

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MARLON DE CARVALHO PASCOAL

SÍNTESE DE CONTROLE ROBUSTO \mathcal{H}_∞ DE ORDEM
REDUZIDA PARA SISTEMAS LINEARES CONTÍNUOS
USANDO ALGORITMO GENÉTICO E OTIMIZAÇÃO POR
ENXAME DE PARTÍCULAS

DISSERTAÇÃO

CORNÉLIO PROCÓPIO

2016

MARLON DE CARVALHO PASCOAL

**SÍNTESE DE CONTROLE ROBUSTO \mathcal{H}_∞ DE ORDEM
REDUZIDA PARA SISTEMAS LINEARES CONTÍNUOS
USANDO ALGORITMO GENÉTICO E OTIMIZAÇÃO POR
ENXAME DE PARTÍCULAS**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de “Mestre em Engenharia Elétrica”

Orientador: Prof. Dr. Cristiano Marcos Agulhari

CORNÉLIO PROCÓPIO

2016

TERMO DE APROVAÇÃO

Título da Dissertação Nº _____:

Síntese de Controle Robusto H_∞ de Ordem Reduzida para Sistemas Lineares Contínuos usando Algoritmo Genético e Otimização por Enxame de Partículas

por

Marlon de Carvalho Pascoal

Orientador: **Prof. Dr. Cristiano Marcos Agulhari**

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM ENGENHARIA ELÉTRICA – Área de Concentração: Sistemas Eletrônicos Industriais, pelo Programa de Pós-Graduação em Engenharia Elétrica – PPGE – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Cornélio Procópio, às 10h do dia 25 de Agosto de 2016. O trabalho foi aprovado pela Banca Examinadora, composta pelos professores:

Prof. Dr. Cristiano Marcos Agulhari
(Presidente)

Prof. Dr. Márcio Roberto Covacic
(UEL)

Prof. Dr. Fábio Renan Durand
(UTFPR)

Visto da coordenação:

Prof. Dr. Alessandro do Nascimento Vargas
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica
UTFPR Câmpus Cornélio Procópio

A Folha de Aprovação assinada encontra-se na Coordenação do Programa.

RESUMO

PASCOAL, M. C.. SÍNTESE DE CONTROLE ROBUSTO \mathcal{H}_∞ DE ORDEM REDUZIDA PARA SISTEMAS LINEARES CONTÍNUOS USANDO ALGORITMO GENÉTICO E OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS. 61 f. Dissertação – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2016.

Este trabalho trata da síntese de controladores \mathcal{H}_∞ por realimentação dinâmica de saída sendo esta de ordem reduzida, para sistemas lineares incertos a tempo contínuo. Os parâmetros invariantes no tempo pertencem a um domínio politópico e podem afetar todas as matrizes do sistema. A busca por um controlador de ordem reduzida é convertida no cálculo de um ganho de realimentação estática de saída para um sistema aumentado. Para resolver o problema, um procedimento com base em desigualdades matriciais lineares (LMIs) de dois estágios é revisado. São analisados, neste trabalho, os efeitos da adição dos procedimentos metaheurísticos Algoritmo Genético (GA) ou Otimização por Enxame de Partículas (PSO) ao procedimento de dois estágios. Os algoritmos são calibrados utilizando uma base de sistemas chamada COMPlib, assim como sistemas estabilizáveis precisamente conhecidos gerados aleatoriamente, e os resultados são comparados com os resultados de uma técnica de síntese considerada como estado da arte, chamada HIFOO. São utilizados como experimentos numéricos sistemas incertos estabilizáveis gerados aleatoriamente e um sistema incerto disponível na literatura. Os resultados produzidos por essas técnicas mostraram que essas foram capazes de minimizar a norma \mathcal{H}_∞ para os sistemas testados e esses resultados ficaram próximos ou melhores se comparados às demais estratégias disponíveis na literatura.

Palavras-chave: algoritmo genético, otimização por enxame de partículas, desigualdades matriciais lineares, realimentação dinâmica de saída, controladores de ordem reduzida

ABSTRACT

PASCOAL, M. C.. SYNTHESIS OF REDUCED-ORDER ROBUST \mathcal{H}_∞ CONTROL FOR CONTINUOUS-TIME UNCERTAIN LINEAR SYSTEMS USING GENETIC ALGORITHM AND PARTICLE SWARM OPTIMIZATION. 61 f. Dissertação – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2016.

This dissertation is concerned with the problem of designing reduced order robust \mathcal{H}_∞ dynamic output feedback controllers, for uncertain continuous-time linear systems. The uncertain time-invariant parameters belong to a polytopic domain and may affect all the system matrices. The search for a reduced-order controller is converted into a problem of computing a static output feedback control design for an augmented system. To solve the problem, a two-stage linear matrix inequality (LMI) procedure is reviewed. It is analyzed, in this work, the effects of adding either a Genetic Algorithm (GA) or a Particle Swarm Optimization (PSO) metaheuristic procedures to the two-stage LMI procedure. The algorithms are calibrated using systems from a database known as COMPlib, and also randomly generated precisely known stabilizable systems, and the results are compared with a state-of-art synthesis technique, named HIFOO. As numeric experiments, randomly generated uncertain stabilizable systems and an uncertain system available in literature are used. The results provided by these techniques showed that they were capable to minimize \mathcal{H}_∞ norm and the results were close or better when compared to other techniques available in the literature.

Keywords: genetic algorithm, particle swarm optimization, linear matrix inequality, Dynamic output feedback, Reduced order controllers

LISTA DE FIGURAS

FIGURA 10 – Diagrama de blocos de um sistema com perturbação externa	19
FIGURA 12 – Fluxograma de um Algoritmo Genético Contínuo	31
FIGURA 13 – Elementos fundamentais para o cálculo do próximo deslocamento de uma partícula	35
FIGURA 15 – Fluxograma do Algoritmo Genético implementado	38
FIGURA 16 – Fluxograma do PSO implementado	39
FIGURA 18 – Comparação dos resultados dos sistemas do COMPLib para GA e HIFOO com $n_c = 0$ e $n_c = n_x$	43
FIGURA 19 – Comparação dos resultados dos sistemas do COMPLib para PSO e HIFOO com $n_c = 0$ e $n_c = n_x$	44
FIGURA 20 – Comparação dos resultados dos sistemas do COMPLib para PSO e GA com $n_c = 0$ e $n_c = n_x$	44
FIGURA 21 – Comparação dos resultados dos sistemas do COMPLib com controla- dores $n_c = 0$ e $n_c = n_x$	45
FIGURA 22 – Comparação dos resultados dos sistemas estabilizáveis precisamente conhecidos gerados aleatoriamente para GA e HIFOO com $n_c = 0$ e $n_c = n_x$	48
FIGURA 23 – Comparação dos resultados dos sistemas estabilizáveis precisamente conhecidos gerados aleatoriamente para PSO e HIFOO com $n_c = 0$ e $n_c = n_x$	48
FIGURA 24 – Comparação dos resultados dos sistemas estabilizáveis precisamente conhecidos gerados aleatoriamente para PSO e GA com $n_c = 0$ e $n_c = n_x$	49
FIGURA 25 – Comparação dos resultados dos sistemas estabilizáveis precisamente conhecidos gerados aleatoriamente para com controladores $n_c = 0$ e $n_c = n_x$	50
FIGURA 26 – Comparação dos resultados dos sistemas estabilizáveis incertos gerados aleatoriamente para PSO e GA com $n_c = 0$ e $n_c = n_x$	52
FIGURA 27 – Comparação dos resultados dos sistemas estabilizáveis incertos gerados aleatoriamente para com controladores $n_c = 0$ e $n_c = n_x$	53

LISTA DE TABELAS

TABELA 13 – Limitantes da norma \mathcal{H}_∞ para GA e PSO comparados aos resultados do HIFOO para os exemplos de sistema do COMPLib	41
TABELA 14 – Limitantes da norma \mathcal{H}_∞ para GA e PSO comparados aos resultados do HIFOO para sistemas estabilizáveis precisamente conhecidos gerados aleatoriamente	46
TABELA 15– Limitantes da norma \mathcal{H}_∞ para GA e PSO para sistemas incertos estabilizáveis gerados aleatoriamente	51
TABELA 16 – Limitantes da norma \mathcal{H}_∞ produzidos pelo GA e PSO para o sistema incerto (66)	54

LISTA DE SIGLAS

SOF	<i>Static Output Feedback</i> (Realimentação Estática de Saída)
ILMIs	<i>Iterative Linear Matrix Inequalities</i> (Desigualdades Matriciais Lineares Iterativas)
BMI	<i>Bilinear Matrix Inequalities</i> (Desigualdades Matriciais Bilineares)
LMI	<i>Linear Matrix Inequalities</i> (Desigualdades Matriciais Lineares)
DOF	<i>Dynamic Output Feedback</i> (Realimentação Dinâmica de Saída)
NP-hard	<i>Hard Nondeterministic Polynomial Time Problem</i> (Problema Não-determinístico difícil em tempo polinomial)
HIFOO	<i>H-Infinity Fixed-Order Optimization</i>
BFGS	<i>Broyden Fletcher Goldfarb Shanno</i>
COMPlib	<i>COstrained Matrix-optimization Problem library</i>
GA	<i>Genetic Algorithms</i> (Algoritmos Genéticos)
PSO	<i>Particle Swarm Optimization</i> (Otimização por Enxame de Partículas)
SSF	<i>Static State Feedback</i> (realimentação estática de estados)
MD5	<i>Message Digest 5</i>

LISTA DE SÍMBOLOS

\mathcal{H}_∞	Norma H-Infinito
Λ_N	Simplex unitário
$K_{sf}(\alpha)$	Ganho de realimentação de estados dependente de parâmetros
γ_{dof}	Norma \mathcal{H}_∞ produzida pela realimentação dinâmica de saída
K_{dof}	Ganho de Realimentação dinâmica de saída
N_{pop}	Tamanho da população
N_{kept}	Quantidade de cromossomos na seleção natural
$Pb_{mutation}$	Probabilidade de mutação
DL	Taxa de Mutação Dinâmica
N_{swarm}	Tamanho do enxame
c_{max}	Taxa de confiança nas partículas informantes

SUMÁRIO

1 INTRODUÇÃO	10
1.1 MOTIVAÇÃO	10
1.2 OBJETIVOS	13
1.2.1 Objetivo Geral	13
1.2.2 Objetivos Específicos	13
1.3 JUSTIFICATIVA	13
1.4 ESTRUTURA DO TRABALHO	14
2 FUNDAMENTAÇÃO TEÓRICA	15
2.1 TEORIA DE CONTROLE MODERNO	15
2.1.1 Estabilidade Quadrática	16
2.1.2 Desigualdades Matriciais Lineares	17
2.1.3 Teoria de Controle Robusto	18
2.1.4 Síntese de Controladores	21
2.2 PROJETO DE CONTROLADOR POR REALIMENTAÇÃO DINÂMICA DE SAÍDA	24
3 METAHEURÍSTICAS DE OTIMIZAÇÃO	28
3.1 ALGORITMOS GENÉTICOS	30
3.2 OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS	32
4 PROPOSTA DE METAHEURÍSTICAS PARA O CONTROLE \mathcal{H}_∞ .	37
5 EXPERIMENTOS NUMÉRICOS	40
6 CONCLUSÕES	56
REFERÊNCIAS	58

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

O projeto de controladores por realimentação estática de saída (SOF, acrônimo em inglês para *Static Output Feedback*) é considerado o mais simples de ser implementado na prática, e vem sendo consideravelmente estudado nas últimas décadas. Na literatura, são encontradas abordagens que lidam com o projeto de controladores SOF para sistemas lineares, conforme em (SYRMOS et al., 1997). Dentre os métodos existentes, é possível destacar métodos com base nas equações de Riccati (TROFINO-NETO; KUCERA, 1993; KUCERA; SOUZA, 1995; GARCIA et al., 2001), abordagens convexas com base em técnicas de otimização (GEROMEL et al., 1998; HENRION; LASSERRE, 2006; GEROMEL et al., 1996), Desigualdades Matriciais Lineares Iterativas (ILMIs, acrônimo em inglês para *Iterative Linear Matrix Inequalities*) expressas como formulações de Desigualdades Matriciais Bilineares (BMI, acrônimo em inglês para *Bilinear Matrix Inequalities*) (GHAOUI et al., 1997; CAO et al., 1998; LEIBFRITZ, 2001; ROSINOVA et al., 2003; HUANG; NGUANG, 2006) e condições suficientes em termos de Desigualdades Matriciais Lineares (LMIs, acrônimo em inglês para *Linear Matrix Inequalities*) (SHAKED, 1997; PREMPAIN; POSTLETHWAITE, 2001; BARA; BOUTAYEB, 2005; PEAUCELLE; ARZELIER, 2005; BENTON; SMITH, 2005; LO; LIN, 2003; HO; LU, 2003; CRUSIUS; TROFINO, 1999; SOUZA; TROFINO, 2000).

Projetar controladores SOF é um dos problemas em aberto mais importantes na prática e na teoria de controle. Da mesma forma, o projeto de controladores por realimentação dinâmica de saída (DOF, acrônimo em inglês para *Dynamic Output Feedback*) de ordem reduzida tem atraído um interesse crescente. Este interesse pode ser justificado simplesmente pelo fato de que, para uma série de aplicações, as variáveis de estado podem não estar completamente acessíveis. Neste caso, duas soluções estão disponíveis para projetar o controlador apropriado: controladores baseados em observação e controladores por realimentação de saída. O projeto de controladores por realimentação de saída é também um dos problemas mais desafiadores na área de pesquisa de teoria de controle, uma

vez que não foram encontrados algoritmos de tempo polinomial ou heurísticas eficientes para a estabilização por realimentação estática de saída conhecidos na literatura (GUMUSSOY et al., 2009). Este problema também é caracterizado como Problema Não-determinístico difícil em tempo polinomial (NP-hard, acrônimo em inglês para *Hard Nondeterministic Polynomial Time Problem*) (FU; LUO, 1997) se analisado com desigualdades matriciais lineares (LMI) e abordagens com restrições não convexas.

Contudo, a não convexidade das condições de síntese é uma fonte de dificuldade (GUMUSSOY et al., 2009). A estabilidade de um sistema linear contínuo, em termos matemáticos, é equivalente à localização dos autovalores da matriz de dinâmica no semiplano esquerdo. Entretanto, se formulado no espaço euclidiano de matrizes reais, essa restrição de estabilidade é não convexa. Dessa forma, os espaços de busca de controladores que estabilizam o sistema são não convexos e algumas vezes também são desconexos. Portanto, a síntese de controladores por realimentação de saída para estabilização é um problema complexo, e essa dificuldade aumenta se critérios de desempenho forem considerados, por exemplo a minimização da norma \mathcal{H}_∞ do sistema controlado. A norma \mathcal{H}_∞ indica a máxima relação entre a energia dos sinais de entrada ou ruído e saída do sistema, portanto a minimização da norma \mathcal{H}_∞ também assegura uma maior robustez do controlador (BOYD et al., 1994). Há várias técnicas propostas para superar essas dificuldades, por exemplo aproximações convexas de regiões de estabilidade não convexas, formulações de LMIs de análise e projeto de restrições e programação não convexa.

Há também abordagens numéricas como o HIFOO (acrônimo em inglês para *H-Infinity Fixed-Order Optimization*) (GUMUSSOY et al., 2009), que é um algoritmo híbrido que combina um algoritmo quasi-Newton (BFGS, acrônimo em inglês para *Broyden Fletcher Goldfarb Shanno*) com uma série de otimizações locais, na tentativa de otimizar localmente o melhor ponto encontrado pelo BFGS. Caso o algoritmo não tenha sucesso, também é combinada uma fase de gradiente de amostragem para refinar a aproximação do minimizador local, retornando um resultado grosseiro, porém otimizado localmente.

Dentre essas e outras propostas, é possível destacar os métodos baseados na teoria de Lyapunov, que normalmente adaptam condições convexas para obter condições LMIs suficientes para o projeto de controle por realimentação de saída robusto estático ou de ordem reduzida, e também métodos de otimização local. O problema é que a maioria desses métodos não são capazes de lidar com incertezas que afetam todas as matrizes do sistema. As incertezas são margens ou variações previstas no modelo matemático do sistema por motivos como dinâmicas não modeladas, incertezas paramétricas, ruídos,

linearizações, etc. Por esse motivo, a denominação sistema robusto implica que o modelo físico pode não estar precisamente representado pelo modelo matemático, além de também poder haver influências externas ao sistema.

Uma extensão que combina algoritmos aleatórios com LMIs para o projeto de controle por realimentação estática de saída é proposta em (PEAUCELLE; ARZELIER, 2001; ARZELIER et al., 2003), que consiste em dois estágios. O primeiro estágio computa um ganho de controle de realimentação de estados dependente de parâmetros que estabiliza o sistema, que é utilizado como entrada para o segundo estágio. O segundo estágio calcula um ganho de realimentação de saída robusto de ordem reduzida que minimiza a norma \mathcal{H}_∞ do controlador, usando condições LMIs suficientes. Em (OSTERTAG, 2009) é proposta uma técnica onde são convertidas condições BMIs em LMIs e uma técnica de dois estágios com base na proposta de (PEAUCELLE; ARZELIER, 2001; ARZELIER et al., 2003) é aplicada no cálculo do controlador de realimentação dinâmica de saída dinâmica que otimiza as normas $\mathcal{H}_\infty/\mathcal{H}_2$ do sistema.

Em (AGULHARI et al., 2011) é proposta uma técnica baseada em (PEAUCELLE; ARZELIER, 2001; ARZELIER et al., 2003, 2009, 2010), onde, diferentemente das demais abordagens disponíveis na literatura, é abordada a síntese de controladores de ordem reduzida usando matrizes de Lyapunov polinomiais dependentes de parâmetros. Essa técnica aborda a incerteza em todas as matrizes do sistema, contudo não há nenhum processo de otimização proposto para o primeiro estágio da técnica.

Sendo assim, o propósito deste trabalho é explorar o primeiro estágio da técnica proposta em (AGULHARI et al., 2011) e propor algoritmos metaheurísticos para otimizar o ganho de realimentação de estados que é utilizado como entrada do segundo estágio, e com isso otimizar os limitantes da norma \mathcal{H}_∞ do controlador do sistema. Esses algoritmos são calibrados e validados através de sistemas estabilizáveis precisamente conhecidos gerados aleatoriamente e também através de sistemas disponíveis na biblioteca COMPlib (acrônimo em inglês para *CO*nstrained *M*atrix-*o*ptimization *P*roblem *l*ibrary) (LEIBFRITZ; LIPINSKI, 2003), uma coleção que contém exemplos testes para projetos de sistemas de controle não lineares e problemas relacionados, comparando os resultados com a ferramenta de *benchmark* HIFOO com propósito de calibragem tanto para controladores estáticos, cuja ordem é igual a zero, quanto controladores de ordem completa. Na sequência, são utilizados sistemas incertos gerados aleatoriamente e um sistema massa-mola disponível na literatura para experimentos numéricos com controladores de ordem estática e completa.

1.2 OBJETIVOS

Os objetivos geral e específicos deste trabalho são descritos a seguir.

1.2.1 OBJETIVO GERAL

O objetivo geral deste trabalho é propor duas adições à técnica proposta em (AGULHARI et al., 2011). Essas adições consistem na busca pelo ganho ótimo da norma \mathcal{H}_∞ através de um controlador por realimentação dinâmica de saída de ordem reduzida para sistemas robustos, por meio do processamento inteligente da superfície do ganho de controle por realimentação de estados dependente de parâmetros através de Algoritmos Genéticos (GA, acrônimo em inglês para *Genetic Algorithms*) e Otimização por Enxame de Partículas (PSO, acrônimo em inglês para *Particle Swarm Optimization*).

1.2.2 OBJETIVOS ESPECÍFICOS

Para alcançar o objetivo estabelecido, são delimitados a seguir os marcos ou objetivos específicos que levarão à conclusão do projeto:

- Implementar os teoremas propostos em (AGULHARI et al., 2011);
- Implementar o GA;
- Implementar o PSO;
- Realizar testes numéricos com sistemas precisamente conhecidos disponíveis na biblioteca COMLib (LEIBFRITZ; LIPINSKI, 2003);
- Realizar testes numéricos com sistemas precisamente conhecidos gerados aleatoriamente;
- Realizar testes numéricos com sistemas incertos e comparar com os resultados da literatura;
- Realizar testes numéricos com sistemas incertos gerados aleatoriamente;

1.3 JUSTIFICATIVA

Existem diversas estratégias na literatura que abordam o problema do projeto de controladores por realimentação de saída, contudo, a maioria dessas propostas não é capaz

de lidar com incertezas que afetam todas as matrizes do sistema. Assim, este trabalho tem como contribuição principal a proposta de duas técnicas de otimização (GA e PSO) para uma técnica de síntese de controladores de realimentação de saída capazes de estabilizar sistemas em que todas as matrizes podem ser afetadas por incertezas, de forma a otimizar a norma \mathcal{H}_∞ do sistema controlado.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está estruturado como segue. No Capítulo 2, são introduzidas as teorias de controle relacionadas a síntese de controladores, sendo também apresentado neste capítulo o controlador por realimentação de saída dinâmico de ordem reduzida que minimiza a norma \mathcal{H}_∞ para sistemas incertos proposto em (AGULHARI et al., 2011). Introduções sobre GAs e PSO com algumas orientações são encontradas no Capítulo 3. Explicações sobre o desenvolvimento do projeto e detalhes de implementação dos algoritmos são apresentados no Capítulo 4. No Capítulo 5, são apresentados os resultados numéricos alcançados, e por fim são apresentadas as conclusões.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão apresentados os principais assuntos relacionados à teoria que foi utilizada para a elaboração deste trabalho. Primeiramente, serão apresentados os conceitos básicos sobre modelagem de sistemas em espaços de estados; na sequência, conceitos de estabilidade quadrática, desigualdades matriciais lineares, índices de desempenho dados pelas normas \mathcal{H}_2 e \mathcal{H}_∞ , teoria de controle robusto, síntese de controladores por realimentação de estados e por realimentação de saída e por fim, o projeto de controladores por realimentação dinâmica de saída que garante a minimização da norma \mathcal{H}_∞ .

2.1 TEORIA DE CONTROLE MODERNO

Segundo (OGATA, 2010), as tendências modernas em engenharia de sistemas estão na direção de maiores complexidades devido às exigências de desempenho e precisão em tarefas complexas. Dessa forma, a Teoria de Controle Moderno surgiu da necessidade de melhorar o desempenho, aumentar a capacidade de resolver problemas mais complexos e facilitar o acesso para computadores de grande porte.

A Teoria de Controle Moderno distingue-se basicamente da Teoria de Controle Clássico pela representação dos sistemas em termos de suas variáveis de estados, que é uma representação temporal, enquanto as técnicas de Controle Clássico usam a representação dos sistemas em termos de funções de transferência, que é uma representação frequencial.

A modelagem de sistemas na Teoria de Controle Moderno pode ser realizada através do conceito de estados. Portanto, um sistema linearizado invariante no tempo pode ser expresso em espaços de estados por

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{10}$$

sendo $x(t) \in \mathcal{R}^n$ os estados do sistema, $u(t) \in \mathcal{R}^m$ as entradas do sistema, $y(t) \in \mathcal{R}^q$

as saídas do sistema, $A \in \mathcal{R}^{n \times n}$ a matriz de estados, $B \in \mathcal{R}^{n \times m}$ a matriz de entradas, $C \in \mathcal{R}^{q \times n}$ a matriz de saída e $D \in \mathcal{R}^{q \times m}$ a matriz de transmissão direta.

O objetivo geral do controle de sistemas é, dado um estado inicial t_0 , calcular uma entrada de controle $u(t)$ definida para todo $t \geq t_0$ que seja capaz de conduzir os estados ao ponto de equilíbrio $x(t) = 0$ em um período finito de tempo. A possibilidade de calcular tal entrada é a definição de controlabilidade completa no tempo t_0 (AGULHARI, 2013 apud KALMAN et al., 1960).

2.1.1 ESTABILIDADE QUADRÁTICA

Segundo (KHALIL, 2002), a Teoria de Estabilidade exerce um papel central na Teoria de Controle e Engenharia. Existem diversas abordagens de estabilidade que surgem de sistemas dinâmicos.

A estabilidade é caracterizada em termos de *Aleksandr Mikhailovich Lyapunov*, matemático e engenheiro que desenvolveu os primeiros conceitos. Um ponto de equilíbrio é estável se todas as soluções em pontos próximos também permaneçam próximas, caso contrário ele é instável. É assintoticamente estável se todas as soluções que partirem de pontos próximos não só permaneçam próximas, como também levem ao ponto de equilíbrio, quando o tempo tender ao infinito.

Teorema 1 (SILVA et al., 2009) (*Teorema de Lyapunov*)

Seja o sistema $\dot{x}(t) = Ax(t)$, $x(t) = 0$ um ponto de equilíbrio desse sistema, e $V(x(t))$ uma função contínua diferenciável, tal que

$$\begin{aligned} V(x(t)) &> 0 \quad \forall x \neq 0 \in B_x, \\ \dot{V}(x(t)) &< 0 \quad \forall x \neq 0 \in B_x, \end{aligned} \tag{11}$$

sendo B_x uma região de vizinhança do ponto de equilíbrio.

Sendo assim, é possível definir que o ponto de equilíbrio é estável se existir uma função diferenciável definida positiva $V(x(t))$ tal que sua derivada no tempo $\dot{V}(x(t))$ é definida semi-negativa e assintoticamente estável se $\dot{V}(x(t))$ é definida negativa.

A classe de funções de forma quadrática é uma forma de verificar a positividade facilmente. Uma função quadrática é definida por:

$$V(x) = x'(t)Px(t), \tag{12}$$

sendo P uma matriz simétrica definida positiva ($P = P' > 0$). Neste caso, $V(x(t))$ é definida positiva se, e somente se, todos os autovalores de P forem positivos.

2.1.2 DESIGUALDADES MATRICIAIS LINEARES

Uma LMI é uma expressão na forma (SCHERER; WEILAND, 2004)

$$F(x) := F_0 + x_1 F_1 + \dots + x_n F_n < 0, \quad (13)$$

sendo $x = (x_1, \dots, x_n)$ um vetor de números reais chamados variáveis de decisão, F_0, \dots, F_n matrizes reais simétricas e < 0 uma desigualdade.

Portanto, uma LMI é uma desigualdade

$$F(x) < 0, \quad (14)$$

onde F é uma função afim que mapeia um espaço vetorial de dimensão finita \mathcal{X} em matrizes simétricas.

A LMI (14) define uma restrição convexa em x . Ou seja, o conjunto

$$\mathcal{S} := \{x | F(x) < 0\}, \quad (15)$$

de soluções onde a LMI $F(x) < 0$ é convexo.

Também é possível expressar um sistema de LMIs

$$F_1(x) < 0, \dots, F_k(x) < 0, \quad (16)$$

que também pode ser representado na forma

$$F(x) := \begin{bmatrix} F_1(x) & 0 & \dots & 0 \\ 0 & F_2(x) & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & F_k(x) \end{bmatrix} < 0. \quad (17)$$

Há dois problemas genéricos inerentes ao estudo de LMIs:

- **Factibilidade:** A questão da existência ou não de elementos $x \in \mathcal{X}$ que satisfaçam $F(x) < 0$ é chamada de factibilidade;
- **Otimização:** Pode-se definir uma função objetivo $f : \mathcal{S} \rightarrow \mathcal{R}$, sendo $\mathcal{S} \in \{x | F(x) <$

0}. O problema para determinar

$$V_{opt} = \inf_{x \in \mathcal{S}} f(x) \quad (18)$$

é chamado de problema de otimização de uma restrição LMI. Este envolve a determinação de V_{opt} , o cálculo da solução x quase ótima, ou seja, para um $\varepsilon < 0$ arbitrário, o cálculo de um $x \in \mathcal{S}$ tal que $V_{opt} \leq f(x) \leq V_{opt} + \varepsilon$, ou o cálculo das soluções ótimas x_{opt} (elementos $x_{opt} \in \mathcal{S}$ tal que $V_{opt} = f(x_{opt})$).

As LMIs surgem naturalmente no contexto de controle a partir do Teorema de *Lyapunov* (Teorema 1). Aplicando a forma quadrática (12) para sistemas lineares invariantes no tempo, são obtidas as LMIs:

$$\begin{aligned} P = P' &> 0, \\ A'P + PA &< 0. \end{aligned} \quad (19)$$

2.1.3 TEORIA DE CONTROLE ROBUSTO

Levando em consideração que a modelagem matemática do sistema é baseada nas leis físicas, observa-se que esses modelos são normalmente não lineares e possivelmente possuem parâmetros distribuídos, o que os torna muito difíceis de analisar. Sistemas não lineares podem ser aproximados, de forma a simplificar, em modelos lineares. Além disso, essas simplificações podem apresentar margens de erro, que podem apresentar uma dinâmica consideravelmente diferente da dinâmica real do sistema.

Dessa forma, a Teoria de Controle Robusto foi desenvolvida de modo a levar em consideração erros no projeto de modelagem de controle de sistemas. Basicamente, a teoria assume que existem incertezas ou erros entre a planta física e o modelo matemático.

Neste trabalho, supõem-se que as incertezas são limitadas, com limitantes conhecidos a priori. Sendo assim, é possível construir, para cada matriz do sistema, politopos convexos, cuja vantagem principal é a relativa simplicidade de manipulação (BOYD et al., 1994). O sistema incerto é denotado por (AGULHARI, 2013):

$$\mathcal{G}_\alpha(t) = \begin{cases} \dot{x}(t) = A(\alpha)x(t) + B_1(\alpha)w(t) + B_2(\alpha)u(t), \\ z(t) = C_1(\alpha)x(t) + D_1(\alpha)w(t) + D_2(\alpha)u(t), \\ y(t) = C_2(\alpha)x(t) + D_y(\alpha)w(t), \end{cases} \quad (20)$$

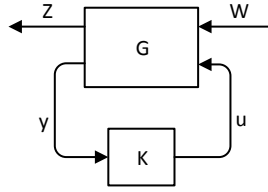


Figura 10: Diagrama de blocos de um sistema com perturbação externa

Fonte: (DOYLE et al., 1989)

sendo $x(t) \in \mathcal{R}^{n_x}$ o vetor de estados, $w(t) \in \mathcal{R}^r$ a entrada exógena, $u(t) \in \mathcal{R}^m$ a entrada de controle, $z(t) \in \mathcal{R}^p$ a saída controlada e $y(t) \in \mathcal{R}^q$ a saída medida, e $\alpha = \{\alpha_1, \dots, \alpha_N\} \in \Lambda_N$ o conjunto de N elementos conhecido como simplex unitário, definido por

$$\Lambda_N = \{\alpha \in \mathcal{R}^N : \sum_{i=1}^N \alpha_i = 1, \alpha_i \geq 0, i = 1, \dots, N\}. \quad (21)$$

Assim, as matrizes do sistema podem ser genericamente descritas por

$$M(\alpha) = \sum_{i=1}^N \alpha_i M_i, \alpha \in \Lambda_N. \quad (22)$$

A matriz politópica $M(\alpha)$ representa quaisquer matrizes de incertezas utilizadas nas equações (20), N é o número de vértices do politopo, $M_i, i = 1, \dots, N$ são os respectivos vértices, ou seja, as matrizes com os valores extremos assumidos pelas incertezas.

Na síntese de controladores, é comum que a estabilização do sistema não seja o único objetivo a ser alcançado. Na maioria dos projetos de controladores, é necessário satisfazer critérios de desempenho com objetivos diversos, como por exemplo a minimização do esforço de controle, energia gasta para executar determinada atividade, entre outros. Neste trabalho, dois critérios de desempenho são abordados: a minimização da norma \mathcal{H}_2 e a minimização da norma \mathcal{H}_∞ .

Considere um sistema linear com perturbação externa (23), representado pela Figura 10:

$$\mathcal{G}_\alpha(t) = \begin{cases} \dot{x}(t) = A(\alpha)x(t) + B_1(\alpha)w(t) + B_2(\alpha)u(t), \\ z(t) = C_1(\alpha)x(t) + D_1(\alpha)w(t) + D_2(\alpha)u(t). \end{cases} \quad (23)$$

Definição 1 (TROFINO et al., 2000) (Norma \mathcal{H}_2) Seja o sistema (23) com a resposta impulsiva $g(t)$, sendo o sistema estritamente próprio e estável, ou seja, com a matriz $A(\alpha)$ estável para qualquer α , $D_1 = 0$ e $u(t) = 0$, então sua norma \mathcal{H}_2 , energia da resposta ao impulso $w(t)$ para a saída $z(t)$, para $-\infty < t < 0$, é denotada por $\|\mathcal{G}_\alpha(t)\|_2$.

$$\|\mathcal{G}_\alpha(t)\|_2 = \sqrt{\int_0^\infty \text{traço}[g(t)'g(t)]dt} \quad (24)$$

A norma \mathcal{H}_2 também pode ser calculada através das LMIs (26) e (28), sendo que W_c e W_o são conhecidos como gramianos de controlabilidade e observabilidade, respectivamente.

$$\|\mathcal{G}_\alpha(t)\|_2 < \text{traço} [C_1(\alpha)W_c(\alpha)C_1(\alpha)'] : \begin{cases} W_c(\alpha) = W_c(\alpha)' > 0 \\ A(\alpha)W_c(\alpha) + W_c(\alpha)A(\alpha)' + B_1(\alpha)B_1(\alpha)' < 0 \end{cases} \quad (26)$$

$$\|\mathcal{G}_\alpha(t)\|_2 < \text{traço} [B_1(\alpha)'W_o(\alpha)B_1(\alpha)] : \begin{cases} W_o(\alpha) = W_o(\alpha)' > 0 \\ A(\alpha)'W_o(\alpha) + W_o(\alpha)A(\alpha) + C_1(\alpha)'C_1(\alpha) < 0 \end{cases} \quad (28)$$

Definição 2 (TROFINO et al., 2005) (Norma \mathcal{H}_∞) Seja o sistema (23) com $u(t) = 0$, então sua norma \mathcal{H}_∞ , valor máximo resultante da relação entre a energia dos sinais de saída $z(t)$ e de entrada $w(t)$, para todo $w(t)$ de energia limitada e para todo α , é denotada por $\|\mathcal{G}_\alpha(t)\|_\infty$

$$\|\mathcal{G}_\alpha(t)\|_\infty = \sup_{w \in \mathcal{L}_2} \frac{\|z(t)\|}{\|w(t)\|} \quad \forall \alpha \in \Lambda_N, \quad (29)$$

sendo \mathcal{L}_2 a função vetorial quadraticamente integrável no intervalo $[0, \infty]$.

Note que a norma \mathcal{H}_∞ indica o ganho máximo entre a entrada e a saída. Portanto, minimizar a norma \mathcal{H}_∞ também assegura uma maior robustez do controlador.

O Lema 1 (Bounded Real Lemma) representa uma condição LMI cujo objetivo é a minimização da norma \mathcal{H}_∞ .

Lema 1 (*Bounded Real Lemma*) (*BOYD et al., 1994*)

Considere o sistema (20) com $u(t) = 0$. A norma \mathcal{H}_∞ do sistema, denotada por $\|\mathcal{G}_\alpha(t)\|_\infty$, é limitada superiormente por um escalar positivo γ se, e somente se, existir uma matriz $P(\alpha) = P'(\alpha) > 0$ que verifique¹

$$\begin{bmatrix} A'(\alpha)P(\alpha) + P(\alpha)A(\alpha) & P(\alpha)B_1(\alpha) & C_1'(\alpha) \\ \star & -\gamma^2 I & D_1'(\alpha) \\ \star & \star & -I \end{bmatrix} < 0. \quad (30)$$

2.1.4 SÍNTESE DE CONTROLADORES

A síntese de controladores na representação de sistemas em espaço de estados pode ser realizada através de controladores estáticos (por realimentação de estados ou de saídas) ou controladores dinâmicos (por realimentação de saída ou por meio de observadores). Os controladores são projetados com diversos propósitos, por exemplo, estabilidade em malha fechada, rejeição de distúrbios, rastreamento do modelo, desacoplamento de entradas e saídas, além de satisfazer critérios de desempenho, como, por exemplo, a minimização das normas \mathcal{H}_2 e \mathcal{H}_∞ .

Considerando o sistema linear (20), um controlador por realimentação estática de estados (SSF, acrônimo em inglês para *Static State Feedback*) pode ser projetado supondo que $u(t) = K_{sf}(\alpha)x(t)$. A entrada de controle é uma função linear do estado, ou seja, uma realimentação de estados linear ou constante, sendo que a matriz $K_{sf}(\alpha)$ é chamada de ganho de realimentação de estados.

Substituindo $u(t)$ em (20), obtêm-se

$$\mathcal{G}_\alpha^{sf}(t) = \begin{cases} \dot{x}(t) = (A(\alpha) + B_2(\alpha)K_{sf}(\alpha))x(t) + B_1(\alpha)w(t), \\ z(t) = (C_1(\alpha) + D_2(\alpha)K_{sf}(\alpha))x(t) + D_1(\alpha)w(t). \end{cases} \quad (31)$$

O projeto do controlador SSF consiste na definição do ganho de realimentação de estados (K_{sf}) de modo que o sistema de malha fechada (31) satisfaça determinadas propriedades ou especificações de projeto, por exemplo a estabilidade. Diversas técnicas podem ser utilizadas na definição dos ganhos K_{sf} , como por exemplo alocação de polos e o Teorema 3, visto mais adiante.

¹O símbolo \star representa blocos simétricos nas LMIs

Um controlador por realimentação estática de estados utilizando como critério de desempenho a minimização da norma \mathcal{H}_2 pode ser representado pelo Teorema 2.

Teorema 2 (TROFINO et al., 2000) (Controle \mathcal{H}_2 - Realimentação de Estados) *Existe um controlador de realimentação de estados que minimiza a norma \mathcal{H}_2 do sistema (23) se, e somente se, existirem matrizes $P(\alpha) = P(\alpha)' > 0$, $W(\alpha)$ e $Y(\alpha)$, com $D_1(\alpha) = 0$ que soluciona o problema:*

$$\min(\text{traço}(W)) : \begin{cases} \begin{bmatrix} W(\alpha) & (C_1(\alpha)P(\alpha) + D_1(\alpha)Y(\alpha)) \\ (P(\alpha)C_1(\alpha)' + Y(\alpha)'D_1(\alpha)') & P(\alpha) \end{bmatrix} \geq 0 \\ \begin{bmatrix} (A(\alpha)P(\alpha) + P(\alpha)A(\alpha)' + B_2(\alpha)Y(\alpha) + Y(\alpha)'B_2(\alpha)') & B_1(\alpha) \\ B_1(\alpha)' & -I \end{bmatrix} < 0 \end{cases} \quad (32)$$

Em caso afirmativo, $\tilde{K}_{sf}(\alpha) = Y(\alpha)P(\alpha)^{-1}$ é o ganho de realimentação de estados que estabiliza o sistema (23) e minimiza a norma \mathcal{H}_2 . Para prova deste teorema, o leitor é referenciado para (TROFINO et al., 2000).

Para que seja possível implementar um controlador por realimentação de estados, é necessário que seja possível aferir todas as variáveis de estados. Se essa condição não puder ser satisfeita, outros controladores podem ser utilizados, como por exemplo o controle por realimentação de saída.

O controlador por realimentação de saída pode ser basicamente classificado em estático (SOF) e dinâmico (DOF). A estrutura básica de um controlador SOF é dada por $u(t) = K_{sof}y(t)$, sendo K_{sof} um vetor constante, enquanto que a estrutura de um controlador DOF é definida por

$$\begin{aligned} \dot{x}_c(t) &= A_c x_c(t) + B_c y(t), \\ u(t) &= C_c x_c(t) + D_c y(t), \end{aligned} \quad (33)$$

sendo $x_c(t) \in \mathcal{R}^{n_c}$ os estados e n_c a ordem do controlador. Note que se $n_c = 0$, então o controlador é equivalente a um SOF.

Considere agora o sistema linear contínuo expresso pela equação (20). Denotando

$$\tilde{K}_{dof} = \begin{bmatrix} A_c & B_c \\ C_c & D_c \end{bmatrix}, \eta(t) = \begin{bmatrix} x(t) \\ x_c(t) \end{bmatrix}, \quad (34)$$

então o projeto do controlador robusto de ordem reduzida pode ser moldado na síntese de um ganho de realimentação estático de saída $\tilde{K}_{dof} \in \mathcal{R}^{(m+n_c) \times (n_c+q)}$ para o sistema aumentado

$$\begin{cases} \dot{\eta}(t) = \tilde{A}(\alpha)\eta(t) + \tilde{B}_1(\alpha)w(t) + \tilde{B}_2(\alpha)u(t), \\ z(t) = \tilde{C}_1(\alpha)\eta(t) + \tilde{D}_1(\alpha)w(t) + \tilde{D}_2(\alpha)u(t), \\ y(t) = \tilde{C}_2(\alpha)\eta(t) + \tilde{D}_y(\alpha)w(t), \end{cases} \quad (35)$$

com

$$\begin{aligned} \tilde{A}(\alpha) &= \begin{bmatrix} A(\alpha) & 0 \\ 0 & 0_{n_c} \end{bmatrix}, & \tilde{B}_1(\alpha) &= \begin{bmatrix} B_1(\alpha) \\ 0 \end{bmatrix}, \\ \tilde{B}_2(\alpha) &= \begin{bmatrix} 0 & B_2(\alpha) \\ I_{n_c} & 0 \end{bmatrix}, & \tilde{C}_1(\alpha) &= [C_1(\alpha) \quad 0], \\ \tilde{D}_1(\alpha) &= [D_1(\alpha)], & \tilde{D}_2(\alpha) &= [0 \quad D_2(\alpha)], \\ \tilde{C}_2(\alpha) &= \begin{bmatrix} 0 & I_{n_c} \\ C_2(\alpha) & 0 \end{bmatrix}, & \tilde{D}_y(\alpha) &= \begin{bmatrix} 0 \\ D_y(\alpha) \end{bmatrix}. \end{aligned}$$

Mas nem sempre é necessário que o controlador tenha a ordem do sistema, ou seja $n_c = n_x$. Controladores de ordem reduzida são aqueles que têm uma dimensão menor que o sistema, $n_c < n_x$.

As matrizes em malha fechada do sistema (35) são definidas por

$$\tilde{A}_{cl}(\alpha) = \tilde{A}(\alpha) + \tilde{B}_2(\alpha)\tilde{K}_{dof}\tilde{C}_2(\alpha), \quad (40)$$

$$\tilde{B}_{cl}(\alpha) = \tilde{B}_1(\alpha) + \tilde{B}_2(\alpha)\tilde{K}_{dof}\tilde{D}_y(\alpha), \quad (41)$$

$$\tilde{C}_{cl}(\alpha) = \tilde{C}_1(\alpha) + \tilde{D}_2(\alpha)\tilde{K}_{dof}\tilde{C}_2(\alpha), \quad (42)$$

$$\tilde{D}_{cl}(\alpha) = \tilde{D}_1(\alpha) + \tilde{D}_2(\alpha)\tilde{K}_{dof}\tilde{D}_y(\alpha). \quad (43)$$

e, para um $\alpha \in \Lambda_N$ fixo, a função de transferência de w a z é

$$H(s, \alpha) = \tilde{C}_{cl}(\alpha)(sI - \tilde{A}_{cl}(\alpha))^{-1}\tilde{B}_{cl}(\alpha) + \tilde{D}_{cl}(\alpha) \quad (44)$$

□

O parâmetro escalar ξ pode ser usado como um grau extra de liberdade para produzir distintos ganhos de realimentação de estado que estabilizem o sistema.

O Teorema 4 é a condição LMI proposta por (AGULHARI et al., 2011), que corresponde ao segundo estágio. O Teorema 4 gera uma condição suficiente para a existência de um controlador de realimentação de saída dinâmico de grau arbitrário. É esperado que o parâmetro dependente que estabiliza o controlador por realimentação de estados $\tilde{K}_{sf}(\alpha)$ já tenha sido obtido no primeiro estágio.

Teorema 4 (AGULHARI et al., 2011) *Seja $\tilde{K}_{sf}(\alpha)$ o controlador de realimentação de estados que estabiliza o sistema. Se existirem as matrizes $P(\alpha) = P(\alpha)' > 0$, $F(\alpha)$, $V(\alpha)$, $H(\alpha)$, R e L e um escalar $\gamma_{dof} > 0$ que, $\forall \alpha \in \Lambda_N$, satisfaça*

$$\begin{bmatrix} \Upsilon_{11}(\alpha) & \Upsilon_{12}(\alpha) & \Upsilon_{13}(\alpha) & \Upsilon_{14}(\alpha) & \Upsilon_{15}(\alpha) \\ * & -V(\alpha) - V(\alpha)' & V(\alpha)\tilde{B}_1(\alpha) & 0 & V(\alpha)\tilde{B}_2(\alpha) \\ * & * & -\gamma_{dof}^2 I & \tilde{D}_1(\alpha)'H(\alpha) & \tilde{D}_y(\alpha)'L' \\ * & * & * & I - H(\alpha) - H(\alpha)' & H(\alpha)'\tilde{D}_2(\alpha) \\ * & * & * & * & -R - R' \end{bmatrix} < 0, \quad (50)$$

com,

$$\begin{aligned} \Upsilon_{11}(\alpha) &= (\tilde{A}(\alpha) + \tilde{B}_2(\alpha)\tilde{K}_{sf}(\alpha))'F(\alpha)' \\ &\quad + F(\alpha)(\tilde{A}(\alpha) + \tilde{B}_2(\alpha)\tilde{K}_{sf}(\alpha)), \\ \Upsilon_{12}(\alpha) &= P(\alpha) - F(\alpha) + (\tilde{A}(\alpha) + \tilde{B}_2(\alpha)\tilde{K}_{sf}(\alpha))'V(\alpha)', \\ \Upsilon_{13}(\alpha) &= F(\alpha)\tilde{B}_1(\alpha), \\ \Upsilon_{14}(\alpha) &= \tilde{C}_1(\alpha)'H(\alpha) + \tilde{K}_{sf}(\alpha)'\tilde{D}_2(\alpha)'H(\alpha), \\ \Upsilon_{15}(\alpha) &= F(\alpha)\tilde{B}_2(\alpha) + \tilde{C}_2(\alpha)'L' - \tilde{K}_{sf}(\alpha)'R', \end{aligned}$$

então $\tilde{K}_{dof} = R^{-1}L$ é o ganho do controlador robusto de realimentação dinâmico de saída que estabiliza o sistema (35) com uma norma \mathcal{H}_∞ garantida por γ_{dof} .

Prova: Se a desigualdade em (50) é satisfeita, então também é possível verificar com $-H(\alpha)'H(\alpha)$ substituindo $I - H(\alpha) - H(\alpha)'$, pois $(I - H(\alpha))(I - H(\alpha)) \geq 0$ implica $-H(\alpha)'H(\alpha) \leq I - H(\alpha) - H(\alpha)'$. Multiplicando a condição resultante por $T_1(\alpha)$ pela

esquerda e $T_1(\alpha)'$ pela direita, sendo

$$T_1(\alpha) = \begin{bmatrix} I & 0 & 0 & 0 & S_1(\alpha)' \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & S_2(\alpha)' \\ 0 & 0 & 0 & I & 0 \end{bmatrix}, \quad (51)$$

e

$$S_1(\alpha) = R^{-1}L\tilde{C}_2(\alpha) - \tilde{K}_{sf}(\alpha), \quad S_2(\alpha) = R^{-1}L\tilde{D}_y(\alpha) \quad (52)$$

tem-se

$$\begin{bmatrix} F(\alpha)\tilde{A}_{cl}(\alpha) + \tilde{A}_{cl}(\alpha)'F(\alpha)' & P(\alpha) - F(\alpha) + \tilde{A}_{cl}(\alpha)'V(\alpha)' \\ \star & -V(\alpha) - V(\alpha)' \\ \star & \star \\ \star & \star \\ & F(\alpha)\tilde{B}_{cl}(\alpha) & \tilde{C}_{cl}(\alpha)'H(\alpha) \\ & V(\alpha)\tilde{B}_{cl}(\alpha) & 0 \\ & -\gamma_{dof}^2 I & \tilde{D}'_{cl}H(\alpha) \\ & \star & -H(\alpha)'H(\alpha) \end{bmatrix} < 0. \quad (54)$$

Pré-multiplicando (54) por $T_2(\alpha)'$ e pós-multiplicando por $T_2(\alpha)$, sendo

$$T_2(\alpha) = \begin{bmatrix} I & 0 & 0 \\ \tilde{A}_{cl}(\alpha) & \tilde{B}_{cl}(\alpha) & 0 \\ 0 & I & 0 \\ 0 & 0 & H(\alpha)^{-1} \end{bmatrix}, \quad (55)$$

produz

$$\begin{bmatrix} P(\alpha)\tilde{A}_{cl}(\alpha) + \tilde{A}_{cl}(\alpha)'P(\alpha) & P(\alpha)\tilde{B}_{cl}(\alpha) & \tilde{C}_{cl}(\alpha)' \\ \star & -\gamma_{dof}^2 I & \tilde{D}_{cl}(\alpha)' \\ \star & \star & -I \end{bmatrix} < 0, \quad (56)$$

que é a versão apresentada no Lema 1, considerando o sistema em malha fechada. Portanto, o ganho de realimentação de saída robusto $K_{dof} = R^{-1}L$ estabiliza o sistema em malha fechada (40) com $\|H(s, \alpha)\|_\infty < \gamma_{dof}$, para todo $\alpha \in \Lambda_N$.

É importante enfatizar que qualquer ganho de realimentação de estados dependente de parâmetros ($\tilde{K}_{sf}(\alpha)$) que estabilize o sistema proposto pode também ser usado como

entrada no segundo estágio (50). Além disso, uma vez que no Teorema 3 o ganho que estabiliza o sistema é $\tilde{K}_{sf}(\alpha) = Z(\alpha)G(\alpha)^{-1}$, uma condição similar pode ser encontrada através das matrizes dependentes de parâmetros $Z(\alpha)$ e $G(\alpha)$.

A técnica de dois estágios é uma abordagem interessante para a síntese de controladores de realimentação de saída, e bons resultados foram relatados na literatura (ARZELIER et al., 2003; MEHDI et al., 2004; ARZELIER et al., 2009, 2010; AGULHARI et al., 2010, 2011). Contudo, os limitantes de γ_{dof} para a norma \mathcal{H}_∞ do sistema controlado são altamente dependentes da escolha do ganho de realimentação de estados $\tilde{K}_{sf}(\alpha)$ selecionado no primeiro estágio, e a busca por um bom ganho ainda é um problema que permanece em aberto. Algumas técnicas para determinar um bom ganho de realimentação de estados no primeiro estágio foi proposta em (ARZELIER et al., 2009, 2010), e melhorias locais iterando o primeiro e segundo estágio também foram apresentados em (AGULHARI et al., 2011).

No próximo capítulo, serão apresentadas duas estratégias inteligentes para substituir o primeiro estágio e encontrar o ganho de realimentação de estados dependente de parâmetros $\tilde{K}_{sf}(\alpha)$ que permita alcançar a menor norma \mathcal{H}_∞ no segundo estágio.

3 METAHEURÍSTICAS DE OTIMIZAÇÃO

Durante o projeto, construção e manutenção de sistemas de engenharia, várias decisões tecnológicas e gerenciais são tomadas por engenheiros em diferentes etapas. O objetivo geral de toda essa cadeia de decisões normalmente está em minimizar o esforço necessário ou maximizar o benefício desejado, sendo que diversas restrições podem preencher esses dois extremos. Segundo (RAO, 2009) a Otimização é o ato de obter os melhores resultados, satisfazendo condições pré-definidas. De uma forma prática, otimização pode ser definida como a busca pelo valor mínimo ou máximo de uma função $f(x)$ e pode ser definida pelo problema (58).

$$\text{Obter } X = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{Bmatrix}, \text{ que minimiza } f(x) \text{ e satisfaça } \begin{cases} g_1(X) \leq 0, \\ g_2(X) \geq 0, \\ \vdots \\ g_M(X) = 0 \end{cases}, \quad (58)$$

sendo N o número de variáveis do problema, M o número de restrições para o problema, $X \in \mathcal{R}^N$ o vetor de solução ótimo, $g_j(X)$, $j = 1, \dots, M$ as restrições propostas e $f(X)$ a função de custo que se deseja minimizar ou maximizar.

Resumidamente, um procedimento de otimização visa encontrar uma ou mais soluções dentre as diversas soluções que podem satisfazer as restrições propostas, classificando-as através de seus custos.

Contudo, segundo (TALBI, 2009), a otimização de muitos problemas de engenharia podem não ser tratáveis e, na prática, soluções consideradas “boas” podem ser obtidas através de heurísticas ou metaheurísticas. Metaheurísticas representam uma família de técnicas de otimização que podem aproximar-se dos resultados ótimos com soluções aceitáveis em tempo razoável e vêm adquirindo um crescente interesse de pesquisadores.

Dentre os algoritmos metaheurísticos de otimização, encontram-se as metaheu-

rísticas populacionais, inspiradas em observações/fenômenos da natureza, que visam, particularmente, o aperfeiçoamento de uma dada população de soluções de maneira iterativa.

A estrutura geral de um algoritmo pertencente a esta classe consiste nos seguintes passos: inicializar uma população de soluções (soluções parciais), gerar uma nova população de soluções com base em informações armazenadas no decorrer do algoritmo, e então por simples procedimentos de seleção, integrá-las com as soluções inicialmente propostas, até que um critério de parada seja satisfeito, e uma solução ótima possa ser identificada (TALBI, 2009).

Basicamente, os tópicos de maior relevância para a compreensão dessa classe são os seguintes (TALBI, 2009):

- População Inicial: normalmente é inicializada de maneira aleatória - uniformemente distribuída em um intervalo de valores. Esta população deve apresentar valores diversificados, caso contrário há risco de uma convergência prematura;
- Memória de Busca: conjunto de informações extraídas e memorizadas durante o processo de busca de uma dada metaheurística;
- Geração: consiste em gerar novas populações, segundo um conjunto de regras (heurísticas) de um dado método;
- Seleção: é um processo de comparação das soluções recém-geradas com as soluções de uma população, em uma determinada iteração, visando atualizar um conjunto de soluções para as próximas iterações;
- Critério de Parada: procedimento que verifica a evolução do algoritmo. Consiste em encerrar/parar o algoritmo quando a medida de diversidade é baixa (aproximadamente constante), ou segundo uma quantidade de simulações ou iterações.

Um dos principais problemas das técnicas apresentadas no subcapítulo 2.2 é a alta carga computacional envolvida, uma vez que o incremento de cada iteração das variáveis (assim como a ordem do controlador) e também a precisão desejada podem criar uma superfície ampla para ser avaliada, o que implica em um grande tempo computacional. O objetivo deste trabalho é propor e comparar metaheurísticas para melhorar este processo iterativo, sendo estas técnicas empregadas substituindo o primeiro estágio, encontrando os ganhos de realimentação de estados que resultem em controladores de realimentação

de saída ótimos. A principal vantagem de utilizar métodos inteligentes é que a busca por soluções ótimas pode cobrir toda a superfície de busca, porém sem computar todas as possibilidades, o que pode diminuir o tempo computacional. O algoritmo aprende com os resultados e estes direcionam aos caminhos que levarão à melhor solução.

Com o objetivo de alcançar ganhos robustos de realimentação de saída com ordem reduzida que minimizam os limitantes da norma \mathcal{H}_∞ , são propostas implementações de GA e PSO com o objetivo de buscar a solução ótima.

3.1 ALGORITMOS GENÉTICOS

Assim como outras metaheurísticas que são inspiradas em comportamentos naturais, o GA é uma técnica de otimização e busca baseada nos princípios de genética natural e teoria Darwiniana.

O método foi desenvolvido por HOLLAND nos anos 1960 e 1970 e foi popularizado com (GOLDBERG, 1989; HAUPT; HAUPT, 2004).

Algumas das vantagens sobre outros métodos inteligentes e numéricos são que os GAs:

- são capazes de otimizar variáveis contínuas e discretas;
- não necessitam de informação derivativa;
- podem sair de um mínimo local;
- produzem uma lista de valores ótimos, não apenas um único.

Há duas abordagens na implementação de GAs: GAs Binários e GAs Contínuos. A diferença básica entre eles é que nos GAs Binários as variáveis contínuas são convertidas em *strings* de suas representações binárias, enquanto que nos GAs Contínuos nenhuma conversão é necessária.

A estrutura geral de um GA consiste nos seguintes passos: inicializar uma população de soluções (cromossomos), calcular o custo de cada cromossomo, selecionar os melhores cromossomos da população, selecionar e realizar cruzamentos entre os cromossomos selecionados e realizar mutações em uma parcela da nova população, até que um critério de parada seja satisfeito, e uma solução ótima possa ser identificada. Este fluxograma é apresentado na Figura 12 e explanado a seguir.

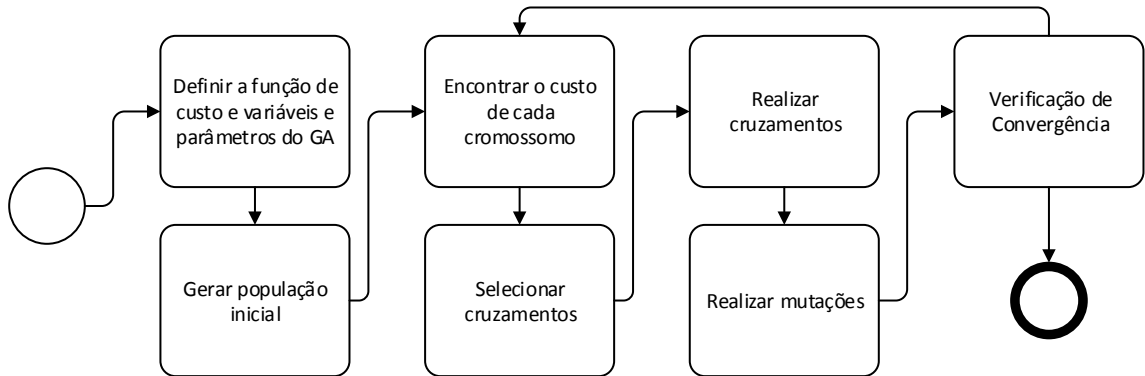


Figura 12: Fluxograma de um Algoritmo Genético Contínuo

Fonte: Adaptado de (HAUPT; HAUPT, 2004)

Para iniciar, uma população inicial de cromossomos com tamanho N_{pop} é necessária. Estes dados podem ser tanto fornecidos como podem ser gerados aleatoriamente. Esta primeira população é então avaliada pela função objetivo que produzirá o custo de cada cromossomo na população. O custo é então usado para gerar uma ordenação ascendente.

A função objetivo é usada para gerar a informação que será usada para determinar quais cromossomos resolvem melhor o problema proposto. A função objetivo pode não somente ser uma fórmula matemática, ela também pode englobar procedimentos complexos e recursos de terceiros. Contudo, é necessário que sejam retornados dados numéricos, chamados de custo, que façam sentido para ordenação.

Os seguintes passos são repetidos até que o critério de parada seja alcançado.

A população é então selecionada usando o um parâmetro N_{kept} que indica o percentual de cromossomos que serão selecionados, os que possuem os melhores custos. Esta etapa é denominada seleção natural.

Uma parcela da população remanescente é então escolhida aleatoriamente utilizando probabilidades de que privilegiem os melhores posicionados para cruzarem e gerarem novos descendentes. Cada par gera dois novos cromossomos usando a equação de (RADCLIFFE, 1991) definida por

$$\begin{aligned}
 p_{new}(1) &= \beta \times p_{mn} + (1 - \beta) \times p_{dn}, \\
 p_{new}(2) &= \beta \times p_{dn} + (1 - \beta) \times p_{mn},
 \end{aligned}
 \tag{59}$$

sendo β um número escalar aleatório no intervalo $[0,1]$ que representa o quanto cada um dos pais influenciará cada novo cromossomo, p_{mn} é $n^{\text{ésima}}$ variável no cromossomo mãe e p_{dn} é $n^{\text{ésima}}$ variável no cromossomo pai.

Agora que a população está novamente preenchida, são selecionados quais cromossomos e genes sofrerão mutação, usando uma taxa de mutação dinâmica $Pb_{mutation}$, como proposto em (DEIF et al., 2012) pela equação (60), com o objetivo de selecionar os genes e cromossomos que sofrerão mutação. Esta etapa é responsável por adicionar novas informações na população, que talvez não sejam produzidas apenas com o cruzamento. Normalmente é a principal responsável pela saída do GA de mínimos locais.

$$DL = \frac{\text{melhor custo da população}}{\text{média de custo da população}}, \quad (60)$$

$$Pb_{mutation} = \frac{e^{1-DL}}{1 + e^{1-DL}}.$$

O uso de uma taxa de mutação dinâmica permite evitar a pré-configuração de um parâmetro fixo, que exige um conhecimento prévio do comportamento do sistema durante o processamento do GA. Além disso, ele permite que o GA execute a mutação uma quantidade maior de cromossomos, e isso permite que o algoritmo alcance a convergência mais rapidamente.

Para cada gene de cada cromossomo selecionado, é adicionado um ruído respeitando suas grandezas. O cromossomo melhor posicionado não é afetado por essas mutações.

A nova população é avaliada pela função objetivo e os seus custos são ordenados.

Esses passos são executados até que o critério de parada seja alcançado, ou seja, alguma análise que determine quando o resultado alcançado é satisfatório e deve encerrar a execução do algoritmo.

3.2 OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

Assim como outras metaheurísticas baseadas em população, alguns algoritmos são baseados no que pode ser visto na natureza. Esses métodos visam simplesmente a melhoria da situação do próprio indivíduo ou do coletivo. A efetividade desses métodos pode ser provada através da persistência de espécies que ainda os praticam.

O PSO, assim como os GAs, é um modelo matemático de otimização que toma como ponto de partida os comportamentos biológicos, e usa metáforas e termos originados

da genética, etologia e também da psicologia. Basicamente, em sua versão histórica, PSO é um método coletivo, anárquico e iterativo, com ênfase na cooperação. É parcialmente aleatório e não há seleção.

De acordo com (CLERC, 2006), para compensar limitações de gradiente de estratégia primitivos, seria necessário deteriorar a situação por um longo tempo, e seguir na eventualidade de alguma melhora. Esse comportamento pode ser suicida, por esse motivo é recomendado lembrar a melhor posição já encontrada, para que seja possível retornar, caso necessário. O indivíduo é também capaz de explorar bases de regras mais amplas, aleatoriamente ou inteligentemente, guiados gradualmente pelo conhecimento adquirido.

Um terreno interessante corresponde a um mínimo local de uma certa função definida no universo de discurso. Essa função pode ser definida por uma fórmula matemática, um algoritmo, ou ainda um processo externo, real ou simulado. O PSO pode também ser implementado para buscar terrenos interessantes ou o mais interessante - o mínimo global.

O PSO pode ser descrito como uma metáfora de um enxame de abelhas. Cada partícula do enxame pode ser abstraída como uma abelha, e essas abelhas não carregam apenas néctar para a colmeia, elas também carregam informações. Como um exemplo de informação, uma abelha trabalhadora pode informar a localização de um bom terreno e então rapidamente as outras abelhas irão se direcionar em direção a este. Contudo, como um conhecimento coletivo do enxame, cada abelha informa os seus resultados e elas gradualmente convergem para a localização desejada, agregando todos os conhecimentos individuais e o conhecimento coletivo.

Inicialmente, o algoritmo inicializa uma população ou enxame com N_{swarm} indivíduos/partículas. Cada partícula possui uma posição: um vetor solução com N variáveis, que podem ser geradas de maneira aleatória dentro de um intervalo estabelecido, ou através de alguma heurística. Embora o número de abelhas em um enxame seja realmente elevado, em PSO, experimentos empiricamente propõem tamanhos em torno de 20 a 40 partículas (CLERC, 2006). O tamanho do enxame também pode ser dinâmico.

Estes indivíduos também possuem uma velocidade, inicialmente com o valor zero. Em um dado instante/iteração cada partícula no espaço de busca é representada por um vetor x_d , e elas se movem em uma velocidade v_d , influenciada pela melhor posição conhecida por cada partícula p_d e também pela melhor posição global g_d , descrita na equação (61) (CLERC, 2006). Este processo se repete até que o critério de parada seja satisfeito.

$$\begin{aligned}
v_d &= c_1 \times v_d + c_{max} \times \beta \times (p_d - x_d) + c_{max} \times \beta \times (g_d - x_d), \\
x_d &= x_d + v_d,
\end{aligned} \tag{61}$$

sendo β a representação de um número real aleatório gerado no intervalo $[0, 1]$, c_1 representa a confiança no próprio movimento e c_{max} representa a confiança nos seus informantes.

Sendo assim, cada um dos vetores solução são avaliados na função objetivo, visando a obtenção do melhor indivíduo do grupo. Em seguida, as posições das partículas são armazenadas. Logo após, o PSO evolui a posição e a velocidade de cada uma de suas partículas de acordo com a equação (61).

Depois destas atualizações, todas as partículas são avaliadas na função objetivo. Em seguida, um cálculo é realizado visando atualizar as melhores posições de cada partícula p_d , ou seja, a melhor posição já alcançada por cada partícula é mantida com o objetivo de influenciá-la nesta direção nas próximas iterações. Após a verificação das melhores posições individuais de cada partícula, o mínimo global g_d é identificado, tal que se o custo de uma dada partícula for menor que o mínimo global obtido na iteração anterior, o vetor posição desta partícula torna-se o novo mínimo global. Na sequência, o algoritmo continua com o seu processo iterativo, até que uma dada condição de parada seja satisfeita.

Durante a evolução do enxame, é possível que as partículas movam para fora do espaço de busca, o que não quebra o processo, contudo pode torná-lo lento. Para evitar esse cenário, é possível confinar as partículas dentro do espaço de busca. Como exemplo, a equação (62) pode ser utilizada.

$$x_d = \min(\max(x_d + v_d, x_{min}), x_{max}). \tag{62}$$

A Figura 13 mostra os componentes fundamentais para o cálculo do próximo deslocamento de uma partícula: de acordo com sua própria velocidade, em direção ao seu melhor desempenho p_d e em direção ao melhor desempenho do seu melhor informante g_d . A combinação linear dos três vetores com coeficientes de confiança são a base de todas as versões dos PSOs clássicos. As três setas em cinza representam essa combinação dos vetores e indicam a próxima posição da partícula.

De acordo com (KAWAKAMI; MENG, 2009), a constante c_1 representa um coeficiente de inércia. Se $c_1 > 1$, a partícula pode se impulsionar e o algoritmo não irá convergir para uma solução. Dessa forma, c_1 deve ser menor que 1 para que as partículas venham a convergir para $(g_d + p_d)/2$, como resultado do movimento de amortecimento. Um

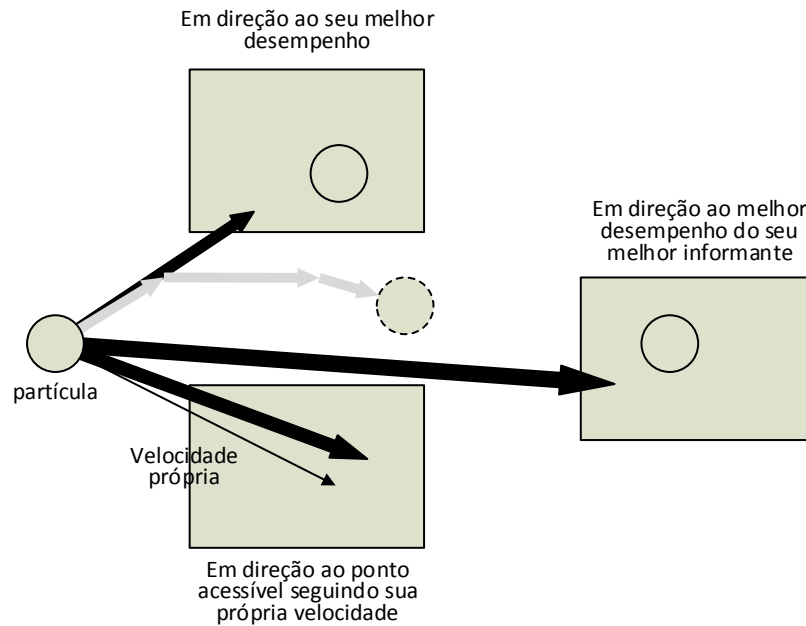


Figura 13: Elementos fundamentais para o cálculo do próximo deslocamento de uma partícula

Fonte: Adaptado de (CLERC, 2006)

algoritmo PSO com um c_1 grande possui vantagem na recuperação de um mínimo local e desvantagem na taxa de convergência. Por outro lado, c_{max} representa um coeficiente gravitacional, atraindo as partículas para g_d ou p_d , com o objetivo de melhorar a eficiência de busca. O algoritmo também pode não convergir se c_{max} for muito grande. Dessa forma, é possível concluir que a alta performance pode ser obtida quando os parâmetros de inércia e gravitação trabalharem em cooperação. Há recomendações para se utilizar os valores de c_1 e c_{max} em pares, por exemplo "0,7 e 1,47" e "0,8 e 1,62" (CLERC, 2006).

Conforme (ALFI, 2012), introduzir o mesmo peso inercial para todas as partículas pode não simular precisamente um modelo biológico por não levar em consideração as diferenças entre as partículas. Para superar essa desvantagem, são propostos Pesos Inerciais Dinâmicos para a equação (61), apresentados na equação (63).

$$\begin{aligned} v_d &= w_d \times v_d + c_{max} \times \beta \times (p_d - x_d) + c_{max} \times \beta \times (g_d - x_d), \\ w_d &= 0,5 \times (1 + \tanh(\alpha \times F(p_d))) + c_{max} \times \beta \times (p_d - x_d), \end{aligned} \quad (63)$$

sendo w_d um vetor de pesos inerciais das partículas na iteração, $\tanh(\cdot)$ uma abreviação da tangente hiperbólica. O parâmetro α deve ser informado previamente e precisa ser

pequeno.

A dificuldade na busca do mínimo global é um comportamento que está diretamente ligado às constantes de confiança, à função objetivo e também ao sistema otimizado. É comum que o algoritmo termine convergindo para um mínimo local ao invés de um mínimo global em superfícies que tenham muitos mínimos locais. Nesses casos, apenas a calibragem das constantes de confiança pode não ser suficiente. Uma solução denominada *Refresh Operator*, é proposta em (KAWAKAMI; MENG, 2009), onde basicamente uma realocação da partícula em uma posição aleatória quando $|v_i| < \xi$, sendo ξ um parâmetro que determina a tolerância de inércia da partícula. Dessa forma, é possível que a partícula tenha uma nova trajetória, seguindo as influências das outras partículas quanto ao mínimo local, e encontre outras regiões de mínimos.

No próximo capítulo, serão apresentadas as estratégias de implementação do GA e PSO no contexto de solucionar os problemas propostos.

4 PROPOSTA DE METAHEURÍSTICAS PARA O CONTROLE \mathcal{H}_∞

A contribuição principal deste trabalho é a proposta de duas novas técnicas baseadas em metaheurísticas de otimização, o GA e o PSO, com o objetivo de substituir o Teorema 3 na formulação de uma proposta de otimização dos limitantes da norma \mathcal{H}_∞ através de controladores por realimentação de saída dinâmica de ordem reduzida. Essa contribuição visa principalmente otimizar sistemas incertos, onde não foram encontradas muitas abordagens disponíveis na literatura.

Os algoritmos do GA e do PSO foram implementados usando as orientações apresentadas no Capítulo 3, e a função objetivo considerada para ambos os algoritmos foi o valor de γ_{dof} obtido a partir do Teorema 4. Nos casos em que a LMI do Teorema 4 não é factível, o retorno da função objetivo é definido como ∞ . As Figuras 15 e 16 mostram as implementações da técnica de síntese de controladores em conjunto com o Algoritmo Genético e também com o PSO. As rotinas foram implementadas usando Matlab® 8.1.0.604 (R2013a), com os pacotes Yalmip (LÖFBERG, 2004), SeDuMi (STURM, 1999) e *Robust LMI Parser* (AGULHARI et al., 2012).

As populações iniciais (para o GA) e os enxames iniciais (para o PSO) foram criados a partir de uma mescla de duas heurísticas: a) ganhos de realimentação de estados $\tilde{K}_{sf}(\alpha)$ que garantam um limitante superior $\mu = \phi + \mu^*$ para a norma \mathcal{H}_2 do sistema, utilizando o Teorema 2, sendo ϕ uma variável aleatória uniformemente distribuída no intervalo $[0, 50]$ e μ^* o limitante ótimo da norma \mathcal{H}_2 do sistema, e b) ganhos de realimentação de estado $\tilde{K}_{sf}(\alpha)$ que estabilizam o sistema, usando o Matlab® *Pole Placement Design (place)*. A utilização de duas heurísticas permite que uma parcela da população inicial seja válida, em termos de factibilidade do segundo estágio, e também a aleatoriedade na alocação da população, que é fundamental para uma boa performance tanto do GA, quanto do PSO. Nos casos em que a heurística a) não consegue encontrar ganhos válidos para o segundo estágio, apenas a heurística b) é utilizada.

Os parâmetros utilizados para o PSO foram $N_{swarm} = 20$ e os parâmetros de

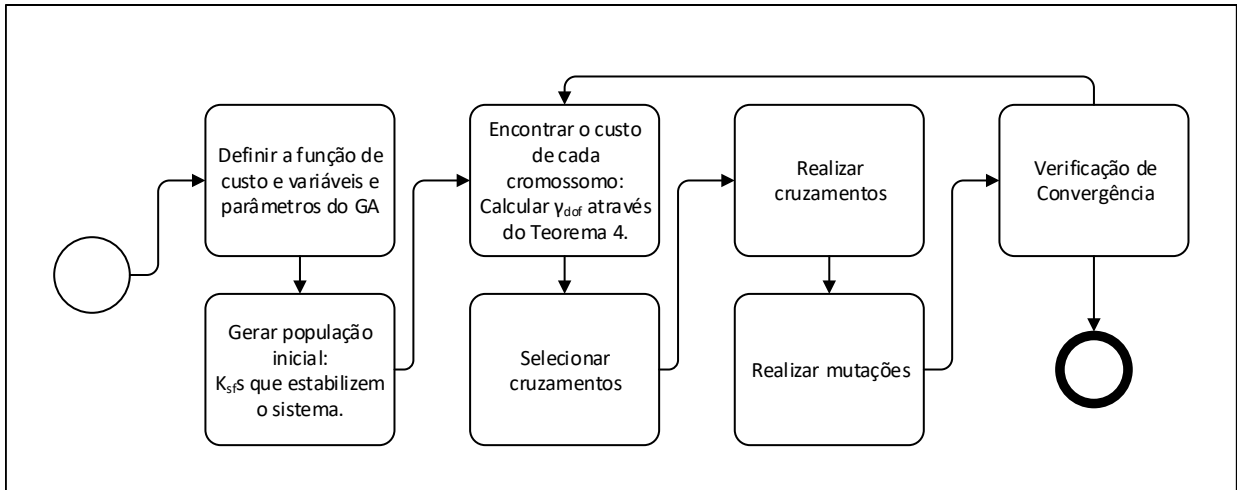


Figura 15: Fluxograma do Algoritmo Genético implementado

Fonte: Autoria própria

confiança foram $c_1 = 0,7$ e $c_{max} = 1,43$. Posteriormente foram implementadas as melhorias *Refresh Operator*, utilizando a heurística b) proposta na geração da população inicial. Neste caso, quando um partícula é redefinida, o seu custo associado é definido como ∞ . Pesos Inerciais Dinâmicos, com $\alpha = 1 \times 10^{-2}$ também foram utilizados.

Foram experimentados diversos parâmetros para N_{pop} e N_{kept} no GA, por exemplo $N_{pop} = 12$ e $N_{pop} = 36$, mas durante testes realizados em diversos cenários, as configurações que apresentaram os melhores resultados relacionados ao tempo de convergência e número de iterações foram $N_{pop} = 24$ e $N_{kept} = 50\%$, sendo estes os parâmetros utilizados para obtenção dos resultados apresentados no Capítulo 5.

O critério de parada para ambos os algoritmos foi definido como sendo a variância do melhor custo das últimas 30 iterações. Se a variância for menor que 1×10^{-9} os algoritmos finalizam, caso contrário o processo executa novamente até que o critério de parada seja satisfeito. Foi estabelecido também um limite de tempo de 3000 minutos, ou 50 horas.

Foi observado que, tanto na execução do PSO quanto do GA, muitas vezes a função objetivo era executada mais de uma vez com os mesmos parâmetros $K_{sf}(\alpha)$ utilizados previamente, ou seja, um ponto já explorado anteriormente. Com o objetivo de otimizar o tempo de execução do processo, visto que a execução da LMI proposta no segundo estágio pode ser um processo muito oneroso em termos computacionais, de acordo com a complexidade do sistema, foi implementado um *cache*, que guarda os resultados da função objetivo, indexando através de uma chave MD5 (acrônimo em inglês para *Message*

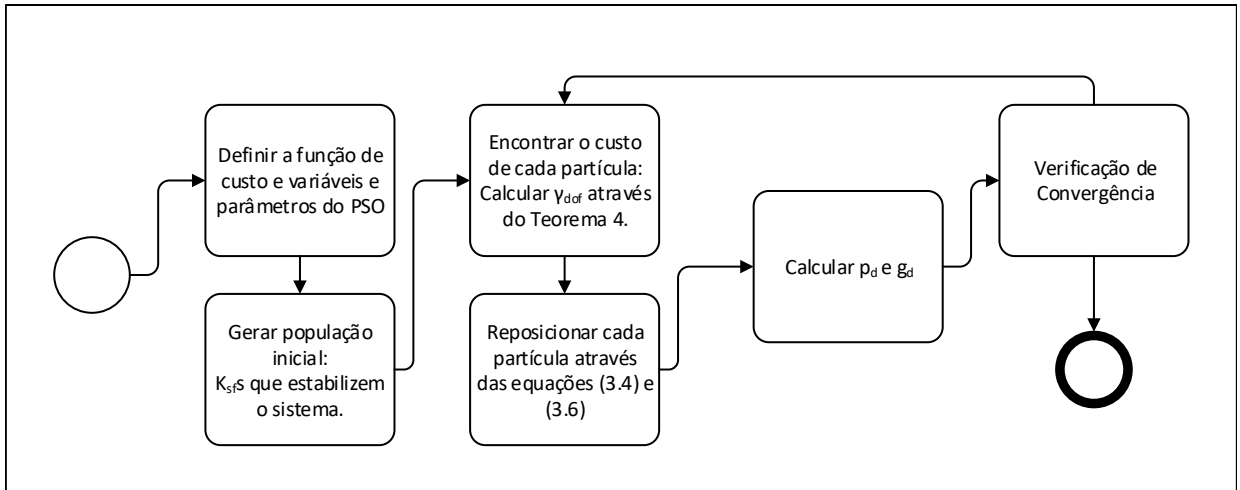


Figura 16: Fluxograma do PSO implementado

Fonte: Autoria própria

Digest 5), padrão amplamente utilizado para geração de chaves de criptografia, gerada a partir dos dados do sistema e também dos parâmetros de entrada. Dessa forma, ao invés de processar novamente pontos que já haviam sido processados previamente, o *cache* é consultado e em caso de existência é poupada a execução da condição LMI e consequente cálculo da função objetivo.

Como os algoritmos foram desenvolvidos de forma genérica para otimizar qualquer sistema estabilizável submetido a testes, algumas particularidades como uma otimização fina dos parâmetros inerentes a cada algoritmo não puderam ser implementadas, uma vez que, em testes, otimizavam os resultados para alguns sistemas, mas acabam prejudicando os resultados de outros. Uma abordagem utilizada com o objetivo de superar essas dificuldades foi a implementação de uma otimização local, conforme anteriormente proposta em (AGULHARI et al., 2011). No contexto dos algoritmos inteligentes, a otimização foi inserida após os novos elementos serem calculados ou reposicionados, através de um conjunto limitado de até 3 iterações, onde $\tilde{K}_{sf}(\alpha) = \tilde{K}_{dof}(\alpha)\tilde{C}_2(\alpha)$. Caso seja encontrado um γ menor que o gerado pelo algoritmo, $\tilde{K}_{sf}(\alpha)$ é substituído pela versão otimizada.

No capítulo seguinte serão apresentados os experimentos numéricos realizados com o objetivo de validar as técnicas desenvolvidas. Esses resultados são também comparados com resultados de ferramentas conhecidas na literatura.

5 EXPERIMENTOS NUMÉRICOS

As rotinas foram implementadas usando Matlab® 8.1.0.604 (R2013a), com os pacotes Yalmip (LÖFBERG, 2004), SeDuMi (STURM, 1999) e *Robust LMI Parser* (AGULHARI et al., 2012). O computador utilizado para os experimentos realizados foi um Intel® Core™ i7-4770 CPU (3.40GHz), 15.00GB RAM, Ubuntu 15.04 x64. Visando utilizar os recursos computacionais da melhor forma possível, a execução dos algoritmos foi realizada em paralelo através da interface de linha de comando (CLI, acrônimo em inglês para *Command Line Interface*) do Matlab®. Esta estratégia permitiu a utilização completa dos 8 núcleos lógicos do processador i7.

Com o objetivo de validar e também calibrar tanto o GA quanto o PSO, foram utilizados exemplos de sistemas disponibilizados pela biblioteca COMPlib (LEIBFRITZ; LIPINSKI, 2003), que são precisamente conhecidos. Esses sistemas foram avaliados tanto por controladores de ordem estática ($n_c = 0$), quanto por controladores de ordem completa ($n_c = n_x$). Os limitantes da norma \mathcal{H}_∞ foram comparados aos resultados obtidos usando o procedimento HIFOO (GUMUSSOY et al., 2009), também avaliados com $n_c = 0$ e $n_c = n_x$. Os resultados são apresentados na Tabela 13.

Alguns sistemas não puderam ser processados, ou não chegaram a convergir devido a características que geraram infactibilidade nas LMIs, ou seja, uma grande parte da região de soluções não foi passível de ser processada computacionalmente, utilizando o *solver* SeDuMi. É possível que utilizando outros *solvers* como o *LMI Lab*, disponível no *Robust Control Toolbox™* do Matlab®, essas regiões pudessem ser processadas, porém o tempo de execução desse *solver* é muito mais elevado que o do SeDuMi. É importante ressaltar também que, mesmo que os sistemas que convergiram possuíssem regiões de infactibilidade, elas não deviam ser predominantes e por isso não impediram a execução dos algoritmos, contudo podem ter influenciado nos resultados finais.

A partir dos dados apresentados na Tabela 13, é possível extrair as seguintes informações, desconsiderando os sistemas que não convergiram, marcados com hífen.

Tabela 13: Limitantes da norma \mathcal{H}_∞ para GA e PSO comparados aos resultados do HIFOO para os exemplos de sistema do COMPlib

Sistema	n_x	HiFOO		GA		PSO	
		$n_c = 0$	$n_c = n_x$	$n_c = 0$	$n_c = n_x$	$n_c = 0$	$n_c = n_x$
AC1	5	0,0054	$5,9319e-4$	$2,4239e-5$	0,0061	0,0046	$2,7166e-5$
AC2	5	0,1115	0,1115	0,1115	0,1115	0,1115	0,1115
AC3	5	3,6656	2,9764	3,5343	3,9352	4,0332	3,2609
AC4	4	0,9355	0,5573	0,9409	0,6251	0,9355	0,7133
AC5	4	672,3769	667,4774	667,8693	666,0209	667,5332	709,4076
AC6	7	4,1140	3,5887	4,1187	4,3522	4,1284	3,6678
AC7	9	0,0651	0,0620	-	-	-	-
AC8	9	2,0050	1,7109	-	-	-	-
AC9	10	1,0056	1,0873	-	-	-	-
AC10	55	∞	∞	-	-	-	-
AC11	5	2,8498	2,8353	2,8333	2,8827	2,8344	2,9674
AC12	4	0,3179	0,0097	$4,8367e-7$	$4,2063e-6$	0,0073	$4,8689e-4$
AC13	28	163,7919	165,1472	-	-	-	-
AC14	40	101,7309	100,0001	-	-	-	-

Fonte: Autoria própria

- 88% dos resultados foram melhores ou iguais ao HIFOO para alguma das técnicas com $n_c = 0$, sendo que:
 - o GA foi melhor que o HIFOO em 75% dos casos avaliados e
 - o PSO foi melhor que o HIFOO em 75% dos casos avaliados;
- 50% dos resultados foram melhores ou iguais ao HIFOO para alguma das técnicas com $n_c = n_x$, sendo que:
 - o GA foi melhor que o HIFOO em 38% dos casos estudados e
 - o PSO foi melhor que o HIFOO em 38% dos casos estudados;
- comparando as duas técnicas desenvolvidas:
 - o PSO foi melhor que o GA em 38% dos casos estudados e não tiveram resultados iguais, para $n_c = 0$;
 - o PSO foi melhor que o GA em 50% dos casos estudados e não tiveram resultados iguais, para $n_c = n_x$;
- Comparando a ordem dos controladores:
 - os resultados do HIFOO com $n_c = n_x$ foram melhores que $n_c = 0$ em 88% dos casos;

os resultados do GA com $n_c = n_x$ foram melhores que $n_c = 0$ em 25% dos casos e;

os resultados do PSO com $n_c = n_x$ foram melhores que $n_c = 0$ em 63% dos casos;

- Em alguns casos, como no AC12, os resultados das técnicas propostas foram extremamente pequenos e tiveram uma margem muito grande com relação ao HIFOO.

Nas figuras a seguir são feitos comparativos entre os resultados de duas técnicas, representadas pelos dois eixos das figuras. O objetivo desta análise é permitir visualizar a dispersão dos resultados e avaliar quais das técnicas apresentaram os melhores resultados. A reta em azul indica a simetria entre as técnicas, ou seja, os pontos próximos dessa reta mostram que as duas técnicas comparadas convergiram para o mesmo limitante da norma \mathcal{H}_∞ , ou que estão muito próximos. Quando os pontos identificados por x, em vermelho, estiverem abaixo da reta azul a técnica do eixo das ordenadas apresenta um resultado melhor para o caso estudado. Da mesma forma, quando os pontos identificados por x, na cor vermelha, estiverem acima da reta azul, então a técnica do eixo das abscissas apresenta resultados melhores para o sistema estudado.

Para critério de análise, nas figuras a seguir foram incluídos apenas os dados dos sistemas que apresentaram resultados para todas as técnicas abordadas, ou seja, os sistemas em que algum dos algoritmos não foram capazes de convergir não estão sendo considerados. Para as avaliações do COMPLib foram desconsiderados do diagrama os resultados do sistema AC5, que apresentou valores muito distantes dos demais.

A Figura 18 apresenta os resultados do HIFOO e do GA para os sistemas do COMPLib, utilizando controladores $n_c = 0$ e $n_c = n_x$. É possível observar que as duas técnicas apresentaram resultados muito próximos para $n_c = 0$, já para $n_c = n_x$ o HIFOO apresentou resultados melhores para dois dos sistemas. O mesmo comportamento pode ser observado na Figura 19, que compara os mesmos casos porém agora com o HIFOO e PSO. Para $n_c = 0$, o HIFOO teve uma pequena vantagem nos resultados, mas os demais foram muito próximos. Já analisando a Figura 20 que compara o GA e PSO, fica evidente a semelhança nos resultados das metaheurísticas com vantagem em um dos casos para o GA com $n_c = 0$. Para $n_c = n_x$ é possível observar que o PSO consegue se destacar, apresentando resultados melhores. Analisando a Figura 21, que compara os resultados para as duas ordens de controladores para cada uma das técnicas estudadas, fica mais evidente que as metaheurísticas não tiveram um resultado tão bom quanto o HIFOO ao aumentar a ordem do controlador. Embora fosse esperado que os resultados para

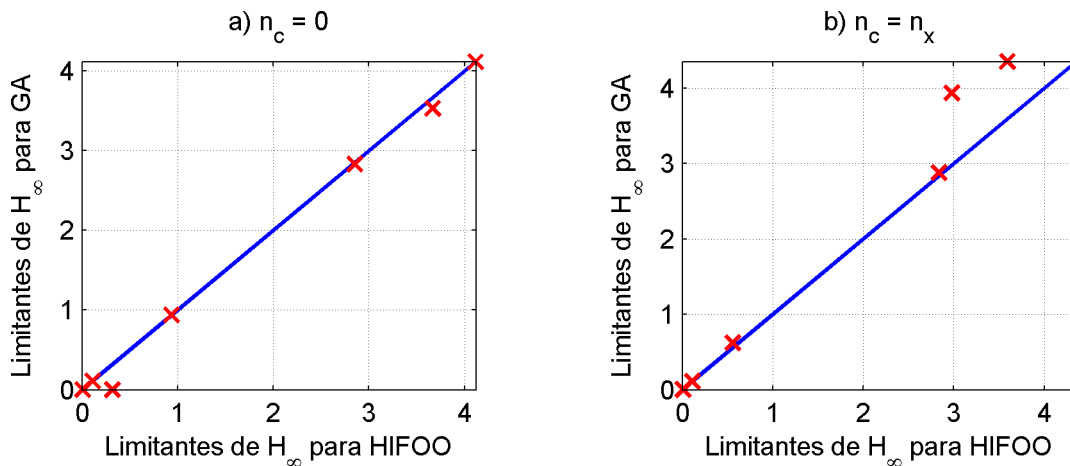


Figura 18: Comparação dos resultados dos sistemas do COMPLib para GA e HIFOO com $n_c = 0$ e $n_c = n_x$

Fonte: Autoria própria

$n_c = n_x$ fossem sempre melhores que para $n_c = 0$, esse comportamento ao aumentar a ordem dos controladores pode ser justificado pela complexidade numérica envolvida. Cada nova variável aumenta consideravelmente as opções de busca das metaheurísticas e estas apresentam uma queda de performance ao comparar com os resultados do HIFOO.

As técnicas desenvolvidas também foram submetidas a testes utilizando sistemas estabilizáveis precisamente conhecidos gerados aleatoriamente, com ordem do sistema (n_x) no intervalo $[2,5]$. Os resultados são dispostos na Tabela 14. Os sistemas podem ser acessados através do repositório online (PASCOAL, 2016), combinando os arquivos *systems_random_1.mat* e *selected_precise_known.mat*.

Para estes, as seguintes informações foram extraídas:

- 36% dos resultados foram melhores ou iguais ao HIFOO para alguma das técnicas com $n_c = 0$, sendo que:
 - o GA foi melhor que o HIFOO em 12% dos casos avaliados e
 - o PSO foi melhor que o HIFOO em 32% dos casos avaliados;
- 18% dos resultados foram melhores ou iguais ao HIFOO para alguma das técnicas com $n_c = n_x$, sendo que:
 - o GA foi melhor que o HIFOO em 10% dos casos estudados e
 - o PSO foi melhor que o HIFOO em 10% dos casos estudados;
- comparando as duas técnicas desenvolvidas:

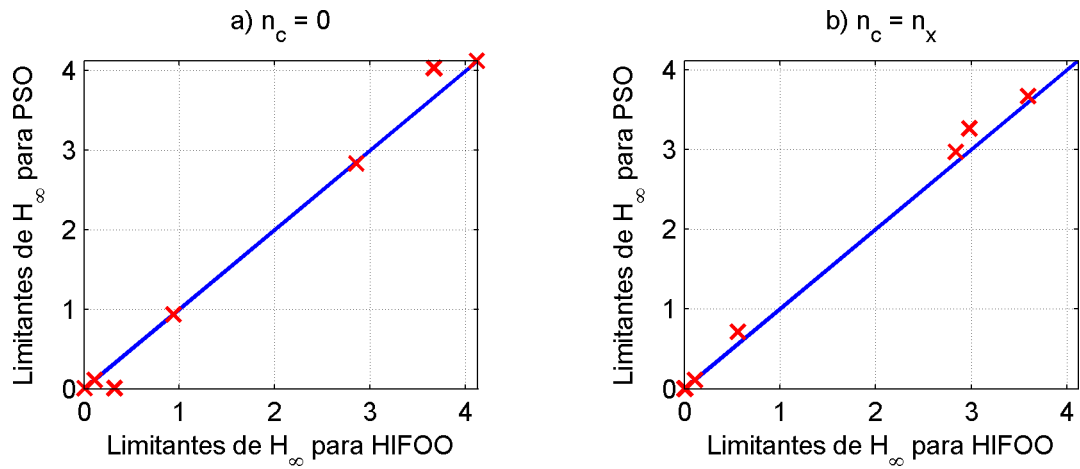


Figura 19: Comparação dos resultados dos sistemas do COMPLib para PSO e HIFOO com $n_c = 0$ e $n_c = n_x$

Fonte: Autoria própria

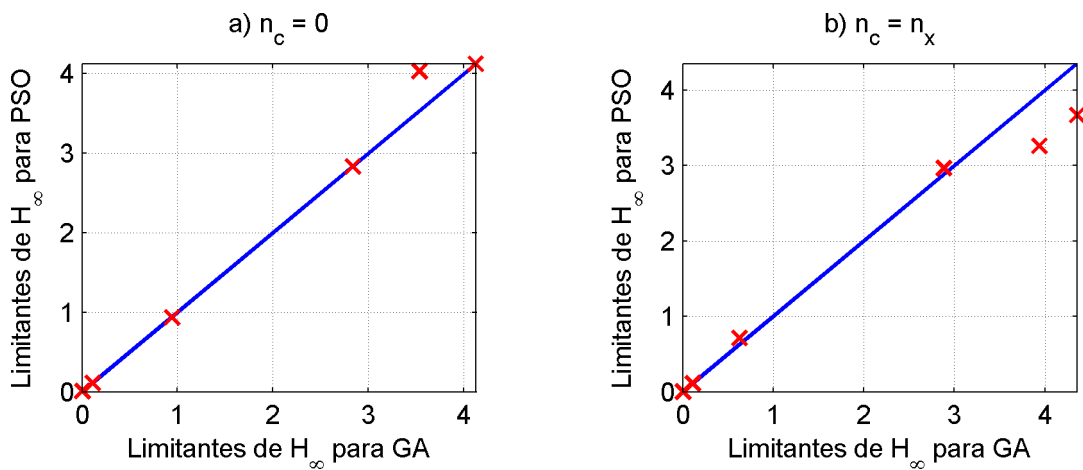


Figura 20: Comparação dos resultados dos sistemas do COMPLib para PSO e GA com $n_c = 0$ e $n_c = n_x$

Fonte: Autoria própria

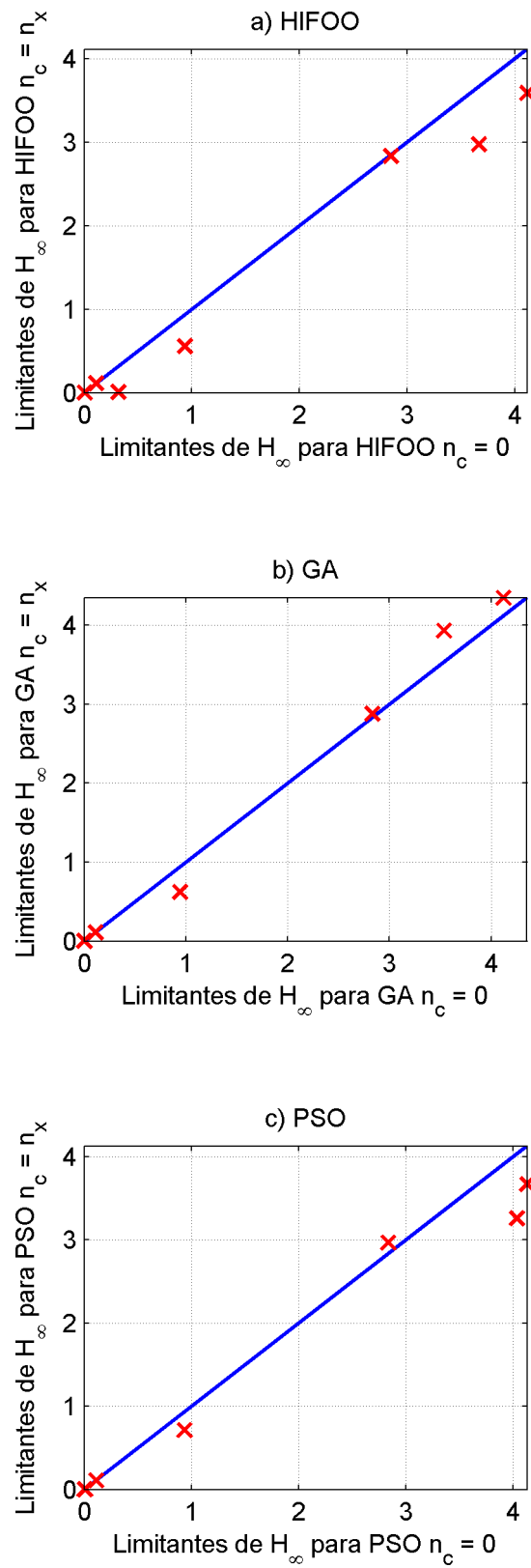


Figura 21: Comparação dos resultados dos sistemas do COMPLib com controladores $n_c = 0$ e $n_c = n_x$

Fonte: Autoria própria

Tabela 14: Limitantes da norma \mathcal{H}_∞ para GA e PSO comparados aos resultados do HIFOO para sistemas estabilizáveis precisamente conhecidos gerados aleatoriamente

Sistema	n_x	HiFOO		GA		PSO	
		$n_c = 0$	$n_c = n_x$	$n_c = 0$	$n_c = n_x$	$n_c = 0$	$n_c = n_x$
1	2	2,5823	1,4713	2,5826	1,9274	2,5825	1,5096
2	4	1,0282	0,6892	1,8474	1,8474	1,8474	1,8474
3	2	10,7478	10,7123	10,7478	10,7127	10,7478	10,7125
4	4	5,7456	5,4503	5,7795	5,4129	5,7456	5,3336
5	3	2,6065	0,9838	2,6257	2,0807	2,6066	1,6676
6	5	13,7483	12,9630	13,7792	13,2814	13,7483	13,1905
7	3	1,9514	0,7831	2,9469	1,8199	1,9557	1,7585
8	3	3,0921	2,6817	3,0921	2,7000	3,0921	2,6841
10	4	0,9103	0,8659	1,9082	1,9096	1,9082	1,9082
11	4	6,3643	6,0059	6,5067	6,6376	6,3643	5,9564
13	5	2,2024	2,1271	2,7688	2,7688	2,7688	2,7688
14	5	0,7906	0,7587	1,6576	1,6572	1,6572	1,6572
15	4	2,3933	2,3890	3,0248	3,0249	3,0249	3,0248
16	2	1,1658	1,1626	1,3654	1,3654	1,3654	1,3654
17	5	37,9884	37,9834	38,1052	337,6159	38,0655	-
18	5	8,3459	8,0891	8,3459	8,1617	8,3459	8,0904
19	4	5,3155	4,8447	5,3155	4,9152	5,3155	4,8586
20	5	7,4606	7,2623	7,4605	7,2658	7,4605	7,2640
21	3	2,3915	2,2941	2,3922	2,3000	2,3923	2,3035
22	5	608,5737	260,1206	-	214,5285	-	-
24	5	1,6829	0,9162	1,9862	1,6850	1,9362	1,6387
25	3	21,3658	20,9060	21,3659	20,9366	21,3658	20,9146
26	4	4,7976	1,1323	4,8642	2,0523	4,7975	1,8393
27	4	22,3124	21,8579	22,3124	21,7382	22,3124	21,7372
28	5	5,4807	2,4460	5,4807	2,7943	5,4807	2,6085
29	4	10,5780	2,4915	10,5781	3,8859	10,5783	2,6305
30	4	2,7825	2,3813	2,7827	2,3877	2,7824	2,3802
31	3	4,4487	4,4370	4,4685	4,4662	4,4487	4,4841
32	3	0,4598	0,4532	1,3479	1,3479	1,3479	1,3479
33	3	0,5101	0,4400	1,6060	1,6060	1,6060	1,6060
34	3	1,8335	1,7881	1,8639	1,8639	1,8639	1,8639
35	5	14,2344	12,7383	15,3439	14,1324	14,2344	12,7892
36	2	3,6333	1,4127	3,6947	1,6040	3,6333	1,5046
37	2	2,1088	2,0692	2,1088	1,7078	2,1088	2,0735
39	4	2,0942	1,0337	2,1291	2,1929	2,0942	2,0508
40	4	11,5692	7,3557	11,6254	7,6171	11,5693	7,6411
41	4	1,2988	1,0721	1,9629	1,9651	1,9629	1,9629
43	3	0,9829	0,8693	1,2787	1,2790	1,2787	1,2787
44	4	1,0312	0,9118	1,2621	1,2779	1,2710	1,2621
45	3	2,2321	1,3209	2,2656	1,4551	2,2322	1,4254
46	3	0,4106	0,3164	1,9731	1,9731	1,9731	1,9731
47	3	2,4355	1,6393	2,4377	1,7464	2,4388	1,6445
48	4	12,6319	10,8031	12,6319	10,9175	12,6319	10,8652
49	3	9,4693	9,4588	9,4917	9,3419	9,4693	9,3726
50	2	103,4826	93,6851	-	93,9957	-	-

Fonte: Autoria própria

o PSO foi melhor que o GA em 44% dos casos estudados e tiveram resultados iguais em 30%, para $n_c = 0$;

o PSO foi melhor que o GA em 68% dos casos estudados e tiveram resultados iguais em 6%, para $n_c = n_x$

- Comparando a ordem dos controladores:

os resultados do HIFOO com $n_c = n_x$ foram melhores que $n_c = 0$ em 88% dos casos;

os resultados do GA com $n_c = n_x$ foram melhores que $n_c = 0$ em 62% dos casos;

os resultados do PSO com $n_c = n_x$ foram melhores que $n_c = 0$ em 76% dos casos.

A Figura 22 apresenta os resultados do HIFOO e do GA para os sistemas estáveis precisamente conhecidos gerados aleatoriamente, utilizando controladores estáticos e de ordem completa. É possível observar que em muitos dos sistemas estudados, os resultados do GA foram semelhantes aos do HIFOO, contudo em uma boa parcela destes o HIFOO apresentou resultados melhores, porém não tão dispersos. Nota-se também que quando os limitantes da norma \mathcal{H}_∞ aproximam-se de zero, o HIFOO leva vantagem, já quando os limitantes da norma \mathcal{H}_∞ tem um valor mais elevado, as duas técnicas convergem para resultados próximos. O mesmo comportamento pode ser observado pela análise da Figura 23, que mostra os resultados do HIFOO comparados aos do PSO. Entretanto, o PSO aproxima-se mais dos resultados do HIFOO para os controladores de ordem estática. Ao comparar as duas metaheurísticas (Figura 24), evidencia-se que ambas produziram resultados muito próximos, mas o PSO apresentou vantagem tanto para $n_c = 0$, quanto para $n_c = n_x$. Já comparando individualmente as técnicas para os dois controladores (Figura 25), é fácil observar que para todas as técnicas, os controladores de ordem completa foram mais eficientes que os de ordem estática. Entretanto, assim como na avaliação dos resultados para os sistemas do COMPlib, aumentar a ordem dos controladores não apresentou melhoras significativas nos resultados ao utilizar as metaheurísticas propostas, já o HIFOO apresentou resultados melhores em muitos dos casos estudados.

Os resultados obtidos para os sistemas precisamente conhecidos, tanto os do COMPlib, quanto os gerados aleatoriamente, são considerados satisfatórios e atingiram os objetivos propostos: aproximar os resultados produzidos pelas metaheurísticas de forma a calibrar com uma técnica de síntese considerada como estado da arte, o HIFOO. Também é possível concluir que os resultados das técnicas desenvolvidas podem ser considerados equivalentes aos do HIFOO, utilizando controladores estáticos. Já para controladores de

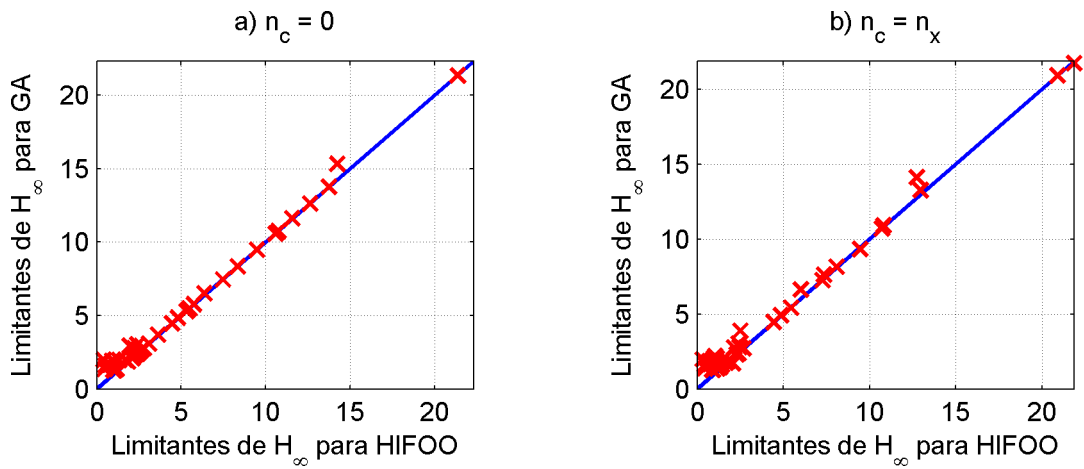


Figura 22: Comparação dos resultados dos sistemas estabilizáveis precisamente conhecidos gerados aleatoriamente para GA e HIFOO com $n_c = 0$ e $n_c = n_x$

Fonte: Autoria própria

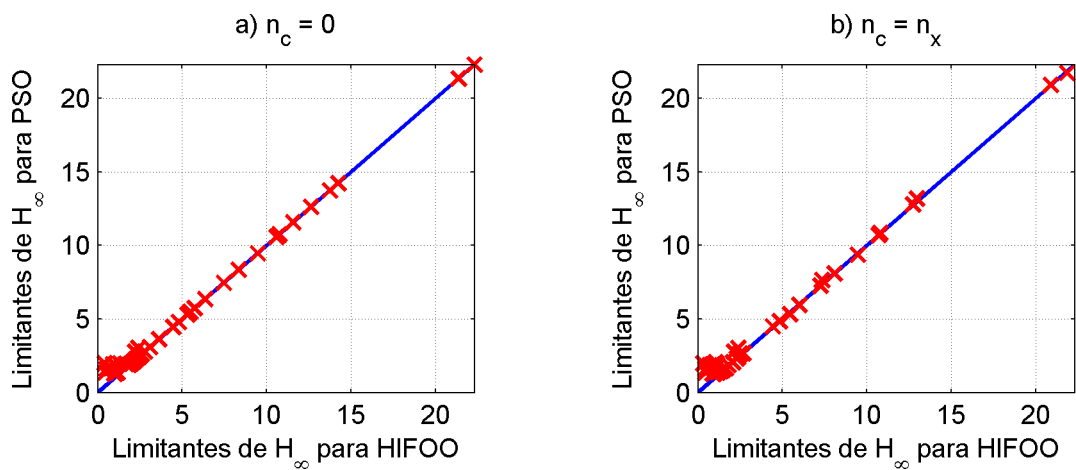


Figura 23: Comparação dos resultados dos sistemas estabilizáveis precisamente conhecidos gerados aleatoriamente para PSO e HIFOO com $n_c = 0$ e $n_c = n_x$

Fonte: Autoria própria

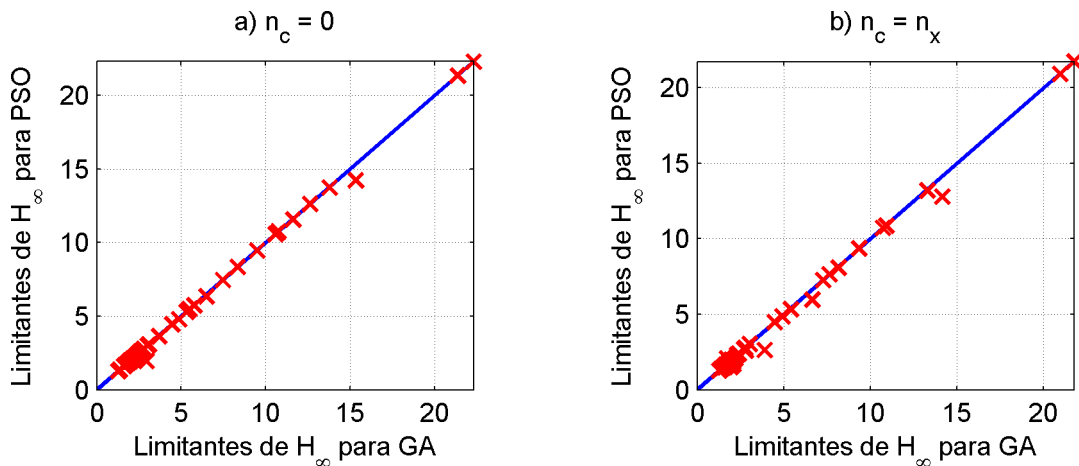


Figura 24: Comparação dos resultados dos sistemas estabilizáveis precisamente conhecidos gerados aleatoriamente para PSO e GA com $n_c = 0$ e $n_c = n_x$

Fonte: Autoria própria

ordem completa, a metaheurística que melhor se aplica é o PSO. Contudo, as duas técnicas estudadas apresentam desvantagens quanto ao HIFOO.

As técnicas propostas, por outro lado, estão preparadas para lidar com sistemas incertos, mesmo se todas as matrizes do sistema forem afetadas por incertezas. Este cenário não é abordado pela maioria das técnicas na literatura, por exemplo o pacote HIFOO.

Para esse cenário, são propostos sistemas estabilizáveis incertos gerados aleatoriamente, com ordem do sistema (n_x) no intervalo $[2,5]$. Os resultados da execução dos algoritmos propostos são apresentados na Tabela 15. Os sistemas podem ser acessados através do repositório online (PASCOAL, 2016), combinando acessando os arquivos *systems_random_2.mat* e *selected_uncertain.mat*.

Para estes casos estudados, foram identificados os seguintes comportamentos:

- comparando as duas técnicas desenvolvidas:
 - o PSO foi melhor que o GA em 47% dos casos estudados e tiveram resultados iguais em 24% dos casos estudados, para $n_c = 0$;
 - o PSO foi melhor que o GA em 53% dos casos estudados e tiveram resultados iguais em 24% dos casos estudados, para $n_c = n_x$;
- Comparando a ordem dos controladores:
 - os resultados do GA com $n_c = n_x$ foram melhores que $n_c = 0$ em 53% dos casos

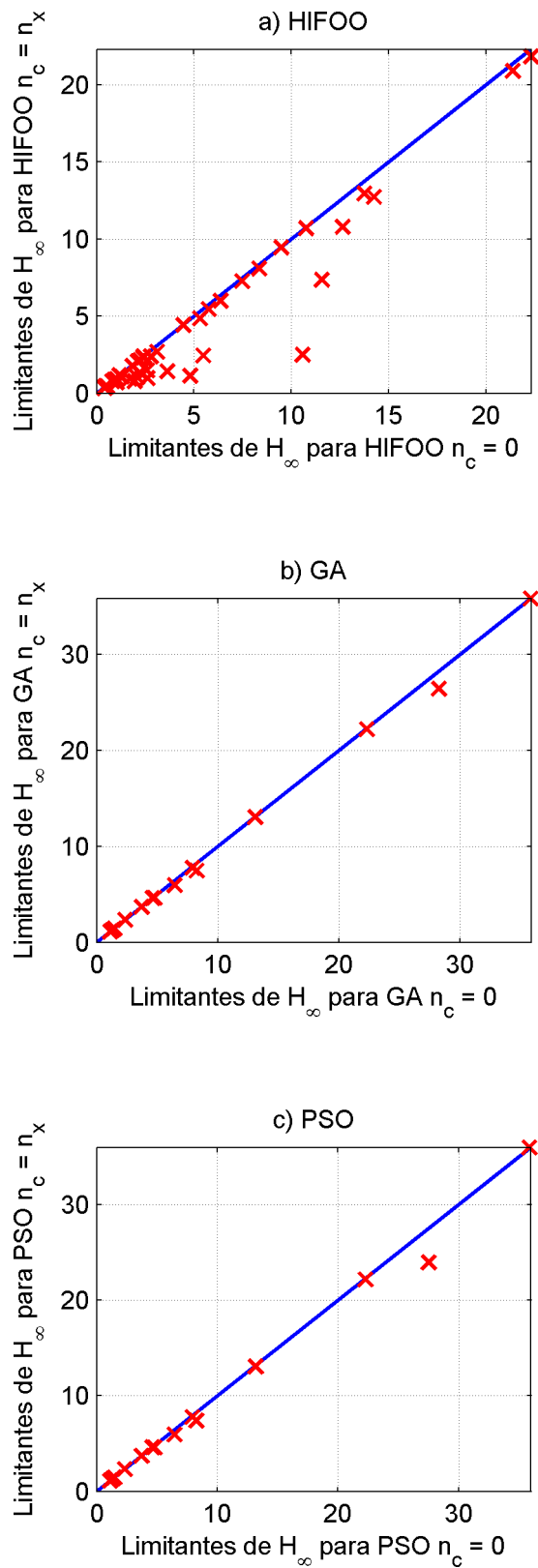


Figura 25: Comparação dos resultados dos sistemas estabilizáveis precisamente conhecidos gerados aleatoriamente para com controladores $n_c = 0$ e $n_c = n_x$

Fonte: Autoria própria

Tabela 15: Limitantes da norma \mathcal{H}_∞ para GA e PSO para sistemas incertos estabilizáveis gerados aleatoriamente

Sistema	n_x	GA		PSO	
		$n_c = 0$	$n_c = n_x$	$n_c = 0$	$n_c = n_x$
1	2	-	-	-	-
2	3	92,3809	-	92,3809	-
3	5	4,6125	4,6125	4,6125	4,6125
4	3	22,2940	22,2322	22,2673	22,1604
5	5	9,9261	-	8,1819	-
6	3	35,8313	35,8834	35,8511	35,9996
7	2	3,7104	3,7104	3,7104	3,7103
8	3	13,0832	13,0499	13,1736	13,0506
9	5	-	-	-	-
10	3	1,4432	1,4426	1,4418	1,4427
11	3	1,2796	1,2797	1,2797	1,2797
12	5	28,2579	26,4681	27,5136	23,9652
13	2	1,1102	1,1102	1,1102	1,1102
14	4	6,4446	5,9541	6,4444	5,9540
15	4	-	-	-	-
16	2	4,7887	4,6484	4,7868	4,6018
17	5	2,3467	2,3576	2,3146	2,3231
18	3	13,0939	13,0659	13,157	13,0498
19	3	7,9193	7,7600	7,9295	7,7670
20	3	8,2640	7,4966	8,2573	7,4224

Fonte: Autoria própria

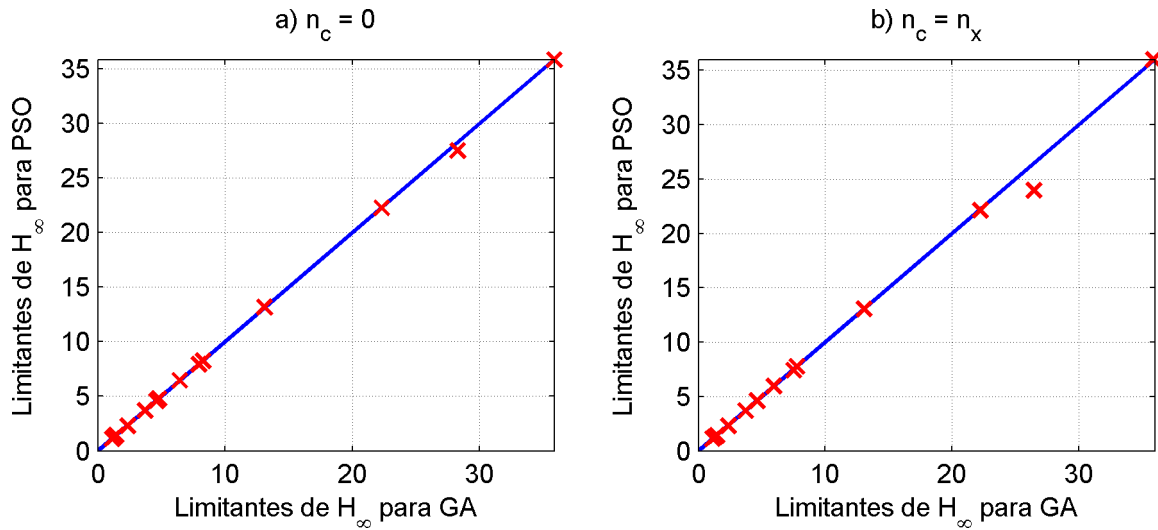


Figura 26: Comparação dos resultados dos sistemas estabilizáveis incertos gerados aleatoriamente para PSO e GA com $n_c = 0$ e $n_c = n_x$

Fonte: Autoria própria

e;

os resultados do PSO com $n_c = n_x$ foram melhores que $n_c = 0$ em 53% dos casos.

A Figura 26 apresenta os resultados do GA e do PSO para os sistemas estabilizáveis incertos gerados aleatoriamente, utilizando controladores de ordem $n_c = 0$ e $n_c = n_x$. É possível observar que de uma forma geral os resultados foram muito semelhantes para os dois controladores, mas com uma pequena vantagem para o PSO. Este comportamento também é notado avaliando individualmente as técnicas (Figura 27).

Como o HIFOO não é capaz de lidar com sistemas incertos, com o objetivo de validar as técnicas propostas para sistemas incertos utilizando *benchmark* como comparativo, foi utilizado o sistema massa-mola sugerido em (AGULHARI et al., 2011) como exemplo, considerando:

$$\left[\begin{array}{c|c|c} A(\alpha) & B_1 & B_2 \\ \hline C_1 & D_1 & D_2 \\ \hline C_2 & D_y & 0 \end{array} \right] = \left[\begin{array}{cccc|cc} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ -\frac{k_1(\alpha)+k_2}{m_1} & \frac{k_2}{m_1} & -\frac{c_0(\alpha)}{m_1} & 0 & 1 & m_1 \\ \hline \frac{k_2}{m_2} & -\frac{k_2}{m_2} & 0 & -\frac{c_0(\alpha)}{m_2} & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{array} \right] \quad (66)$$

sendo $m_1 = 2$, $m_2 = 1$, $k_2 = 0,5$ e $1 \leq k_1 \leq 4$, $1 \leq c_0 \leq 4$, produzindo um politopo de

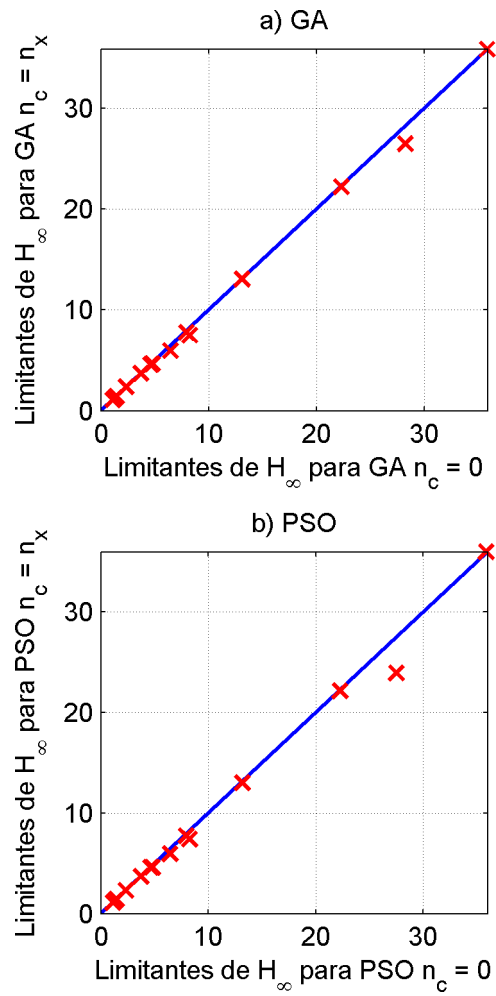


Figura 27: Comparação dos resultados dos sistemas estabilizáveis incertos gerados aleatoriamente para com controladores $n_c = 0$ e $n_c = n_x$

Fonte: Autoria própria

Tabela 16: Limitantes da norma \mathcal{H}_∞ produzidos pelo GA e PSO para o sistema incerto (66)

Técnica	$n_c = 0$			$n_c = n_x$		
	min	median	max	min	median	max
método iterativo		3,09			5,71	
GA	0,000005	0,000013	0,000030	0,000006	0,000018	0,000080
PSO	0,000112	0,011600	0,019600	0,000103	0,007601	0,021059

Fonte: Autoria própria

$N = 4$ vértices. Os resultados obtidos aplicando as técnicas propostas, assim como o método iterativo apresentado em (AGULHARI et al., 2011), são apresentados na Tabela 16.

Esses resultados foram coletados a partir de 10 iterações para ambos o GA e o PSO, com ordem dos controladores reduzida e completa, já que a execução dos algoritmos pode convergir em resultados diferentes. Na Tabela 16 são apresentados para cada cenário informações estatísticas do mínimo, mediana e o máximo da norma \mathcal{H}_∞ . O objetivo de se apresentar resultados estatísticos é de poder afirma, mesmo sabendo que as metaheurísticas nem sempre irão convergir para os mesmos resultados, de uma forma geral eles sempre estão próximos. É possível observar que os resultados das técnicas desenvolvidas apresentaram melhorias significativas, sendo que a maioria deles estão muito próximos de zero. De uma forma geral, o GA apresentou resultados melhores que o PSO em todos os casos e, diferentemente do método iterativo, os controladores de ordem completa também apresentaram resultados muito semelhantes aos dos controladores de ordem reduzida.

Ao comparar os resultados das duas estratégias, é possível observar que elas nem sempre convergem para os mesmos resultados. Isso pode ser justificado pelo comportamento dos algoritmos quanto ao critério em que eles navegam pelas soluções, além das regiões de infactibilidade. Também é possível observar, de uma forma geral, que, comparando a ordem dos controladores ($n_c = 0$ e $n_c = n_x$), os controladores de ordem completa foram melhores que os estáticos, conforme esperado, de acordo com a literatura.

De uma forma geral, é possível observar que as técnicas propostas são capazes de minimizar os limitantes da norma \mathcal{H}_∞ em quase todos os casos analisados. Os resultados, contudo, variam de acordo com o experimento. Estatisticamente, esses mostraram-se muito próximos, ou melhores em alguns casos, da ferramenta de *benchmark* HIFOO, para os sistemas precisamente conhecidos. Além disso, as figuras apresentadas permitem uma

análise gráfica, onde é possível observar que as metaheurísticas estudadas estão realmente muito alinhadas. Para os sistemas incertos, os resultados foram significativamente melhores que o método iterativo proposto em (AGULHARI et al., 2011), que incentivaram o início da pesquisa.

O objetivo ao comparar os resultados das metaheurísticas com os do HIFOO para sistemas precisamente conhecidos é validar as propostas com resultados de ferramentas conhecidas também como calibragem dos algoritmos. Uma vez que estes estivessem alinhados, os resultados para os sistemas incertos, contribuição principal do trabalho, estariam validados indiretamente com o HIFOO, já que este não pode ser utilizado para sistemas incertos. Tendo em vista esta meta estabelecida, é possível concluir que trabalho desenvolvido atingiu os resultados esperados e que as implementações das metaheurísticas são capazes de minimizar os limitantes da norma \mathcal{H}_∞ para o controle de sistemas em que as incertezas possam estar presentes em todas as matrizes do sistema.

6 CONCLUSÕES

Duas técnicas para a síntese de controladores robustos que minimizam os limitantes da norma \mathcal{H}_∞ por realimentação de saídas com ordem reduzida para sistemas incertos de tempo contínuo foram apresentadas neste trabalho. As técnicas propostas são adaptações da técnica de dois estágios apresentadas em (PEAUCELLE; ARZELIER, 2001; ARZELIER et al., 2003) e consistem na utilização de um GA ou um PSO para melhorar os resultados. As técnicas propostas foram implementadas, calibradas e testadas com sistemas precisamente conhecidos disponíveis na biblioteca COMPlib e sistemas estabilizáveis precisamente conhecidos gerados aleatoriamente, e os resultados foram comparados aos produzidos pelo HIFOO (GUMUSSOY et al., 2009). Essas técnicas foram também testadas com um sistema incerto já proposto na literatura e sistemas incertos estabilizáveis gerados aleatoriamente, e os resultados obtidos foram comparados com (AGULHARI et al., 2011).

Os resultados coletados a partir dos experimentos numéricos mostraram que as metaheurísticas implementadas foram capazes de manter-se próximas da ferramenta de *benchmark* HIFOO na maioria dos casos estudados e em alguns apresentaram resultados significativamente melhores. Comparando a ordem dos controladores, nos controladores de ordem completa este índice foi reduzido, mas ainda assim as metaheurísticas desenvolvidas apresentaram resultados similares aos do HIFOO. De uma forma geral, o PSO apresentou resultados estatisticamente melhores que o GA, mas estes se mantiveram também muito próximos na maioria dos casos estudados. Esses resultados surpreenderam, uma vez que o HIFOO é um pacote consolidado e reconhecido pela literatura como uma ferramenta de *benchmark*, devido à sua vasta utilização e melhorias ao longo do tempo.

Para os sistemas incertos avaliados, o PSO teve um ligeira vantagem com relação ao GA para os sistemas gerados aleatoriamente, sendo que eles convergiram para os mesmos resultados em uma boa parcela dos sistemas avaliados. Já para o sistema massa-mola, tanto o PSO quanto o GA produziram resultados melhores do que os encontrados na literatura, como esperado. O GA, neste caso apresentou resultados melhores que o PSO em todos os casos. Isso prova a efetividade dessas técnicas, especialmente porque os métodos

encontrados na literatura não são capazes, na maioria, de lidar com incertezas, o que pode ser considerado como principal vantagem das técnicas propostas.

REFERÊNCIAS

- AGULHARI, C. M. **Estabilidade e controle de sistemas lineares e variantes no tempo com parâmetros incertos**. Tese (Doutorado) — Universidade Estadual de Campinas - Faculdade de Engenharia Elétrica e de Computação, 2013.
- AGULHARI, C. M.; OLIVEIRA, R. C. L. F.; PERES, P. L. D. Robust \mathcal{H}_∞ static output-feedback design for time-invariant discrete-time polytopic systems from parameter-dependent state-feedback gains. In: **Proc. 2010 Amer. Control Conf.** Baltimore, MD, USA: [s.n.], 2010. p. 4677–4682.
- AGULHARI, C. M.; OLIVEIRA, R. C. L. F.; PERES, P. L. D. LMI relaxations for reduced-order robust \mathcal{H}_∞ control of continuous-time uncertain linear systems. **IEEE Trans. Autom. Control**, v. 57, n. 6, p. 1532–1537, December 2011.
- AGULHARI, C. M.; OLIVEIRA, R. C. L. F. de; PERES, P. L. D. Robust LMI Parser: a computational package to construct LMI conditions for uncertain systems. In: **XIX Brazilian Conference on Automation (CBA 2012)**. Campina Grande, PB, Brazil: [s.n.], 2012. p. 2298–2305.
- ALFI, A. Particle swarm optimization algorithm with dynamic inertia weight for online parameter identification applied to lorenz chaotic system. **ICIC International**, v. 8, n. 2, p. 1191–1203, February 2012.
- ARZELIER, D. et al. **Mixed LMI/Randomized methods for static output feedback control design: Stability and performance**. [S.l.], September 2009.
- ARZELIER, D. et al. Mixed LMI/randomized methods for static output feedback control design. In: **Proc. 2010 Amer. Control Conf.** Baltimore, MD, USA: [s.n.], 2010. p. 4683–4688.
- ARZELIER, D.; PEAUCELLE, D.; SALHI, S. Robust static output feedback stabilization for polytopic uncertain systems: Improving the guaranteed performance bound. In: **Proc. 4th IFAC Symp. on Robust Contr. Design (ROCOND 2003)**. Milan, Italy: [s.n.], 2003. p. 425–430.
- BARA, G. I.; BOUTAYEB, M. Static output feedback stabilization with \mathcal{H}_∞ ; performance for linear discrete-time systems. **IEEE Transactions on Automatic Control**, v. 50, n. 2, p. 250–254, Feb 2005. ISSN 0018-9286.
- BENTON, R. E.; SMITH, D. A static-output-feedback design procedure for robust emergency lateral control of a highway vehicle. **IEEE Transactions on Control Systems Technology**, v. 13, n. 4, p. 618–623, July 2005. ISSN 1063-6536.
- BOYD, S. et al. **Linear matrix inequalities in system and control theory**. 2nd. ed. [S.l.]: SIAM Studies in Applied Mathematics, 1994. 193 p. ISBN 0-89871-334-X.

- CAO, Y.-Y.; LAM, J.; SUN, Y. X. Static output feedback stabilization: An ILMI approach. **Automatica**, v. 34, p. 1641–1645, 1998.
- CLERC, M. **Particle Swarm Optimization**. [S.l.]: ISTE Publishing, 2006. ISBN ISBN-13: 978-1-905209-04-0 and ISBN-10: 1-905209-04-5.
- CRUSIUS, C. A. R.; TROFINO, A. Sufficient LMI conditions for output feedback control problems. **IEEE Trans. Autom. Control**, v. 44, p. 1053–1057, 1999.
- DEIF, S.; KAMAL, H. A.; TAWFIK, M. Enhancing genetic algorithms using a dynamic mutation value approach: An application to the control of flexible robot systems. **CiiT Int. J. of Artificial Intelligent Syst. and Machine Learning**, v. 4, n. 1, December 2012.
- DOYLE, J. C. et al. State-space solutions to standard \mathcal{H}_2 and \mathcal{H}_∞ control problems. **IEEE Trans. Autom. Control**, v. 34, n. 8, p. 831–847, August 1989.
- FU, M.; LUO, Z. Computational complexity of a problem arising in fixed order output feedback design. v. 30, n. 5, p. 209–215, June 1997.
- GAHINET, P.; APKARIAN, P. A linear matrix inequality approach to \mathcal{H}_∞ control. v. 4, n. 4, p. 412–448, July-August 1994.
- GARCIA, G.; PRADIN, B.; ZENG, F. Stabilization of discrete time linear systems by static output feedback. **IEEE Trans. Autom. Control**, v. 46, p. 1954–1958, 2001.
- GEROMEL, J. C.; PERES, P. L. D.; SOUZA, S. R. Convex analysis of output feedback control problems: Robust stability and performance. **IEEE Trans. Autom. Control**, v. 41, n. 7, p. 997–1003, 1996.
- GEROMEL, J. C.; SOUZA, C. C. de; SKELTON, R. E. Static output feedback controllers: Stability and convexity. **IEEE Trans. Autom. Control**, v. 43, p. 120–125, 1998.
- GHAOUI, L. E.; OUSTRY, F.; AITRAMI, M. A cone complementarity linearization algorithm for static output-feedback and related problems. **IEEE Trans. Autom. Control**, p. 1171–1175, 1997.
- GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. [S.l.]: Reading, MA: Addison-Wesley, 1989. ISBN 0-201-15767-5.
- GUMUSSOY, S. et al. Multiobjective robust control with HIFOO 2.0. In: **Proc. 6th IFAC Symp. on Robust Contr. Design (ROCOND 2009)**. Haifa, Israel: [s.n.], 2009. p. 144–149. Disponível em: <www.cs.nyu.edu/overton/software/hifoo>.
- HAUPT, R. L.; HAUPT, S. E. **Practical genetic algorithms**. 2nd. ed. [S.l.]: Wiley-Interscience, 2004. 535 p. ISBN 0-471-45565-2.
- HENRION, D.; LASSERRE, J. B. Convergent relaxations of polynomial matrix inequalities and static output feedback. **IEEE Trans. Autom. Control**, v. 51, p. 192–202, 2006.
- HO, D. W. C.; LU, G. Robust stabilization for a class of discrete-time nonlinear systems via output feedback: The unified lmi approach. **International Journal of Control**, v. 76, n. 2, p. 105–115, 2003. Disponível em: <<http://dx.doi.org/10.1080/0020717031000067367>>.

- HOLLAND, J. H. **Adaptation in Natural and Artificial Systems**. [S.l.]: Ann Arbor: University of Michigan Pres, 1975.
- HUANG, D.; NGUANG, S. K. Robust \mathcal{H}_∞ static output feedback control of fuzzy systems: An ILMI approach. **IEEE Trans. Syst., Man, Cybern. B, Cybern.**, v. 36, p. 216–222, 2006.
- KALMAN, R. E. et al. Contributions to the theory of optimal control. **Boletín de la Sociedad Matemática Mexicana**, Citeseer, v. 5, n. 2, p. 102–119, 1960.
- KAWAKAMI, K.; MENG, Z. Improvement of particle swarm optimization. **PIERS ONLINE**, v. 5, n. 3, p. 261–264, 2009.
- KHALIL, H. **Nonlinear Systems**. Prentice Hall, 2002. (Pearson Education). ISBN 9780130673893. Disponível em: <https://books.google.com.br/books?id=t_d1QgAACAAJ>.
- KUCERA, V.; SOUZA, C. de. A necessary and sufficient condition for output feedback stabilizability. **Automatica**, v. 10, p. 1357–1359, 1995.
- LEIBFRTZ, F. An LMI-based algorithm for designing suboptimal static $\mathcal{H}_2 / \mathcal{H}_\infty$ output feedback controllers. **SIAM. J. Control Optim.**, v. 39, p. 1711–1735, 2001.
- LEIBFRTZ, F.; LIPINSKI, W. **Description of the benchmark examples in Ccompleib 1.0**. [S.l.], 2003. Disponível em: <<http://www.complib.de/>>.
- LO, J.-C.; LIN, M.-L. Robust \mathcal{H}_∞ ; nonlinear control via fuzzy static output feedback. **IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications**, v. 50, n. 11, p. 1494–1502, Nov 2003. ISSN 1057-7122.
- LÖFBERG, J. YALMIP: A toolbox for modeling and optimization in MATLAB. In: **Proc. 2004 IEEE Int. Symp. on Comput. Aided Control Syst. Des.** Taipei, Taiwan: [s.n.], 2004. p. 284–289. Disponível em: <<http://control.ee.ethz.ch/~joloef/yalmip.php>>.
- MEHDI, D.; BOUKAS, E. K.; BACHELIER, O. Static output feedback design for uncertain linear discrete time systems. v. 21, n. 1, p. 1–13, March 2004.
- OGATA, K. **Modern control engineering**. [S.l.]: Pearson, 2010.
- OSTERTAG, E. Less conservative $\mathcal{H}_2/\mathcal{H}_\infty$; design of dynamic output-feedback controllers via lmi's. In: **Control Conference (ECC), 2009 European**. [S.l.: s.n.], 2009. p. 2875–2880.
- PASCOAL, M. de C. **Dissertação**. GitHub, 2016. Disponível em: <<https://github.com/marlonpascoal/dissertacao>>.
- PEAUCELLE, D.; ARZELIER, D. An efficient numerical solution for \mathcal{H}_2 static output feedback synthesis. In: **Proc. 2001 Eur. Control Conf.** Porto, Portugal: [s.n.], 2001.
- PEAUCELLE, D.; ARZELIER, D. Ellipsoidal sets for resilient and robust static output-feedback. **IEEE Trans. Autom. Control**, v. 50, p. 899–904, June 2005.
- PIPELEERS, G. et al. Extended LMI characterizations for stability and performance of linear systems. **Systems & Control Letters**, v. 58, n. 7, p. 510–518, July 2009.

- PREMPAIN, E.; POSTLETHWAITE, I. Static output feedback stabilization with \mathcal{H}_∞ performance for a class of plants. **Syst. Control Lett.**, v. 43, p. 159–166, 2001.
- RADCLIFFE, N. J. Formal analysis and random respectful recombination. **Proceedings of 4th International Conference on Genetic Algorithms**, Morgan Kauffmann, San Mateo, CA, p. 222–229, 1991.
- RAO, S. **Engineering Optimization: Theory and Practice: Fourth Edition**. [S.l.]: John Wiley and Sons, 2009. ISBN 9780470183526.
- ROSINOVA, D.; VESELY, V.; KUCERA, V. A necessary and sufficient condition for static output feedback stabilizability of linear discrete-time systems. **KYBERNETIKA**, v. 39, p. 447–459, 2003.
- SCHERER, C.; WEILAND, S. **Linear Matrix Inequalities in Control**. [S.l.], 2004.
- SHAKED, U. An LPD approach to robust \mathcal{H}_2 and \mathcal{H}_∞ static output feedback design. **IEEE Trans. Autom. Control**, v. 48, p. 866–872, 1997.
- SILVA, F. H. et al. Função energia generalizada de controle para estabilização de sistemas não lineares. **Sba: Controle & Automação Sociedade Brasileira de Automatica**, SciELO Brasil, v. 20, n. 2, p. 133–145, 2009.
- SKELTON, R. E.; IWASAKI, T.; GRIGORIADIS, K. **A Unified Algebraic Approach to Linear Control Design**. [S.l.]: Bristol, PA : Taylor & Francis, 1998. ISBN 0748405925 9780748405923.
- SOUZA, C. E. de; TROFINO, A. An LMI approach to stabilization of linear discrete-time periodic systems. **Int. J. Control**, v. 73, p. 696–703, 2000.
- STURM, J. F. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones: *Optim. Method Softw.* v. 11–12, p. 625–653, 1999. Disponível em: <<http://sedumi.mcmaster.ca/>>.
- SYRMOS, V. et al. Static output feedback: A survey. **Automatica**, v. 33, p. 125–137, 1997.
- TALBI, E.-G. **Metaheuristics : from design to implementation**. Hoboken (N.J.): J. Wiley & Sons, 2009. ISBN 978-0-470-27858-1. Disponível em: <<http://opac.inria.fr/record=b1128380>>.
- TROFINO, A.; BARBOSA, K.; COUTINHO, D. Controle robusto. **Apostila da disciplina de controle robusto do curso de pós-graduação de engenharia elétrica da UFSC**, 2000.
- TROFINO, A.; COUTINHO, D.; BARBOSA, K. Improved \mathcal{H}_2 and \mathcal{H}_∞ conditions for robust analysis and control synthesis of linear systems. **Controle y Automacao**, Sociedade Brasileira de Automatica, v. 16, n. 4, p. 427–434, 10 2005. ISSN 0103-1759.
- TROFINO-NETO, A.; KUCERA, V. Stabilization via static output feedback. **IEEE Trans. Autom. Control**, v. 38, p. 764–765, 1993.