22.561 22.818 -6.057 -17.367 6.681 -13.440 22.495 -18.478];
% LimiteSuperior = 40;
% LimiteInferior = -40;
%
% ----- sistema 9-----
% Sistema originalmente com polo lento, polo instável e par oscilante
% -----
% A =[-1.2118 -11.5493 0.2651 -8.8681;
%      1.0000 0 0 0;
%      0 1.0000 3.5486 4.2691;
%      0 0 1.0000 0];
%
% B =[ 0 0;
%      1.0000 0;
%      0 1.0000;
%      0 3.8451];
%
% C =[ 0 0 -1.1664 0;
%      1.0000 0 0 0];
%
% D =[0 0; 0 0];
%
% Gbest = [-5.0590 5.0608 -4.9572 5.3564 -5.0060 5.2792 5.2291 -5.0539];
% LimiteSuperior = 12;
% LimiteInferior = -12;
%
% zeroTransm = -13.4626
% polos = 4.4592 -0.6630+3.3089i -0.6630-3.3089i -0.7963
%
% -----sistema 10 -----
% -----
% A =[ 18.4912 7.8909 15.9886 15.6917;
%      -15.1820 -9.9663 -17.1957 0.2169;
%      -16.2010 6.9520 3.0005 10.7168;
%      -8.6882 -15.2438 -1.4261 19.6383];
%
% B =[-3.7072 -3.0477;
%      0.4717 3.4976;
%      -4.6744 4.4812;
%      -1.1140 -2.4999];
%
% C =[-4.6905 3.9579 4.2866 2.8799;
%      -2.3737 -2.9962 3.1995 -1.3321];
%
% D =[0 0; 0 0];
%
% Gbest = [-14.961 -14.856 14.498 2.1645 -14.251 -14.590 -14.673 -14.755];
% LimiteSuperior = 16;
% LimiteInferior = -16;
%
% zeroTransm = -21.7140+18.6451i -21.7140-18.6451i
% polos = 1.5725+19.1421i 1.5725-19.1421i 14.0094+8.2252i 14.0094-8.2252i
%
% ----- sistema 11 -----
% -----
% A =[-6.8530 -12.5246 18.2465 -10.4518;

```

```

%      19.3631    13.9257   -10.1509   -16.0693;
%      -13.7707    0.9465    16.2948   -17.3035;
%      13.0875    12.5993    17.1144   -9.8791];
%
% B =[1.5814   -4.6883;
%      -0.1418   -0.7047;
%      3.5762   -4.4464;
%      0.3279    2.0842];
%
% C =[-3.9819    1.8342    1.2762   -2.7716;
%      3.0475    2.3606    4.7959   -2.9629];
%
% D =[0  0; 0  0];
%
% Gbest = [15.456 13.858 2.1038 -15.991 -15.9825 0.637 15.5273 14.8124];
% LimiteSuperior = 18;
% LimiteInferior = -18;
%
% zeroTransm = 1.0e+003 *[-2.0651  -0.0075]
% polos = -0.0445+23.9443i -0.0445 -23.9443i 6.7887+17.5936i 6.7887-7.5936i
%
% ----- sistema 12 -----
% -----
% A =[ 11.6214    14.3321   -12.4837   -15.7262;
%      3.5644    -3.3624    16.9501    11.8207;
%      -2.5511   -17.6942    13.5752    18.3370;
%      -10.9077  -19.9900    -7.1417    10.0687];
%
% B =[4.4462   -3.8008;
%      2.6490   -3.3736;
%      3.0215    4.9237;
%      0.5822   -0.8280];
%
% C = [ 1.1729 -2.9603 -2.3959 1.8612 ;
%      -3.8210 -1.0485 1.3911 4.3661];
%
% D =[0  0; 0  0];
%
% Gbest = [15.3044 -13.0755 -7.8480 9.7858 -13.0618 0.982 9.449 7.506];
% LimiteSuperior = 18;
% LimiteInferior = -18;
%
% zeroTransm = -40.8257 -25.5580
% polos = 7.6228+19.4855i 7.6228-19.4855i 15.7800 0.8774
%
% ----- sistema 13 -----
% -----
% A =[ 5.8417    2.2526   -13.3105    2.7156;
%      -3.5901   -15.3413    8.7019   -17.6372;
%      -10.9008    2.4444   -3.6742   -3.1410;
%      -7.7008    15.3225    13.6190   -18.6970];
% B =[ 1.6848   -2.1947;
%      1.2273   -4.7501;
%      -2.2379    2.0700;
%      -1.7605    3.6538];
%
% C =[-2.9196   -0.8470   -4.0785   -4.6227;
%      4.8826   -1.4605    0.8838   -3.0526];
%
% D =[0  0; 0  0];
%

```

```

% Gbest = [0.0020 -1.6905 -2.3338 11.5069 -3.7235 9.1822 12.525 0.3996];
% LimiteSuperior = 15;
% LimiteInferior = -15;
%
% zeroTransm = 1.0e+002 *[-2.5231+0.5520i -2.5231-0.5520i]
% polos = 12.2651 -13.3765 -15.3797+16.1429i -15.3797-16.1429i
%
% ----- sistema 14 -----
% -----
% A =[-2.6581 -13.7570 1.5945 15.0432;
% 9.1880 -1.6442 -5.3180 -2.1910;
% -7.7012 1.3672 8.1402 -13.5060;
% -6.7734 5.8689 2.0378 10.2364];
%
% B =[-4.2075 4.5664;
% 3.1366 -0.4260;
% -4.5978 4.6222;
% 2.6427 -0.2581];
%
% C =[-3.4642 4.4737 3.7096 1.6042;
% -3.3540 1.1481 3.8857 -1.6879];
%
% D =[0 0;
% 0 0];
%
% Gbest =[14.3356 5.7074 14.4472 -14.5918 -14.9629 6.6508 14.9736 -14.555];
% LimiteSuperior = 15;
% LimiteInferior = -15;
%
% zeroTransm = -9.3727+36.4111i -9.3727-36.4111i
% polos = 0.9762+16.1881i 0.9762-16.1881i 0.9086 11.2134
%
% ----- sistema 15 -----
% Sistema de difícil estabilização
% -----
% A =[ 17.0796 0.7363 6.8539 17.8668;
% -19.3720 2.3262 15.8907 -3.0062;
% 14.9101 4.3109 17.3640 -16.1070;
% -2.6580 6.7595 -10.9038 14.5635];
%
% B =[ 3.0455 -1.8962;
% -2.1216 -3.1408;
% 4.0214 3.9771;
% 2.3834 -4.9988];
%
% C =[ 2.2274 1.3308 2.5207 -2.1092;
% -3.2721 3.9749 -4.8228 2.9573];
%
% D =[0 0; 0 0];
%
% Gbest = [8.8263 8.6573 -0.0510 -10.2275 -0.5033 -10.2680 2.9796 -14.534];
% LimiteSuperior = 18;
% LimiteInferior = -18;
%
% zeroTransm = 1.0e+002*[-1.6760+0.9281i -1.6760-0.9281i]
% polos = -12.5556 26.8861 18.5014+10.6275i 18.5014-10.6275i
%
% _____ inicialização _____

Gbest=zeros(2,4);
Nmax=100;

```

```

NumeroParticulas=80;

fatorConvergencia=4.1;
dimensaoX=2/abs(2-fatorConvergencia-sqrt(fatorConvergencia^2-
4*fatorConvergencia));
c1=2.1; %fator de relevancia da busca global
c2=fatorConvergencia-c1; %fator de relevancia da busca local

estabiliza=false;
MaiorNotaGbest=0;
para=false;
duracaoMedia=0;
%___ Inicio do Algoritmo - Criando as paticulas e suas
%respectivas matrizes K de solucao _____

AmenosBK = struct();
nota=struct();
Lbest=struct();
VetorPosicao=struct();
VetorVelocidade=struct();
MaiorNotaLbest=struct();

for k=1: NumeroParticulas
    AmenosBK(k).solucoes=zeros(4,4);
    nota(k).solucoes = 0;
    Lbest(k).solucoes=zeros(2,4);
    VetorPosicao(k).solucoes=zeros(2,4);
    VetorVelocidade(k).solucoes=zeros(2,4);
    MaiorNotaLbest(k).solucoes=0;
end

for i=1:Nmax

    fprintf('iteracao %d',i)
    disp('-----')
    ---')

    for k=1:NumeroParticulas

        if i==1
            % Na primeira iteracao os vetores posicao e velocidade são
            gerados automaticamente dentro do limite de [LimiteInferior,
            LimiteSuperior]

            VetorPosicao(k).solucoes=NumeroRandomicoPSOnovaMatriz(LimiteSup
            erior,LimiteInferior);

            VetorVelocidade(k).solucoes=NumeroRandomicoPSOnovaMatriz(Limite
            Superior,LimiteInferior);

            Lbest(k).solucoes=VetorPosicao(k).solucoes;

        else

            % A partir da segunda iteracao cada particula ajusta sua
            posicao com base na melhor resposta obtida por ela mesma (Lbest) e
            pelo enxame (Gbest)

            VetorVelocidade(k).solucoes=dimensaoX*(VetorVelocidade(k).soluc

```

```

oes+c1*rand(1)*(Gbest-VetorPosicao(k).solucoes)+
c2*rand(1)*(Lbest(k).solucoes-VetorPosicao(k).solucoes));

VetorPosicao(k).solucoes=VetorPosicao(k).solucoes+VetorVelocida
de(k).solucoes;
end
posicao=VetorPosicao(k).solucoes;

for z=1:2
    for j=1:4
        while posicao(z,j)>=LimiteSuperior % Caso os valores
atualizados de posicao estejam fora dos limites estabelecidos

                posicao(z,j)=posicao(z,j)-(rand(1));
            end
            while posicao(z,j)<=LimiteInferior
                posicao(z,j)=posicao(z,j)+rand(1);
            end
        end
    end

    VetorPosicao(k).solucoes=posicao;
    AmenosBK(k).solucoes=(A-B*VetorPosicao(k).solucoes);
    [estabiliza,autovalores,situacao] =
TestarSeEstabiliza(AmenosBK(k).solucoes);

    % ===== Metodos de avaliacao=====

    tempoInicio=tic;
    [nota(k).solucoes, ACHOU]=TestarSolucaoPSO (AmenosBK(k).solucoes,
B,C,D,estabiliza,autovalores,VetorPosicao(k).solucoes,situacao);
    tempoFim=toc(tempoInicio);

    % =====

    duracaoMedia = duracaoMedia+tempoFim;
    if ACHOU == true
        tempoAlgoritmo=toc(tempoInicio1);
        para=true;
        duracaoMediaFinal=duracaoMedia/(NumeroParticulas);
    end
    if para == true
        break
    end

    % ===== atualizacoes de Lbst e Gbest =====

    if nota(k).solucoes>MaiorNotaLbest(k).solucoes
        Lbest(k).solucoes=VetorPosicao(k).solucoes;
        MaiorNotaLbest(k).solucoes=nota(k).solucoes;
    end
    if MaiorNotaLbest(k).solucoes>MaiorNotaGbest
        MaiorNotaGbest=MaiorNotaLbest(k).solucoes;
        Gbest=Lbest(k).solucoes;
    end
end
if para==true
    break
end

```

```

if i==Nmax && ACHOU==false
    amenosBK=A-B*Gbest;

    disp('Não atendeu as especificacoes no numero maximo de iteracoes
especificado, mas a melhor resposta encontrada foi:')
    Gbest
    x0=[0;0;0;0];
    tempo=0:0.05:4;
    u=ones(2,length(tempo));
    sistema=ss(amenosBK,B,C,D);
    [saida,tempo,x]=lsim(sistema,u,tempo,x0);
    figure(1)
    plot(tempo,saida)
end

```

```

% -----
function ValordeK = NumeroRandomicoPSOnovaMatriz
(LimiteSuperior,LimiteInferior )
% Calcula Numero aleatório dentro do range LimiteSuperior e LimiteInferior

for j=1:8
    NumeroRandomico(j)=LimiteInferior+(LimiteSuperior-
LimiteInferior)*rand(1);
end
    ValordeK=[NumeroRandomico(1) NumeroRandomico(2) NumeroRandomico(3)
NumeroRandomico(4);NumeroRandomico(5) NumeroRandomico(6)
NumeroRandomico(7) NumeroRandomico(8)];
end

```

```

% -----
function
[estabiliza,autovaloresOrdenados,situacao]=TestarSeEstabiliza(AmenosBK)
%Teste de estabilizacao
format short
estabiliza=false;
situacao=0;
autovalores=eig(AmenosBK);

if real(autovalores(:, :))<0
    estabiliza=true;
    %autovalores de A-BK jah ordenados em ordem crescente.
    autovaloresOrdenados=sort(real([autovalores(1,1) autovalores(2,1)
autovalores(3,1) autovalores(4,1)]));
    quantComplex=length(find(imag(autovalores)));
    posicao=find(imag(autovalores));
    if autovaloresOrdenados(1,4)==autovaloresOrdenados(1,3) &&
quantComplex==2

%         disp('%          X          |')
%         disp('%  --X--X-----|')
%         disp('%          X          |')

        situacao=0;
        autovaloresOrdenados(1,3)=autovalores(posicao(1),1);
        autovaloresOrdenados(1,4)=autovalores(posicao(2),1);
    else

```

```

        if autovaloresOrdenados(1,4)==autovaloresOrdenados(1,3) &&
quantComplex==4

%           disp('      X          X          |')
%           disp(' -----')
%           disp('      X          X          |')

        if autovaloresOrdenados(1,1)==real(autovalores(1,1))
            autovaloresOrdenados=[autovalores(1,1) autovalores(2,1)
autovalores(3,1) autovalores(4,1)];
        else
            autovaloresOrdenados=[autovalores(4,1) autovalores(3,1)
autovalores(2,1) autovalores(1,1)];
        end
        situacao=1;
    else

        if autovaloresOrdenados(1,3)==autovaloresOrdenados(1,2) &&
quantComplex==2

%           disp('          X          |')
%           disp(' --X-----X-----')
%           disp('          X          |')

        posicao=find(imag(autovalores));
        autovaloresOrdenados(1,3)=autovalores(posicao(1),1);
        autovaloresOrdenados(1,2)=autovalores(posicao(2),1);
        situacao=2;

    else
        if autovaloresOrdenados(1,1)==autovaloresOrdenados(1,2)
&& quantComplex==2

%           disp('%      X          |')
%           disp('% -----X---X-----')
%           disp('%      X          |')

        situacao=3;
        posicao=find(imag(autovalores));
        autovaloresOrdenados(1,1)=autovalores(posicao(1),1);
        autovaloresOrdenados(1,2)=autovalores(posicao(2),1);

    else
        if
autovaloresOrdenados(1,3)~=autovaloresOrdenados(1,4) &&
autovaloresOrdenados(1,3)~=autovaloresOrdenados(1,2) &&
autovaloresOrdenados(1,2)~=autovaloresOrdenados(1,1) && quantComplex==0

%           disp('%          |')
%           disp('% --X---X---X---X-----')
%           disp('%          |')

        situacao=4;

    else
        if
autovaloresOrdenados(1,3)==autovaloresOrdenados(1,4) &&
autovaloresOrdenados(1,2)~=autovaloresOrdenados(1,1) && quantComplex==0
%           disp('%          |')

```



```

if situacao==0

%
%          X          |
%  --X--X-----|-----
%          X          |

-----CALCULO DE OVERSHOOT-----

A estimação do overshoot e do tempo de subida é mais complexa. A taxa
de amortecimento do termo de 2a ordem dá uma
aproximação do overshoot. O efeito do pólo real Reduz o overshoot e
aumenta o Ts. Se o pólo real está localizado à direita dos pólos complexos
conjug a resposta é lenta, ou seja, o sistema se comporta como um sist
sobre-amortecido.

if autovalores(1,2)/real(autovalores(1,4))>=10

    omegaN=sqrt(real(autovalores(1,4))^2+imag(autovalores(1,4))^2);
    Qsi=-real(autovalores(1,4))/omegaN;
    max_overshoot=exp(-Qsi*pi/(sqrt(1-(Qsi^2))))*100; % disp('2a ordem')

else

    k=1;
    U=[1;1];
    m=1;
    n=1;
    omegaD=abs(imag(autovalores(1,4)));

    if tempo_de_acomodacao<(2*pi/abs(imag(autovalores(1,4))))

        passo=tempo_de_acomodacao/10;

    else

        passo= 0.5*pi/(2*abs(imag(autovalores(1,4)));

    end

    MatrizTransEstados=expm(AmenosBK*passo);
    picol(1)=0;
    pico2(1)=0;
    plotagem=true;
    p1=false;
    p2=false;
    tamanho=round(tempo_de_acomodacao/passo);

    if tamanho<=tamanhoMaximo
        for tempo=0:passo:(tempo_de_acomodacao)
            k=k+1;
            X(:,k)=MatrizTransEstados*X(:,k-1)+passo*B*U;
            saida1(k)=C(1,:)*(X(:,k));
            saida2(k)=C(2,:)*(X(:,k));

            if k>2
                if abs(saida1(k-2))<abs(saida1(k-1)) && abs(saida1(k-
1))>abs(saida1(k))
                    m=m+1;
                    picol(m)=(saida1(k-1));

```

```

        end
        if abs(saida2(k-2))<abs(saida2(k-1)) && abs(saida2(k-
1))>abs(saida2(k))
            n=n+1;
            pico2(n)=(saida2(k-1));
        end
    end
end

overshoot1=100*(max(pico1)-abs(saida1(1,k)))/abs(saida1(1,k));
overshoot2=100*(max(pico2)-abs(saida2(1,k)))/abs(saida2(1,k));
max_overshoot=max([overshoot1 overshoot2]);
pico1(1)=saida1(1,k);
pico2(1)=saida2(1,k);

if all(pico1<0)
    overshoot1=100*(max(abs(pico1))-abs(saida1(:,k)))/abs(saida1(1,k));
else
    if max(pico1)==0
        overshoot1=0;
    else
        overshoot1=100*(max(pico1)-(saida1(:,k)))/abs(saida1(1,k));
    end
end

if all(pico2<0)
    overshoot2=100*(max(abs(pico2))-abs(saida2(1,k)))/abs(saida2(1,k));
else
    if max(pico2)==0
        overshoot2=0;
    else
        overshoot2=100*(max(pico2)-(saida2(1,k)))/abs(saida2(1,k));
    end
end

max_overshoot=max([overshoot1 overshoot2]);
end
end

if max_overshoot<0
    max_overshoot=0;
end

if max_overshoot<=criterioParada_Overshoot
    overshoot=true;
    if tempo_de_acomodacao<=criterioParada_Ts
        tempoEstabilizacao=true;
    end
end

if max_overshoot==0
    nota=nota+(1/0.1);
else
    nota=nota+(1/max_overshoot);
end

```

```

----- avaliacao do tempo de acomodacao:-----

if tempo_de_acomodacao>0.05&&tempo_de_acomodacao<=10*criterioParada_Ts
    nota=nota+1/tempo_de_acomodacao;
end
else
===== SITUACAO 1 =====
if situacao==1

%      X      X      |
%  -----
%      X      X      |

    if abs(imag(autovalores(1,1)))>=abs(imag(autovalores(1,4)))
        passo= 0.5*pi/(2*abs(imag(autovalores(1,1))));
    else
        passo= 0.5*pi/(2*abs(imag(autovalores(1,4))));
    end
    tamanho=tempo_de_acomodacao/passo;
    if real(autovalores(1,1))/real(autovalores(1,4))<7 ||
abs(imag(autovalores(1,1)))>abs(imag(autovalores(1,4))) &&
tamanho<=tamanhoMaximo
        k=1;
        U=[1;1];
        m=1;
        n=1;
        frequencia=pi/abs(imag(autovalores(1,4)));
        MatrizTransEstados=expm(AmenosBK*passo);
        picol(1)=0;
        pico2(1)=0;

        for tempo=0:passo:(tempo_de_acomodacao)
            k=k+1;
            X(:,k)=MatrizTransEstados*X(:,k-1)+passo*B*U;
            saida1(k)=C(1,:) * (X(:,k));
            saida2(k)=C(2,:) * (X(:,k));
            if k>2
                if abs(saida1(k-2))<abs(saida1(k-1)) &&
abs(saida1(k-1))>abs(saida1(k))
                    m=m+1;
                    picol(m)=(saida1(k-1));
                end
                if abs(saida2(k-2))<abs(saida2(k-1)) &&
abs(saida2(k-1))>abs(saida2(k))
                    n=n+1;
                    pico2(n)=(saida2(k-1));
                end
            end
        end

        picol(1)=saida1(1,k);
        pico2(1)=saida2(1,k);

        if all(picol<0)
            overshoot1=100*(max(abs(picol))-
abs(saida1(:,k)))/abs(saida1(1,k));
        else
            if max(picol)==0
                overshoot1=0;
            else

```

```

                                overshoot1=100*(max(pico1)-
(saida1(:,k)))/abs(saida1(1,k));
                                end
                                end

                                if all(pico2<0)
                                overshoot2=100*(max(abs(pico2))-
abs(saida2(1,k)))/abs(saida2(1,k));
                                else
                                if max(pico2)==0
                                overshoot2=0;
                                else
                                overshoot2=100*(max(pico2)-
(saida2(1,k)))/abs(saida2(1,k));
                                end
                                end

                                max_overshoot=max([overshoot1 overshoot2]);
else
omegaN=sqrt(real(autovalores(1,4))^2+imag(autovalores(1,4))^2);
Qsi=-real(autovalores(1,4))/omegaN;
max_overshoot=exp(-Qsi*pi/(sqrt(1-(Qsi^2))))*100;
end
if max_overshoot<0
max_overshoot=0;
end

if tempo_de_acomodacao<=criterioParada_Ts
tempoEstabilizacao=true;
if max_overshoot<=criterioParada_Overshoot
overshoot=true;
end
end

if max_overshoot==0
nota=nota+(1/0.1);
else
nota=nota+(1/max_overshoot);
end

if tempo_de_acomodacao>0.05&& tempo_de_acomodacao
<=5*criterioParada_Ts
nota=nota+1/tempo_de_acomodacao;
end
else
=====SITUACAO 2=====
if situacao==2

%           X           |')
%  --X-----X-----')
%           X           |')

%  -----calculo de overshoot-----

passo= 0.25*pi/(2*abs(imag(autovalores(1,3)));
tamanho=tempo_de_acomodacao/passo;

if real(autovalores(1,2))/autovalores(1,4)>7.5||
tamanho>tamanhoMaximo
max_overshoot=0;

```

```

else
    k=1;
    U=[1;1];
    m=1;
    n=1;
    MatrizTransEstados=expm(AmenosBK*passo);
    picol(1)=0;
    pico2(1)=0;
    plotagem=true;

    for tempo=0:passo:(tempo_de_acomodacao)
        k=k+1;
        X(:,k)=MatrizTransEstados*X(:,k-1)+passo*B*U;
        saida1(k)=C(1,:)*(X(:,k));
        saida2(k)=C(2,:)*(X(:,k));
        if k>2
            if abs(saida1(k-2))<abs(saida1(k-1)) &&
abs(saida1(k-1))>abs(saida1(k))
                m=m+1;
                picol(m)=(saida1(k-1));
            end
            if abs(saida2(k-2))<abs(saida2(k-1)) &&
abs(saida2(k-1))>abs(saida2(k))
                n=n+1;
                pico2(n)=(saida2(k-1));
            end
        end
    end

    overshoot1=100*(max(picol)-
abs(saida1(:,k)))/abs(saida1(1,k));
    overshoot2=100*(max(pico2)-
abs(saida2(1,k)))/abs(saida2(1,k));
    max_overshoot=max([overshoot1 overshoot2]);
    picol(1)=saida1(1,k);
    pico2(1)=saida2(1,k);

    if all(picol<0)
        overshoot1=100*(max(abs(picol))-
abs(saida1(:,k)))/abs(saida1(1,k));
    else
        if max(picol)==0
            overshoot1=0;
        else
            overshoot1=100*(max(picol)-
(saida1(:,k)))/abs(saida1(1,k));
        end
    end

    if all(pico2<0)
        overshoot2=100*(max(abs(pico2))-
abs(saida2(1,k)))/abs(saida2(1,k));
    else
        if max(pico2)==0
            overshoot2=0;
        else
            overshoot2=100*(max(pico2)-
(saida2(1,k)))/abs(saida2(1,k));
        end
    end
end

```

```

max_overshoot=max([overshoot1 overshoot2]);

end
if max_overshoot<0
    max_overshoot=0;
end

% -----atualizacao de nota-----
if tempo_de_acomodacao<=criterioParada_Ts
    tempoEstabilizacao=true;

    if max_overshoot<=criterioParada_Overshoot
        overshoot=true;
    end
end

if max_overshoot==0
    nota=nota+(1/0.1);
else
    nota=nota+(1/max_overshoot);
end

if tempo_de_acomodacao>0.05 &&
tempo_de_acomodacao<=5*criterioParada_Ts
    nota=nota+1/tempo_de_acomodacao;
end

else

===== SITUACAO 3 =====
if situacao==3

%   X           |')
% -----X-----X-----')
%   X           |')

passo= 0.25*pi/(2*abs(imag(autovalores(1,1)));
tamanho=tempo_de_acomodacao/passo;

if real(autovalores(1,1))/autovalores(1,4)<7.5 &&
tamanho<=tamanhoMaximo
    k=1;
    U=[1;1];
    m=1;
    n=1;
    MatrizTransEstados=expm(AmenosBK*passo);
    pico1(1)=0;
    pico2(1)=0;
    plotagem=true;

    for tempo=0:passo:(tempo_de_acomodacao)
        k=k+1;
        X(:,k)=MatrizTransEstados*X(:,k-1)+passo*B*U;
        saida1(k)=C(1,:)*(X(:,k));
        saida2(k)=C(2,:)*(X(:,k));

        if k>2
            if abs(saida1(k-2))<abs(saida1(k-1))&&
abs(saida1(k-1))>abs(saida1(k))

```

```

        m=m+1;
        picol(m)=(saida1(k-1));
    end

        if abs(saida2(k-2))<abs(saida2(k-1)) &&
abs(saida2(k-1))>abs(saida2(k))
            n=n+1;
            pico2(n)=(saida2(k-1));
        end
    end
end

overshoot1=100*(max(picol)-
abs(saida1(:,k)))/abs(saida1(1,k));
overshoot2=100*(max(pico2)-
abs(saida2(1,k)))/abs(saida2(1,k));
picol(1,1)=saida1(1,k);
pico2(1,1)=saida2(1,k);

if all(picol<0)
    overshoot1=100*(max(abs(picol))-
abs(saida1(:,k)))/abs(saida1(1,k));
else
    if max(picol)==0
        overshoot1=0;
    else
        overshoot1=100*(max(picol)-
(saida1(:,k)))/abs(saida1(1,k));
    end
end

if all(pico2<0)
    overshoot2=100*(max(abs(pico2))-
abs(saida2(1,k)))/abs(saida2(1,k));
else
    if max(pico2)==0
        overshoot2=0;
    else
        overshoot2=100*(max(pico2)-
(saida2(1,k)))/abs(saida2(1,k));
    end
end

max_overshoot=max([overshoot1 overshoot2]);
else
    max_overshoot=0;
    teste=true;
end

% -----Atualizacao da nota-----

if tempo_de_acomodacao<=criterioParada_Ts
    tempoEstabilizacao=true;

    if max_overshoot<=criterioParada_Overshoot
        % OVERSHOOT LOGICO
        overshoot=true;
    end
end

if max_overshoot==0

```

```

        nota=nota+(1/0.1);
    else
        nota=nota+(1/max_overshoot);
    end

    if tempo_de_acomodacao>0.05 &&
tempo_de_acomodacao<=5*criterioParada_Ts
        nota=nota+1/tempo_de_acomodacao;
    end
else
    ===== SITUACAO 4 =====
    if situacao==4

        %
        % --X---X---X---X-----
        %
        % -----calculo de Ts e Overshoot-----

        tempo_de_acomodacao=-3.9/real(autovalores(1,4));
        max_overshoot=0;
        overshoot=true;

        % -----atualizacao da nota-----
        nota=nota+(1/0.1); % Referente ao calculo do overshoot.

        if tempo_de_acomodacao<=criterioParada_Ts
            tempoEstabilizacao=true;
        end

        if tempo_de_acomodacao>0.05 &&
tempo_de_acomodacao<=5*criterioParada_Ts
            nota=nota+1/tempo_de_acomodacao;
        end
    else

        ===== SITUACAO 5 =====
        if situacao==5
            % Em um sistema de 2a ordem c polos reais e iguais o
sistema eh criticamente amortecido e Qsi=1 e omegaN igual aos polos do
sistema

            % -----calculo de overshoot-----
            max_overshoot=0; % sistema com dinâmica
criticamente amortecida e amortecida
            overshoot=true;
            nota=nota+(1/0.1);

            ----- buscar o tempo de acomodacao:-----

            tempo_de_acomodacao=-3.9/real(autovalores(1,4));

            if tempo_de_acomodacao<=criterioParada_Ts
                tempoEstabilizacao=true;
            end

            if tempo_de_acomodacao>0.05 && tempo_de_acomodacao
<=5*criterioParada_Ts
                nota=nota+1/tempo_de_acomodacao;
            end
        end
    end
end

```

```

end
else

===== SITUACAO 6 =====
if situacao==6
% Em um sistema de 2a ordem c polos reais e iguais o
sistema eh criticamente amortecido e Qsi=1 e omegaN igual aos polos do
sistema

-----calculo de overshoot-----
max_overshoot=0;
overshoot=true;
nota=nota+(1/0.1);

----- buscar o tempo de acomodacao:-----

tempo_de_acomodacao=-3.9/real(autovalores(1,4));

if tempo_de_acomodacao<=criterioParada_Ts
tempoEstabilizacao=true;
end
if tempo_de_acomodacao>0.05 &&
tempo_de_acomodacao<=5*criterioParada_Ts
nota=nota+1/tempo_de_acomodacao;
end
end
end
end
end
end
end

% =====VERIFICAR CRITERIO DE PARADA=====
% =====

if tempoEstabilizacao==true && overshoot==true
sistema=ss(AmenosBK,B,C,D);
ACHOU=true;
disp ('o sistema estabiliza e atende as especificacoes para
a solucao:')
MatrizKParticula
autovalores
tempo_de_acomodacaoEmax_overshoot_Namostras =
[tempo_de_acomodacao max_overshoot k];
tempo_de_acomodacaoEmax_overshoot_Namostras

testa e plota a solucao para verificar confiabilidade dos
criterios de paradas impostos: Ts e Mp
x0=[0 0 0 0]';
tempoFinal=round(tempo_de_acomodacao+3);
temposimulacao=0:0.02:tempoFinal;
r=1*ones(2,length(temposimulacao));
[saida,temposimulacao,x]=lsim(sistema,r,temposimulacao,x0);

if plotagem==true
figure(1)
plot(saida2)
hold on
plot(saida1)
title('gráfico de resposta APROXIMADA')

```

```
        text(930,1.3,'resposta ao degrau')
        hold off
        figure(2)
    end

    plot(temposimulacao,saida)
    grid
    title('gráfico de resposta ao degrau unitário')
    text(930,1.3,'resposta ao degrau')

%=====
        end
    else
        nota=0;
    end
end
```

APENDICE C

Tabela I - Tabela de tempos médios para o PSO ao longo de 10s de simulação com probabilidade de 1% de ocorrer variação a cada amostra de tempo.

Prob_Var	Lim_Var	Média de Variações em 10s	Média de Acionamentos do PSO (Nac)	tempo médio do PSO/Nac com a Simulação
1%	1%	5	1,0667	0,003794
1%	2%	5	1,2	0,003667
1%	5%	5	1,4	0,003758
1%	10%	5	1,6	0,00457
1%	20%	5	1,8667	0,005826

Fonte: Autora.

Tabela II - Tabela de tempos médios para o PSO ao longo de 10s de simulação com probabilidade de 5% de ocorrer variação a cada amostra de tempo.

Prob_Var	Lim_Var	Média de Variações em 10s	Média de Acionamentos do PSO (Nac)	tempo médio do PSO/Nac com a Simulação
5%	1%	26	2,6667	0,00395
5%	2%	26,13333333	3	0,003923
5%	5%	26,26666667	4,8	0,007343
5%	10%	25,93333333	7,2667	0,008072
5%	20%	26,26666667	11,667	0,009693

Fonte: Autora.

Tabela III - Tabela de tempos médios para o PSO ao longo de 10s de simulação com probabilidade de 10% de ocorrer variação a cada amostra de tempo.

Prob_Var	Lim_Var	Média de Variações em 10s	Média de Acionamentos do PSO (Nac)	tempo médio do PSO/Nac com a Simulação
10%	1%	78,86666667	3,2667	0,004083
10%	2%	78,66666667	4,0667	0,005968
10%	5%	79,4	8,3333	0,008282
10%	10%	78,66666667	15,733	0,008441
10%	20%	78,8	36,067	0,011351

Fonte: Autora.

Tabela IV - Tabela de tempos médios para o PSO ao longo de 10s de simulação com probabilidade de 20% de ocorrer variação a cada amostra de tempo.

Prob_Var	Lim_Var	Média de Variações em 10s	Média de Acionamentos do PSO (Nac)	tempo médio do PSO/Nac com a Simulação
20%	1%	167	3,3333	0,003993
20%	2%	166,3333333	4,2	0,006255
20%	5%	167,3333333	10,8	0,008074
20%	10%	166,6666667	25,2	0,008805
20%	20%	167,3333333	71,733	0,012337

Fonte: Autora.

Tabela V - Tabela de tempos médios para o PSO ao longo de 10s de simulação com probabilidade de 40% de ocorrer variação a cada amostra de tempo.

Prob_Var	Lim_Var	Média de Variações em 10s	Média de Acionamentos do PSO (Nac)	tempo médio do PSO/Nac com a Simulação
40%	1%	375,3333333	3,3333	0,00408
40%	2%	375,7333333	4,6	0,006376
40%	5%	375,4	14,067	0,008435
40%	10%	375,4666667	44,733	0,008863
40%	20%	375,2666667	155,1	0,012201

Fonte: Autora.