

UNIVERSIDADE FEDERAL DE MINAS GERAIS
Escola de Engenharia
Programa de Pós-Graduação em Engenharia Elétrica

Lucas Jorge Caldeira Carvalho

ABORDAGENS NEBULOSAS NO CONTROLE DE DRONE QUADROTOR

Belo Horizonte
2022

Lucas Jorge Caldeira Carvalho

ABORDAGENS NEBULOSAS NO CONTROLE DE DRONE QUADROTOR

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Víctor Costa da Silva Campos

Belo Horizonte
2022

C331a Carvalho, Lucas Jorge Caldeira.
Abordagens nebulosas no controle de drone quadrotor [recurso eletrônico] / Lucas Jorge Caldeira Carvalho. - 2022.
1 recurso online (110 f. : il., color.) : pdf.

Orientador: Víctor Costa da Silva Campos.

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Apêndice: f. 110.

Bibliografia: f. 105-109.
Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia elétrica - Teses. 2. Algoritmos genéticos - Teses. 3. Aeronave não tripulada - Teses. I. Campos, Víctor Costa da Silva. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 621.3(043)

Ficha catalográfica elaborada pela bibliotecária Roseli Alves de Oliveira CRB/6 2121
Biblioteca Prof. Mário Werneck, Escola de Engenharia da UFMG



UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
Programa de Pós-Graduação em Engenharia Elétrica

**ATA DA 1256ª DEFESA DE DISSERTAÇÃO DE MESTRADO
DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

ATA DE DEFESA DE DISSERTAÇÃO DE MESTRADO do aluno **Lucas Jorge Caldeira Carvalho** - registro de matrícula de número 2020681530. Às 09:30 horas do dia 20 do mês de janeiro de 2022, reuniu-se na Escola de Engenharia da UFMG a Comissão Examinadora da DISSERTAÇÃO DE MESTRADO para julgar, em exame final, o trabalho intitulado "**Abordagens Nebulosas No Controle de Drone Quadrotor**" da Área de Concentração em Sinais e Sistemas, Linha de Pesquisa Modelagem, Análise e Controle de Sistemas Não Lineares. O Prof. Víctor Costa da Silva Campos, orientador do aluno, abriu a sessão apresentando os membros da Comissão e, dando continuidade aos trabalhos, informou aos presentes que, de acordo com o Regulamento do Programa no seu Art. 8.16, será considerado APROVADO na defesa da Dissertação de Mestrado o candidato que obtiver a aprovação unânime dos membros da Comissão Examinadora. Em seguida deu início à apresentação do trabalho pelo Candidato. Ao final da apresentação seguiu-se a arguição do candidato pelos examinadores. Logo após o término da arguição a Comissão Examinadora se reuniu, sem a presença do Candidato e do público, e elegeu o Prof. Víctor Costa da Silva Campos para presidir a fase de avaliação do trabalho, constituída de deliberação individual de APROVAÇÃO ou de REPROVAÇÃO e expedição do resultado final. As deliberações individuais de cada membro da Comissão Examinadora foram as seguintes:

Membro da Comissão Examinadora	Instituição de Origem	Deliberação	Assinatura
Prof. Dr. Víctor Costa da Silva Campos - Orientador	(UFMG)	Aprovado	
Prof. Dr. Renan Lima Pereira	Departamento de Sistemas e Controle (Instituto Tecnológico d	Aprovado	
Prof. Dr. Guilherme Vianna Raffo	DELT (UFMG)	Aprovado	

Tendo como base as deliberações dos membros da Comissão Examinadora a Dissertação de Mestrado foi aprovada..... O resultado final de aprovação..... foi comunicado publicamente ao Candidato pelo Presidente da Comissão, ressaltando que a obtenção do Grau de Mestre em ENGENHARIA ELÉTRICA fica condicionada à entrega do TEXTO FINAL da Dissertação de Mestrado. O Candidato terá um prazo máximo de 30 (trinta) dias, a partir desta data, para fazer as CORREÇÕES DE FORMA e entregar o texto final da Dissertação de Mestrado na secretaria do PPGEE/UFMG. As correções de forma exigidas pelos membros da Comissão Examinadora deverão ser registradas em um exemplar do texto da Dissertação de Mestrado, cuja verificação ficará sob a responsabilidade do Presidente da Banca Examinadora. Nada mais havendo a tratar o Presidente encerrou a reunião e lavrou a presente ATA, que será assinada pelo Presidente da Comissão Examinadora. Belo Horizonte, 20 de janeiro de 2022.

ASSINATURA DO PRESIDENTE DA COMISSÃO EXAMINADORA

Em 1944, o Brasil enviou para a Europa 25.334 homens para lutarem na Segunda Guerra Mundial.

Muitos voltaram para contar suas histórias.

Outros morreram para fazê-la.

—Sabaton, *Smoking Snakes*

AGRADECIMENTOS

A **Deus**, por ter me dado a vida e, principalmente, por dar sentido a ela enchendo-a de pessoas especiais. Agradeço também, e sobretudo, pelas dificuldades impostas, pois, como disse Caio Fernando Abreu, não peço por cargas mais leves, e sim por ombros mais fortes.

A **essas pessoas especiais**, de valor imensurável, pelo apoio que sempre me deram durante toda a minha vida. Gostaria de citar, dentre elas, minha **vó "Ziza"**, **meus pais**, minha **tia "Tê"** e meu **tio Adilson**, e meu **tio "Zazi"** e minha **tia Ana**, que abraçaram meu sonho de estudar e me deram todo o apoio e suporte para realizá-lo. Agradeço também a minha namorada, **Ariel Estevam**, por estar sempre ao meu lado, acreditando e lutando comigo. Não poderia faltar também, quando se trata de incentivo e companheirismo, aquele agradecimento especial ao **DM -4°**.

Deixo um grande e especial agradecimento ao meu orientador, **Víctor Costa da Silva Campos**, pelo incentivo e dedicação, sempre disponível, com a maior boa vontade do mundo, para me atender. Agradeço também aos **membros da banca**, por aceitarem participar desse momento especial.

A felicidade só é real quando compartilhada.

—Christopher McCandless, *Na natureza selvagem*

O interesse em relação às pesquisas sobre veículos aéreos não tripulados (VANTs) vem aumentando de forma significativa nos últimos anos. O quadrotor é um tipo de VANT que possui diversas vantagens e aplicações. O acoplamento dinâmico e o comportamento altamente não linear impõem um caráter desafiador ao controle dessas aeronaves. Neste trabalho, sistemas nebulosos são usados para obter controladores não lineares capazes de resolver problemas de rastreamento de trajetória. Duas abordagens foram desenvolvidas: uma usando sistemas nebulosos como aproximadores universais para projeto de controladores nebulosos adaptativos diretos, em que uma nova proposta usando diferenciador robusto e exato é usada para estimar parâmetro não conhecido da lei de adaptação e algoritmo genético para otimizar os parâmetros encontrados com a sintonia manual; outra usando transformação do produto tensorial, técnica numérica baseada na decomposição em valores singulares de alta ordem (HOSVD), para encontrar um modelo nebuloso que aproxima o modelo do quadrotor, e, de posse dele, desenvolver um controlador PDC (compensação paralela distribuída) com estabilidade exponencial.

Palavras-chave: Sistemas Nebulosos, Transformação do Produto Tensorial, Controle Adaptativo, Algoritmo Genético, Quadrotor.

ABSTRACT

The interest in research of unmanned aerial vehicles (UAVs) has increased significantly in recent years. The quadcopter is a type of UAV that has several advantages and applications. The dynamic coupling and the highly nonlinear behavior impose a challenging character to the control of these aircrafts. In this project, fuzzy systems are used to obtain non-linear controllers capable of solving trajectory tracking problems. Two approaches were developed: first using fuzzy systems as universal approximators for designing direct adaptive fuzzy controllers, in which a new proposal using a robust and exact differentiator is used to estimate an unknown adaptation law parameter and genetic algorithm to optimize the parameters found with manual tuning; another using tensor product transformation, a numerical technique based on higher order singular value decomposition (HOSVD), to find a fuzzy model that approximates the quadcopter model, and, in possession of it, develop a PDC (parallel distributed compensation) controller with exponential stability.

Keywords: Fuzzy Systems, Tensor Product Model Transformation, Adaptive Control, Genetic Algorithm, Quadcopter.

LISTA DE FIGURAS

Figura 1 – Diagrama de corpo livre do quadrotor.	21
Figura 2 – Associação entre variações das velocidades angulares de cada rotor e movimentos lineares e angulares da aeronave.	22
Figura 3 – Dados sobre opiniões de pessoas em relação aos filmes.	37
Figura 4 – Matrizes modo- n de um tensor de 3ª ordem.	39
Figura 5 – Exemplo de HOSVD de um tensor de 3ª ordem.	41
Figura 6 – Implicação geométrica das convexidades das funções de ponderação no fecho convexo resultante.	52
Figura 7 – Funções de ponderação obtidas após a aplicação do HOSVD.	55
Figura 8 – Funções de ponderação obtidas após a redução de regras e aplicação de propriedades de convexificação.	56
Figura 9 – Ilustração do método de Lyapunov para investigar estabilidade.	61
Figura 10 – Representação de pontos de operação de um controlador PDC.	65
Figura 11 – Derivadas real e estimada pelo RED da função $f(t) = 5t + \text{sen}(t)$ na ausência (a) e na presença de ruído (c); derivadas real e estimada pelo RED da função $f(t) = 5t + \text{sen}(t) + 0,001\cos(30t)$ na ausência (b) e na presença de ruído (d).	80
Figura 12 – Fluxograma de um AG.	82
Figura 13 – Esquema da metodologia seguida com os controladores PDC.	91
Figura 14 – Esquema da metodologia seguida com os controladores adaptativos.	92
Figura 15 – Funções de pertinência do modelo sem simplificação; (a) $\mathbf{u}_{1,1}$ (azul) e $\mathbf{u}_{2,1}$ (vermelho); (b) $\mathbf{u}_{1,2}$ (azul), $\mathbf{u}_{2,2}$ (vermelho), $\mathbf{u}_{3,2}$ (amarelo), $\mathbf{u}_{4,2}$ (roxo) e $\mathbf{u}_{5,2}$ (verde); (c) $\mathbf{u}_{1,3}$ (azul) e $\mathbf{u}_{2,3}$ (vermelho); (d) $\mathbf{u}_{1,4}$ (azul) e $\mathbf{u}_{2,4}$ (vermelho); (e) $\mathbf{u}_{1,5}$ (azul), $\mathbf{u}_{2,5}$ (vermelho) e $\mathbf{u}_{3,5}$ (amarelo); (f) $\mathbf{u}_{1,5}$ (azul), $\mathbf{u}_{2,5}$ (vermelho) e $\mathbf{u}_{3,5}$ (amarelo).	94
Figura 16 – Funções de pertinência do modelo simplificado; (a) $\mathbf{u}_{1,1}$ (azul) e $\mathbf{u}_{2,1}$ (vermelho); (b) $\mathbf{u}_{1,2}$ (azul), $\mathbf{u}_{2,2}$ (vermelho), $\mathbf{u}_{3,2}$ (amarelo); (c) $\mathbf{u}_{1,3}$ (azul) e $\mathbf{u}_{2,3}$ (vermelho); (d) $\mathbf{u}_{1,4}$ (azul); (e) $\mathbf{u}_{1,5}$ (azul) e $\mathbf{u}_{2,5}$ (vermelho); (f) $\mathbf{u}_{1,5}$ (azul) e $\mathbf{u}_{2,5}$ (vermelho).	95
Figura 17 – Rastreamento de trajetória com o controlador PDC.	96
Figura 18 – Sinais de controle do controlador PDC.	97
Figura 19 – Comparação entre as saídas duas abordagens adaptativas.	98
Figura 20 – Comparação entre as entradas de controle das duas abordagens adaptativas.	99

Figura 21 – Comparação entre as convergências dos AG implementados; com probabilidade de mutação fixa (esquerda); com probabilidade de mutação variável e população inicial expandida (direita).	99
Figura 22 – Comparação entre as saídas dos controladores com sintonia manual e com sintonia via AG.	100
Figura 23 – Comparação entre as entradas de controle dos controladores com sintonia manual e com sintonia com o uso de AG.	101

LISTA DE ALGORITMOS

Algoritmo 1 – PSEUDO-CÓDIGO DO ALGORITMO GENÉTICO UTILIZADO NESTE TRABA- BALHO	85
---	----

LISTA DE ABREVIATURAS

AG	Algoritmo Genético
CNO	Próxima Normalizada
FS	Acrônimo para a expressão em inglês <i>Fuzzy System</i> (em português: Sistema Nebuloso)
GDL	Graus de Liberdade
HOSVD	Acrônimo para a expressão em inglês <i>Higher Order Singular Value Decomposition</i> (em português: Decomposição em Valores Singulares de Alta Ordem)
INO	Normalizada Inversa
LIT	Linear Invariante no Tempo
LMI	Acrônimo para a expressão em inglês <i>Linear Matrix Inequality</i> (em português: Desigualdade Matricial Linear)
MIMO	Acrônimo para a expressão em inglês <i>Multiple Input Multiple Output</i> (em português: Múltiplas Entradas, Múltiplas Saídas)
MISO	Acrônimo para a expressão em inglês <i>Multiple Input Single Output</i> (em português: Múltiplas Entradas, Saída Única)
NN	Não Negativa
NO	Normalizada
PDC	Acrônimo para a expressão em inglês <i>Parallel Distributed Compensation</i> (em português: Compensação Paralela Distribuída)
qLPV	Acrônimo para a expressão em inglês <i>quasi-Linear Parameter Varying</i> (em português: quasi-Linear a Parâmetros Variantes)
RED	Acrônimo para a expressão em inglês <i>Robust Exact Differentiator</i> (em português: Diferenciador Robusto e Exato)
RMSE	Acrônimo para a expressão em inglês <i>Root Mean Squared Error</i> (em português: Raiz do Erro Quadrático Médio)
RNO	Normalizada Relaxada
SN	Soma Normalizada
SVD	Acrônimo para a expressão em inglês <i>Singular Value Decomposition</i> (em português: Decomposição em Valores Singulares)
TORA	Acrônimo para a expressão em inglês <i>Translational Oscillations by a Rotational Actuator</i> (em português: Oscilações Translacionais por um Atuador Rotacional)
TP	Acrônimo para a expressão em inglês <i>Tensorial Product</i> (em português: Produto Tensorial)
TS	Takagi-Sugeno
VANT	Veículo Aéreo Não Tripulado

LISTA DE TABELAS

Tabela 1 – Parâmetros do quadrotor.	87
Tabela 2 – Parâmetros usados na abordagem com três controladores adaptativos.	97
Tabela 3 – Parâmetros usados na abordagem com quatro controladores adaptativos.	97
Tabela 4 – Parâmetros encontrados pelo AG no projeto com quatro controladores adaptativos.	100

Lista de Figuras	8
Lista de Algoritmos	10
Lista de Abreviaturas	11
Lista de Tabelas	12
1 Introdução	16
1.1 Motivação	16
1.2 Objetivos	17
1.3 Organização do trabalho	17
1.4 Notações	18
I Referencial Teórico	19
2 Modelagem do Quadrotor	20
2.1 Princípio de Funcionamento de um Quadrotor	20
2.2 Cinemática	22
2.3 Dinâmica	24
2.3.1 Dinâmica Translacional	24
2.3.2 Dinâmica Rotacional	26
2.4 Modelos Completo e Simplificado	27
2.5 Sensores e Atuadores da Aeronave	28
3 Sistemas Nebulosos	29
3.1 Conjuntos Nebulosos	29
3.2 Lógica Nebulosa	30
3.3 Modelo Takagi-Sugeno	31
4 Transformação do Produto Tensorial	34
4.1 Decomposição em Valores Singulares de Tensores de Alta Ordem	35
4.1.1 Conceitos Preliminares	38
4.1.2 Técnica da Decomposição em Valores Singulares de Alta Ordem	40
4.2 Modelos TP	42
4.3 Transformação do produto tensorial	45
4.3.1 Discretização	46
4.3.2 Aplicação do HOSVD	47

SUMÁRIO	14
4.3.3	Representações Convexas 48
4.3.3.1	Soma Normalizada e Não Negatividade 49
4.3.3.2	Normalizada e Próxima Normalizada 50
4.3.3.3	Normalizada Relaxada e Normalizada Inversa 51
4.3.4	Aproximação para as Matrizes de Peso 53
5	Estabilidade de Sistemas Dinâmicos via Desigualdades Matriciais Lineares 57
5.1	Desigualdades Matriciais Lineares 57
5.2	Teoria de Lyapunov 59
5.3	Análise da Estabilidade via LMIs 62
5.4	Programas para Resolução de LMIs 62
6	Controlador PDC 63
6.1	Lei de controle 63
6.2	Conjunto de LMIs para estabilidade assintótica 65
6.3	Rastreamento de trajetória 67
7	Controlador Nebuloso Adaptativo Direto 69
7.1	Considerações Iniciais 70
7.2	Lei de Controle 72
7.3	Lei de Adaptação 73
7.4	Convergência do Erro de Rastreamento 76
8	Diferenciador Robusto e Exato 79
9	Algoritmo Genético 81
II	Projeto 86
10	Metodologia 87
10.1	Controlador PDC 88
10.2	Controlador Nebuloso Adaptativo Direto 91
11	Resultados 93
11.1	Controlador PDC 93
11.2	Controladores Adaptativos Diretos 96
III	Discussões 102
12	Conclusão 103

13 Trabalhos Futuros	104
Referências Bibliográficas	105
Anexo A	110

Resumo do capítulo

O interesse em relação às pesquisas sobre VANTs, veículos aéreos não tripulados, vem aumentando de forma significativa nos últimos anos. Um representante dessa classe de veículos é o quadrotor, aeronave que é elevada e impulsionada por quatro rotores, cujo acoplamento dinâmico e o comportamento altamente não linear impõem um carácter desafiador ao seu controle. O presente trabalho objetiva utilizar sistemas nebulosos para encontrar controladores capazes de realizar o controle de rastreamento de trajetória de um quadrotor.

1.1 MOTIVAÇÃO

O interesse em relação às pesquisas sobre VANTs, veículos aéreos não tripulados, vem aumentando de forma significativa nos últimos anos, principalmente devido à vasta gama de aplicações que vão desde o uso para o entretenimento até a utilização no campo civil e militar (RESENDE et al., 2018).

O quadrotor, quadricóptero, ou quadrrrotor, é um tipo de VANT que é elevado e impulsionado por quatro rotores (RABAH et al., 2018). Ele possui a vantagem de locomoção em locais com pouco espaço, capacidade de realizar manobras precisas em baixa e alta velocidade, pode levantar voo vertical, aterrissar em locais de difícil acesso, além de possuir a capacidade de realizar voos pairados (MORAIS, 2017). A menor complexidade mecânica destas aeronaves em relação a outros modelos resulta em facilidades de construção e manutenção. Por outro lado, o acoplamento dinâmico e o comportamento altamente não linear impõem um carácter desafiador ao controle dos quadrotores (BENIGNO, 2015).

Sistemas inteligentes, como a lógica nebulosa, são considerados muito bem-sucedidos em lidar com sistemas complexos, não lineares e variantes no tempo. Técnicas de controle inteligente, como sistemas nebulosos, redes neurais artificiais e algoritmos genéticos, têm mostrado capacidade de lidar com sistemas não lineares na presença de incertezas (AL-MAHTURI et al., 2018).

Há na literatura diversas pesquisas que contribuem para o estudo de controle de quadrotores utilizando sistemas nebulosos. (ABRO et al., 2021) propõem um controle de modo deslizante nebuloso para controlar a atitude e a posição dessa classe de aeronaves. (SHIRZADEH et al., 2020) propõem uma estratégia de controle adaptativo robusta baseada em uma rede neural nebulosa do tipo-2 para rastreamento de trajetórias de um quadrotor. (LEEL; KIM, 2014) e (YACEF et al., 2012) apresentam um controle robusto via desigualdades matriciais lineares (LMIs) para um quadrotor usando modelo nebuloso Takagi-Sugeno (TS). (BASRI; HUSAIN; DANAPALA-SINGAM, 2015) propõem uma técnica de controle híbrido, em que um controlador baseado na

técnica de *backstepping* (realimentação de estados) é o controlador principal, e um controlador de compensação contendo uma abordagem de controle nebuloso é usado para eliminar o efeito das incertezas causadas por distúrbios externos.

1.2 OBJETIVOS

O presente trabalho objetiva utilizar sistemas nebulosos para encontrar controladores capazes de realizar o controle de rastreamento de trajetória de um quadrotor em tempo contínuo. Ou seja, controladores que sejam capazes de, dada uma entrada de referência $y_d(t)$, fazer com que a saída do sistema $y(t)$ seja tal que o erro $e(t) = y_d(t) - y(t)$ esteja dentro de um limite considerado válido. Pretende-se também analisar o esforço de controle necessário para que esse propósito seja atingido, considerando-o um requisito de desempenho.

Especificamente, pretende-se:

- Projetar controladores nebulosos adaptativos diretos;
- Analisar a viabilidade da nova proposta deste trabalho de usar diferenciador robusto e exato para permitir a implementação em tempo contínuo dos controladores adaptativos;
- Implementar um algoritmo evolucionário e analisar sua eficácia no auxílio da sintonia dos controladores adaptativos;
- Obter um modelo Takagi-Sugeno (TS) do quadrotor por meio da técnica de transformação do produto tensorial;
- Usar o modelo TS encontrado para projeto de controladores via LMIs;
- Analisar o desempenho dos controladores através de simulações computacionais.

1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado em três grandes divisões, denominadas partes. Cada parte é subdividida em capítulos, totalizando, ao todo, treze deles.

A **Parte I**, composta por oito capítulos, destina-se a apresentar os conceitos necessários para o entendimento da dissertação. O **Capítulo 2** versa sobre a modelagem matemática do quadrotor, contemplando o tipo de abordagem teórica seguida, a dinâmica rotacional e translacional da aeronave e os sensores e as variáveis de controle considerados neste trabalho; Uma introdução, focada em modelos Takagi-Sugeno (TS), sobre sistemas nebulosos é feita no **Capítulo 3**, mostrando a importância, o contexto de surgimento e o que são a lógica e os conjuntos nebulosos; No **Capítulo 4**, apresenta-se a técnica de transformação do produto tensorial, que permite a representação de um sistema quasi-Linear a Parâmetros Variantes (qLPV) na forma de modelos TS, discutindo sobre a álgebra multilinear básica envolvida e a decomposição em

valores singulares de alta ordem (HOSVD), principal ferramenta matemática da técnica. Para elucidar a teoria, mostra-se um exemplo de aplicação dessa técnica em um sistema mecânico; O **Capítulo 5** expõe uma discussão sobre a estabilidade de sistemas não lineares, definindo estabilidade no sentido de Lyapunov, LMIs e como esses conceitos se relacionam; A teoria envolvida no desenvolvimento de controladores PDC (compensação paralela distribuída), um dos tipos de controlador desenvolvidos neste trabalho, é apresentada no **Capítulo 6**, expondo a lei de controle, a análise da estabilidade e mostrando como a síntese desse tipo de controlador se relaciona com a etapa da transformação do produto tensorial; Discute-se, no **Capítulo 7**, a teoria do outro tipo de controlador projetado: os controladores nebulosos adaptativos diretos. Apresenta-se a lei de controle, relacionada a sistemas de inferência Takagi-Sugeno, a lei de adaptação, baseada no método do gradiente, e analisa-se a estabilidade desses controladores e a convergência de seus parâmetros; Uma nova abordagem usando diferenciador robusto e exato (RED) para estimar derivadas desconhecidas de controladores adaptativos, permitindo suas implementações em tempo contínuo, é realizada neste trabalho. Esse dispositivo é exposto no **Capítulo 8**. Um exemplo teórico de sua aplicação é mostrado; Os conceitos relacionados ao algoritmo genético (AG) desenvolvido e implementado neste trabalho são apresentados no **Capítulo 9**.

A **Parte II**, composta por dois capítulos, destina-se a detalhar o projeto dos controladores PDC e adaptativos. O **Capítulo 10** detalha a metodologia seguida neste trabalho para projetar e validar os resultados dos controladores, e o **Capítulo 11** expõem os resultados obtidos.

A **Parte III**, que também é composta por dois capítulos, apresenta as discussões deste trabalho. No **Capítulo 12**, a conclusão embasada no entendimento crítico dos resultados é mostrada, apresentando os prováveis motivos para muitos deles. Por último, no **Capítulo 13**, novas abordagens e abordagens alternativas são apresentadas como sugestões para trabalhos futuros.

1.4 NOTAÇÕES

Abaixo encontram-se as notações usadas neste trabalho.

1. s_* = sen(*), c_* = cos(*) e t_* = tan(*);
2. $\{a, b, \dots\}$: grandezas escalares;
3. $\{\mathbf{a}, \mathbf{b}, \dots\}$: grandezas vetoriais;
4. $\{\mathbf{A}, \mathbf{B}, \dots\}$: grandezas matriciais;
5. $\{\mathcal{A}, \mathcal{B}, \dots\}$: grandezas tensoriais¹.

¹Conforme será discutido mais a frente, grandezas escalares (tensores de ordem zero), vetoriais (tensores de ordem um) e matriciais (tensores de ordem dois) são classes particulares de grandezas tensoriais.

Parte I

Referencial Teórico

MODELAGEM DO QUADROTOR

Resumo do capítulo

A modelagem seguida neste trabalho é baseada no formalismo de Euler-Lagrange. Embora o quadricóptero possua seis graus de liberdade (GDL), ele é equipado apenas com quatro hélices, não sendo, portanto, possível controlar independentemente todos eles. O modelo do quadrotor foi obtido pelo estudo da dinâmica rotacional e translacional, em separado, da aeronave. Em algumas aplicações de controle, é conveniente realizar uma simplificação do modelo, que pode ser feita pela simplificação dos pequenos ângulos. Quatro sensores podem ser usados para obter as leituras das malhas controladas.

Na literatura existem duas abordagens clássicas para a modelagem matemática de veículos aéreos: uma baseada no formalismo de Newton-Euler, que utiliza as leis de Newton para modelar o sistema, e outra no formalismo de Euler-Lagrange, que se baseia na descrição das energias cinética e potencial para realizar essa modelagem (MORAIS, 2017). Essa última abordagem será a utilizada neste trabalho.

Para modelar e encontrar as equações dinâmicas do quadrotor são definidos sistemas de referências. Considera-se E um sistema de referência definido pelos eixos x, y, z fixo e com origem na superfície terrestre, e B um sistema de referência do corpo definido pelos eixos x_B, y_B, z_B com origem no centro de massa do quadrotor (Figura 1).

A modelagem foi dividida em dois subsistemas de acordo com o tipo de movimentos analisado: rotação e translação. Essa modelagem desacoplada, que considera independência entre os movimentos de rotação e translação, foi possível porque consideramos que o centro de massa da aeronave e a origem de seu sistema de referência (B) são congruentes.

2.1 PRINCÍPIO DE FUNCIONAMENTO DE UM QUADROTOR

Embora o quadricóptero possua seis graus de liberdade (GDL), ele é equipado apenas com quatro hélices, não sendo, portanto, possível controlar independentemente todos eles (BRESCIANI, 2008). Neste trabalho, os quatro movimentos controlados são os da atitude (rolagem $[\phi]$, arfagem $[\theta]$ e guinada $[\psi]$), que representa a posição angular, e altitude $[z]$ (Figura 1). Os movimentos translacionais acabam sendo dependentes desses outros.

O quadrotor move-se devido ao trabalho realizado pelo conjunto dos quatro motores com seus respectivos propulsores, dispostos horizontalmente em formato cruciforme. Os propulsores, devido ao seu formato e à rotação dos motores aos quais estão acoplados, criam um fluxo de ar no sentido contrário ao do movimento desejado, causando, devido à terceira lei de Newton, a tração necessária ao voo. A variação coletiva das forças de propulsão ($\Omega_1, \Omega_2, \Omega_3$ e Ω_4), resultante da velocidade angular dos motores ($\omega_1, \omega_2, \omega_3$ e ω_4), governa a navegação tridimen-

sional da aeronave. Os movimentos associados a um quadrotor estão relacionados à variação de rotação de cada um dos motores. Os quatro rotores giram em sentidos opostos em pares para anular os torques de reação gerados na estrutura da aeronave, devido à terceira lei de Newton, pelas suas rotações (PAULA, 2012).

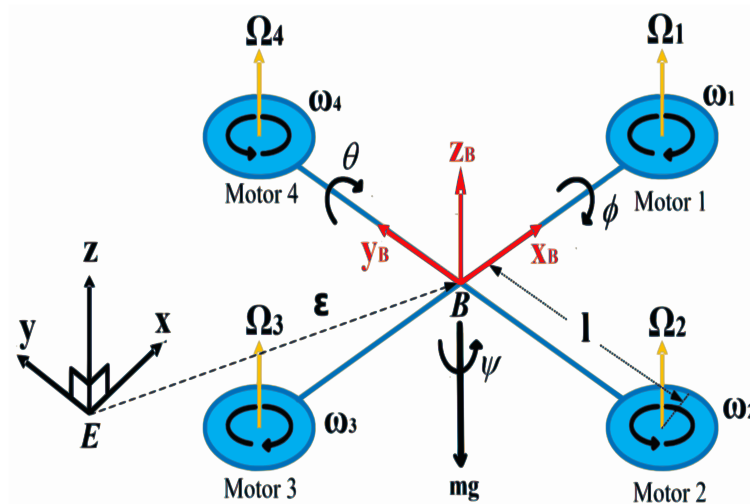


Figura 1 – Diagrama de corpo livre do quadrotor.

Fonte : Adaptado de (TSAI et al., 2021).

A Figura 2 ilustra cenários que relacionam variações das velocidades angulares dos rotores e os consequentes movimentos lineares e angulares da aeronave. São eles:

1. Ao se elevar a velocidade angular no propulsor 1 e igualmente reduzir-se a velocidade no propulsor 3, uma variação negativa será gerada na atitude arfagem e, conseqüentemente, um deslocamento negativo no eixo x será produzido (Figura 2(a)). Ao se reduzir a velocidade no propulsor 1 e aumentar no propulsor 3, uma variação positiva será gerada na atitude arfagem, o que resultará em um positivo deslocamento no eixo x (Figura 2(b));
2. Com a variação das velocidades dos propulsores 2 e 4, gera-se movimento ao longo do eixo y (Figura 2(c) e Figura 2(d)), por meio da variação da rolagem causada pelos propulsores;
3. Para elevar a altitude do quadrotor, aumenta-se igualmente a velocidade de todos os propulsores (Figura 2(e)). Para reduzir a altitude (z), a redução equilibrada das velocidades é necessária (Figura 2(f));
4. Para alterar a orientação, ou guinada, da aeronave, um desequilíbrio na somatória dos momentos deve ocorrer (Figura 2(g) e Figura 2(h))

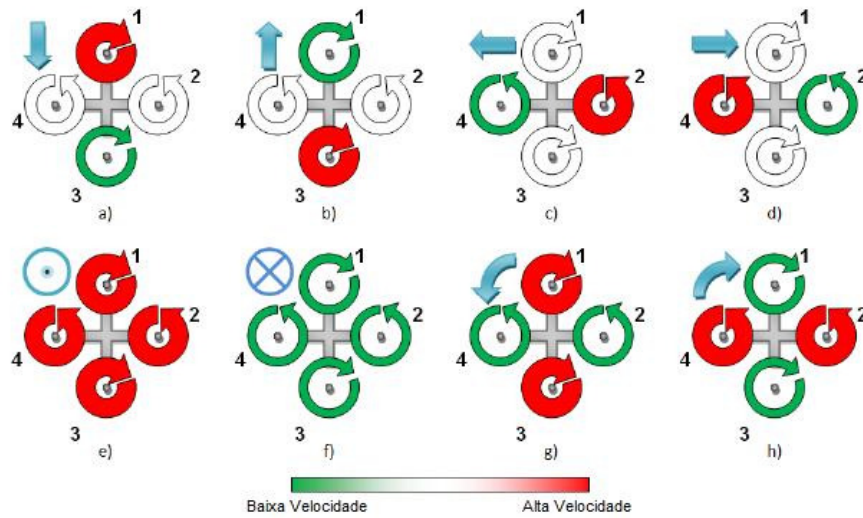


Figura 2 – Associação entre variações das velocidades angulares de cada rotor e movimentos lineares e angulares da aeronave.

Fonte : (PFEIFER, 2013).

2.2 CINEMÁTICA

A posição linear absoluta do quadricóptero é definida em relação ao referencial fixo E pelo vetor ϵ (2.1), representado na Figura 1. A atitude, ou seja, a posição angular é obtida conhecendo-se os três ângulos de Euler η_e (2.2), de modo que o sistema de coordenadas do corpo é obtido por meio de três rotações consecutivas em torno de x , y e z , respectivamente. O ângulo de rolagem (ϕ) determina a rotação em torno do eixo x ; o ângulo de arfagem (θ), a rotação em torno do eixo y ; e o ângulo ψ , em torno do eixo z . O vetor \mathbf{q} (2.3), chamado de vetor de coordenadas generalizadas, acopla as informações sobre a posição linear e angular da aeronave (LUUKKONEN, 2011).

$$\epsilon = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad (2.1)$$

$$\eta_e = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}, \quad (2.2)$$

$$\mathbf{q} = \begin{bmatrix} \epsilon \\ \eta_e \end{bmatrix}. \quad (2.3)$$

Para que se obtenha o equacionamento completo da máquina, é necessário relacionar o comportamento dinâmico do referencial móvel com o fixo. A mudança da base B para base E é feita por uma matriz de rotação. Esse tipo de matriz transforma o conjunto de coordenadas que representa um objeto em um conjunto cartesiano ortogonal, sem alterar sua forma ou tamanho

(EVANS, 2001). Elas são amplamente utilizadas para descrever a orientação de qualquer corpo móvel devido à facilidade da utilização da álgebra matricial e são obtidas por meio de três rotações cuja ordem importa.

Considerando-se um instante inicial, em que a origem de B coincide com a origem de E , x coincide com x_B , y com y_B e z com z_B , obtém-se, por meio de três rotações ordenadas, as seguintes matrizes de rotação:

1. $\mathbf{R}(x, \phi)$: rotação de um ângulo ϕ em torno do eixo x :

$$\mathbf{R}(x, \phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi \end{bmatrix}; \quad (2.4)$$

2. $\mathbf{R}(y, \theta)$: rotação de um ângulo θ em torno do eixo y :

$$\mathbf{R}(y, \theta) = \begin{bmatrix} c_\theta & 0 & -s_\theta \\ 0 & 1 & 0 \\ s_\theta & 0 & c_\theta \end{bmatrix}; \quad (2.5)$$

3. $\mathbf{R}(z, \psi)$: rotação de um ângulo ψ em torno do eixo z :

$$\mathbf{R}(z, \psi) = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.6)$$

em que $c_* = \cos(*)$ e $s_* = \sin(*)$.

A matriz de rotação do sistema de coordenadas do corpo para o sistema de coordenadas fixo (B para E) é a matriz de rotação completa \mathbf{R} , que é obtida mediante o produto das três matrizes de rotação anteriores,

$$\mathbf{R} = \mathbf{R}(x, \phi) \cdot \mathbf{R}(y, \theta) \cdot \mathbf{R}(z, \psi) = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi + s_\psi c_\phi & c_\psi s_\theta c_\phi - s_\psi s_\phi \\ -s_\psi c_\theta & -s_\psi s_\theta s_\phi + c_\psi c_\theta & -s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}. \quad (2.7)$$

Sendo assim, a relação das coordenadas de um ponto genérico $P(x, y, z)$ no sistema de coordenadas fixo com suas coordenadas no sistema de coordenadas móvel é obtida com o uso da matriz de rotação completa:

$$P(x, y, z) = \mathbf{R} \cdot P(x_B, y_B, z_B).$$

2.3 DINÂMICA

As equações foram desenvolvidas utilizando as considerações feitas por (BOUABDALLAH, 2007):

- A estrutura é de um corpo rígido;
- A estrutura é simétrica;
- O centro de gravidade coincide com o centro do corpo inercial;
- As hélices são rígidas.

A equação de Euler-Lagrange, ponto de partida para a modelagem proposta neste trabalho, é baseada na energia cinética e na energia potencial de um sistema em termo de suas coordenadas generalizadas (BOUABDALLAH, 2007):

$$\mathbf{T} = \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}}, \quad (2.8)$$

em que:

- $\mathcal{L}(\dot{\mathbf{q}}, \mathbf{q})$: função de Lagrange ou Lagrangiana ($E_{cinética} - E_{potencial}$);
- $E_{cinética}$: energia cinética total, dada pela soma das energias cinéticas de translação com a energia cinética de rotação ($E_{trans} + E_{rot}$);
- $E_{potencial}$: energia potencial total;
- \mathbf{q} : coordenada generalizada;
- $\dot{\mathbf{q}}$: velocidade generalizada;
- \mathbf{T} : conjunto de forças generalizadas (forças translacionais $[\mathbf{F}_e]$ e torques generalizados $[\boldsymbol{\tau}_{\eta_e}]$) que regem o sistema.

2.3.1 DINÂMICA TRANSLACIONAL

Sendo m a massa do quadrotor, a energia cinética relacionada ao movimento translacional do veículo é dada por

$$E_{trans} = \frac{m}{2} \dot{\boldsymbol{\epsilon}}^T \dot{\boldsymbol{\epsilon}} = \frac{m}{2} (\dot{x}^2 + \dot{y}^2 + \dot{z}^2). \quad (2.9)$$

A energia potencial está relacionada à altitude da aeronave. Sendo g o valor da gravidade:

$$E_{potencial} = mgz. \quad (2.10)$$

A Lagrangiana translacional é dada por

$$\mathcal{L}_{trans} = E_{trans} - E_{potencial}. \quad (2.11)$$

Em relação à dinâmica translacional, (2.8) é reescrita como

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}_{trans}}{\partial \dot{\epsilon}} \right) - \frac{\partial \mathcal{L}_{trans}}{\partial \epsilon} = \mathbf{F}_{\epsilon}. \quad (2.12)$$

Substituindo (2.11) em (2.12) e resolvendo as derivadas parciais, obtemos:

$$m\ddot{\epsilon} + mg \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{F}_{\epsilon}, \quad (2.13)$$

em que \mathbf{F}_{ϵ} é o vetor de força de translação aplicada à aeronave causada pelas entradas de controle. Esse vetor pode ser obtido pela transformação, utilizando a matriz de rotação, do vetor de força linear $\hat{\mathbf{F}}$ aplicada a cada hélice causada pela rotação das mesmas:

$$\mathbf{F}_{\epsilon} = \mathbf{R}\hat{\mathbf{F}}. \quad (2.14)$$

Ignorando-se pequenas forças causadas por campos elétricos e campos magnéticos, que geralmente são de magnitude muito menor do que as entradas de controle, temos que (CASTILLO; LOZANO; DZUL, 2005)

$$\hat{\mathbf{F}} = \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix}, \quad (2.15)$$

com $U_1 = \Omega_1 + \Omega_2 + \Omega_3 + \Omega_4$ (Figura 1). A força Ω_i produzida em cada motor depende de uma constante $k_i > 0$, obtida experimentalmente e que relaciona a velocidade angular de cada motor com a força produzida por essa rotação, e de sua respectiva velocidade angular ω_i , $i = 1, 2, 3, 4$,

$$\Omega_i = k_i \omega_i^2. \quad (2.16)$$

Usando-se (2.15), (2.14) e (2.13), chega-se no seguinte modelo dinâmico para o movimento de translação do quadrotor (BENIGNO, 2015):

$$\begin{aligned} \ddot{x} &= (c_{\psi}s_{\theta}c_{\phi} + s_{\psi}s_{\theta}) \frac{U_1}{m}, \\ \ddot{y} &= (s_{\psi}s_{\theta}c_{\phi} - c_{\psi}s_{\theta}) \frac{U_1}{m}, \end{aligned} \quad (2.17)$$

$$\ddot{z} = -g + (c_{\theta}c_{\phi}) \frac{U_1}{m}.$$

g é a gravidade, m a massa do quadricóptero e $U_1 = \Omega_1 + \Omega_2 + \Omega_3 + \Omega_4$.

2.3.2 DINÂMICA ROTACIONAL

A energia rotacional total do sistema é

$$E_{rot} = \frac{1}{2} \dot{\boldsymbol{\eta}}_e^T \mathbf{J}(\boldsymbol{\eta}_e) \dot{\boldsymbol{\eta}}_e, \quad (2.18)$$

em que $\mathbf{J}(\boldsymbol{\eta}_e)$ é a Jacobiana das coordenadas generalizadas. Ela pode ser associada à Jacobiana $\mathbf{w}_{\boldsymbol{\eta}_e}$ pelo tensor de inércias do corpo, \mathbf{I}_c ,

$$\mathbf{J}(\boldsymbol{\eta}_e) = \mathbf{w}_{\boldsymbol{\eta}_e}^T \mathbf{I}_c \mathbf{w}_{\boldsymbol{\eta}_e}. \quad (2.19)$$

$\mathbf{w}_{\boldsymbol{\eta}_e}$ relaciona as velocidades angulares da aeronave no referencial não inercial com a taxa de variação dos ângulos de Euler:

$$\mathbf{w}_{\boldsymbol{\eta}_e} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & c_\theta s_\phi \\ 0 & -s_\phi & c_\theta c_\phi \end{bmatrix}. \quad (2.20)$$

Como a estrutura do quadrotor considerada é simétrica, temos o seguinte tensor de inércia

$$\mathbf{I}_c = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \quad (2.21)$$

I_{xx} , I_{yy} e I_{zz} são os momentos de inércia do veículo em torno, respectivamente, dos eixos x_B , y_B e z_B . Por simetria, $I_{xx} = I_{yy}$.

Para rotação, a equação de Euler-Lagrange fica

$$\frac{d}{dt} \left(\frac{\partial L_{rot}}{\partial \dot{\boldsymbol{\eta}}_e} \right) - \frac{\partial L_{rot}}{\partial \boldsymbol{\eta}_e} = \boldsymbol{\tau}_{\boldsymbol{\eta}_e}. \quad (2.22)$$

$\boldsymbol{\tau}_{\boldsymbol{\eta}_e}$ é o vetor de torques, definido como

$$\boldsymbol{\tau}_\eta = \begin{bmatrix} lU_2 \\ lU_3 \\ U_4 \end{bmatrix}, \quad (2.23)$$

sendo l a distância do centro de massa do quadrotor à cada hélice, $U_2 = \Omega_3 - \Omega_1$, $U_3 = \Omega_2 - \Omega_4$ e $U_4 = l(\Omega_2 + \Omega_4 - \Omega_1 - \Omega_3)$.

A equação dinâmica que rege o quadrotor é dada por (BENIGNO, 2015)

$$\mathbf{J}(\boldsymbol{\eta}_e) \ddot{\boldsymbol{\eta}} + \mathbf{C}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) \dot{\boldsymbol{\eta}} = \boldsymbol{\tau}_\eta. \quad (2.24)$$

$\mathbf{C}(\boldsymbol{\eta}_e, \dot{\boldsymbol{\eta}}_e)$ é a matriz de Coriolis, que define os efeitos giroscópicos e o sistema centrípeto (LUUKKONEN, 2011).

A partir de (2.18) a (2.24), e usando a notação $t_* = \tan(*)$, obtêm-se as equações da dinâmica rotacional (BENIGNO, 2015)

$$\begin{aligned}\ddot{\phi} &= -\dot{\psi}\dot{\theta}c_\phi + \frac{lc_\psi}{I_{xx}}U_2 - \frac{ls_\psi}{I_{xx}}U_3 + \frac{I_{yy}-I_{zz}}{I_{xx}}(\dot{\psi} - \dot{\theta}s_\phi)\dot{\theta}c_\phi, \\ \ddot{\theta} &= \frac{\dot{\psi}\dot{\phi}}{c_\phi} + \dot{\phi}\dot{\theta}t_\phi + \frac{ls_\psi}{I_{yy}c_\phi}U_2 + \frac{lc_\psi}{I_{yy}c_\phi}U_3 - \frac{I_{yy}-I_{zz}}{I_{yy}}(\dot{\psi} - \dot{\theta}s_\phi)\frac{\dot{\phi}}{c_\phi}, \\ \ddot{\psi} &= \dot{\psi}\dot{\phi}t_\phi + \frac{\dot{\phi}\dot{\theta}}{c_\phi} + \frac{ls_\psi t_\phi}{I_{yy}}U_2 + \frac{lc_\psi t_\phi}{I_{yy}}U_3 + \frac{1}{I_{zz}}U_4 - \frac{I_{yy}-I_{zz}}{I_{yy}}(\dot{\psi} - \dot{\theta}s_\phi)\dot{\phi}t_\phi.\end{aligned}\quad (2.25)$$

2.4 MODELOS COMPLETO E SIMPLIFICADO

Em resumo, o modelo do quadrotor é obtido pelo acoplamento das equações que regem a dinâmica translacional (2.17) com as equações que regem a dinâmica rotacional (2.25). Nesse trabalho, que trata do controle da aeronave, é interessante destacar, conforme visto abaixo, as dinâmicas relacionadas as malhas que serão controladas:

$$\begin{aligned}\ddot{z} &= -g + (c_\theta c_\phi)\frac{U_1}{m}, \\ \ddot{\phi} &= -\dot{\psi}\dot{\theta}c_\phi + \frac{lc_\psi}{I_{xx}}U_2 - \frac{ls_\psi}{I_{xx}}U_3 + \frac{I_{yy}-I_{zz}}{I_{xx}}(\dot{\psi} - \dot{\theta}s_\phi)\dot{\theta}c_\phi, \\ \ddot{\theta} &= \frac{\dot{\psi}\dot{\phi}}{c_\phi} + \dot{\phi}\dot{\theta}t_\phi + \frac{ls_\psi}{I_{yy}c_\phi}U_2 + \frac{lc_\psi}{I_{yy}c_\phi}U_3 - \frac{I_{yy}-I_{zz}}{I_{yy}}(\dot{\psi} - \dot{\theta}s_\phi)\frac{\dot{\phi}}{c_\phi}, \\ \ddot{\psi} &= \dot{\psi}\dot{\phi}t_\phi + \frac{\dot{\phi}\dot{\theta}}{c_\phi} + \frac{ls_\psi t_\phi}{I_{yy}}U_2 + \frac{lc_\psi t_\phi}{I_{yy}}U_3 + \frac{1}{I_{zz}}U_4 - \frac{I_{yy}-I_{zz}}{I_{yy}}(\dot{\psi} - \dot{\theta}s_\phi)\dot{\phi}t_\phi.\end{aligned}\quad (2.26)$$

Em algumas aplicações de controle, é conveniente realizar uma simplificação do modelo, que pode ser feita pela simplificação dos pequenos ângulos (YOUNES et al., 2014; KHALIFA et al., 2013; WANG et al., 2016; MICHAEL et al., 2010). No presente trabalho, essa simplificação foi feita com $\sin(*) \approx *$ e $\cos(*) \approx 1$ e não foi aplicada à malha da altitude, pois a faria perder a dependência em relação aos estados.

$$\begin{aligned}\ddot{z} &= -g + (c_\theta c_\phi)\frac{U_1}{m}, \\ \ddot{\phi} &= -\dot{\psi}\dot{\theta} + \frac{l}{I_{xx}}U_2 - \frac{l\psi}{I_{xx}}U_3 + \frac{I_{yy}-I_{zz}}{I_{xx}}(\dot{\psi} - \dot{\theta})\dot{\theta}, \\ \ddot{\theta} &= \dot{\psi}\dot{\phi} + \dot{\phi}\dot{\theta} + \frac{l\psi}{I_{yy}}U_2 + \frac{l}{I_{yy}}U_3 - \frac{I_{yy}-I_{zz}}{I_{yy}}(\dot{\psi} - \dot{\theta})\dot{\phi}, \\ \ddot{\psi} &= \dot{\psi}\dot{\phi} + \dot{\phi}\dot{\theta} + \frac{l\dot{\psi}\phi}{I_{yy}}U_2 + \frac{l\phi}{I_{yy}}U_3 + \frac{1}{I_{zz}}U_4 - \frac{I_{yy}-I_{zz}}{I_{yy}}(\dot{\psi} - \dot{\theta})\dot{\phi}.\end{aligned}\quad (2.27)$$

2.5 SENSORES E ATUADORES DA AERONAVE

Este trabalho não abrange a implementação prática dos controladores propostos, atendo-se aos projetos e implementações computacionais. Para isso, considerou-se a seguinte entrada de controle:

$$\mathbf{u} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}. \quad (2.28)$$

Na prática, essa entrada deve ser convertida em tensões de alimentação dos motores da aeronave. Essa conversão é feita transformando U_i , $i = 1, 2, 3, 4$, em velocidades angulares de cada um dos motores, e essas velocidades em tensões por meio das equações que regem os motores.

Neste trabalho, o quadrotor de interesse é equipado com quatro sensores para obter as leituras das malhas a serem controladas (z , ϕ , θ e ψ) (ROCKENBACH, 2018):

1. **Sensor de ultrassom:** usado para estimar a altitude da aeronave. Possui um transmissor e um receptor de som de alta frequência. Quando é emitida uma onda de som, ela é refletida por uma superfície e medida pelo receptor. A distância entre o sensor e a superfície depende do tempo de ida e volta da onda;
2. **Acelerômetro:** tem a capacidade de medir acelerações estáticas, como a gravidade, e acelerações dinâmicas, causadas pela variação da velocidade do corpo. Esse sensor é capaz de mensurar componentes vetoriais paralelas ao campo gravitacional da Terra, logo ele é incapaz de medir rotações paralelas ao plano terrestre. Portanto, o acelerômetro pode ser utilizado para estimar os ângulos de arfagem e rolamento, mas não é capaz de estimar o ângulo de guinada, pois este representa uma rotação em relação ao eixo paralelo ao plano terrestre;
3. **Giroscópio:** baseia-se na lei da inércia, princípio físico formulado por Newton, na qual um corpo em movimento mantém-se em movimento até que uma força externa seja aplicada sobre ele. Mais especificamente, os giroscópios detectam a aceleração de Coriolis nas peculiaridades dos movimentos oscilatórios. A partir disso, é possível obter a velocidade angular nos três eixos de orientação;
4. **Magnetômetro:** dispositivo eletrônico com função semelhante à bússola tradicional. A finalidade é quantificar a posição em relação ao polo norte magnético e conseqüentemente obter o ângulo de guinada.

SISTEMAS NEBULOSOS

Resumo do capítulo

O advento dos conjuntos nebulosos foi causado pela necessidade de um método capaz de expressar de uma maneira sistemática quantidades imprecisas, vagas, mal definidas. Para expressar conhecimentos nebulosos é muito comum o uso de elementos qualitativos ao invés de valores quantitativos. Elementos típicos incluem "quase todo", "mais ou menos", "alto", "não muitos", "médio", etc. Para se atribuir um significado aos termos linguísticos, associa-se cada um desses a um conjunto nebuloso definido sobre um universo de discurso comum. Na criação das regras nebulosas, o processo de inferência deve estar baseado nos operadores lógicos se-então. O modelo Takagi-Sugeno (TS), descrito por esse tipo de operadores lógicos, consiste em um sistema de inferência capaz de descrever, de forma exata ou aproximada, sistemas dinâmicos não lineares por meio de um conjunto de sistemas dinâmicos lineares, localmente válidos, interpolados de forma suave, não linear e convexa.

Uma extensão da lógica binária, desenvolvida por Boole (lógica booleana) e que utiliza apenas dois valores 0 ou 1, é a lógica multivalorada, pela qual uma variável pode assumir vários valores, por exemplo, falso e verdadeiro, ao mesmo tempo. A lógica nebulosa é uma lógica multivalorada capaz de capturar informações vagas, geralmente descritas numa linguagem natural, e convertê-las em um formato numérico de fácil manipulação (SAMPAIO; OLIVEIRA; IGNACIO, 2007).

3.1 CONJUNTOS NEBULOSOS

Seja X um determinado universo de discurso, no qual representa-se um elemento genérico por x , e um determinado conjunto designado por A . Na teoria clássica de conjuntos, um dado elemento tem carácter binário: o elemento pertence ($x \in A$) ou não pertence ($x \notin A$) ao conjunto.

Através de uma função característica $f_A(x)$ é possível definir essa propriedade fazendo o mapeamento dos elementos do universo de discurso no conjunto binário 0, 1:

$$f_A(x) : X \rightarrow \{0, 1\}, \quad f_A(x) = \begin{cases} 1 & \text{se e somente se } x \in A \\ 0 & \text{se e somente se } x \notin A \end{cases} \quad (3.1)$$

O advento dos conjuntos nebulosos foi causado pela necessidade de um método capaz de expressar de uma maneira sistemática quantidades imprecisas, vagas, mal definidas (SIMÕES; SHAW, 2007). Claramente, a "classe de todos os reais muito maiores que 1", ou "a classe de mulheres bonitas" ou "a classe de homens altos" não constituem classes ou conjuntos no senso

matemático comum (ZADEH, 1965), portanto, não podem ser representados por conjuntos clássicos.

Um conceito fundamental na teoria dos conjuntos nebulosos é o grau de pertinência. O grau de pertinência de um determinado objeto a um conjunto nebuloso é um número real definido no intervalo $[0, 1]$ que representa quão verdadeira é a afirmação de que um certo elemento pertence a um dado conjunto. Um conjunto nebuloso "A" pode ser caracterizado por sua função característica, também denominada função de pertinência e denotada por $\mu_A(x)$, assim definida:

$$\mu_A(x) = X \rightarrow [0, 1], \quad (3.2)$$

em que X é denominado *universo do discurso*, definido como o espaço ou domínio das variáveis deste conjunto, e $x \in U$ (NUNEZ, 2001).

Assim como nos demais conjuntos, podemos fazer operações com conjuntos nebulosos:

- Interseção (e): $\mu_{A \cup B} = \text{prod}(\mu_A, \mu_B)$;
- União (ou): $\mu_{A \cap B} = \max(\mu_A, \mu_B)$;
- Complemento (não): $\mu_{\bar{A}} = 1 - \mu_A$.

Essas operações podem ser feitas de modos diferentes (MIZUMOTO; TANAKA, 1981). Por exemplo, a interseção pode ser calculada usando o operador de mínimo, $\mu_{A \cup B} = \min(\mu_A, \mu_B)$, em vez do operador de produto.

3.2 LÓGICA NEBULOSA

A diferença entre a lógica tradicional e a lógica nebulosa pode ser melhor entendida com dois exemplos enunciados abaixo. Eles são encontrados no trabalho de (MOZELLI, 2008).

Na lógica tradicional, a forma básica de se chegar a uma conclusão é através do silogismo. Em um silogismo, existem duas premissas a partir das quais se pode deduzir uma conclusão. Tradicionalmente, a lógica segue o *modus ponendo ponens*, também chamado de *modus ponens*, no qual o fato de se constatar a veracidade da premissa A implica na veracidade de B , pois há uma outra premissa que estabelece que A implica em B ($A \rightarrow B$).

Por exemplo, considerando as duas premissas abaixo:

$$\begin{aligned} A &: \text{"a temperatura da água é } 100^\circ\text{C"} \text{ e} \\ B &: \text{"a água está fervendo"}, \end{aligned}$$

caso a premissa A seja verdadeira, também será verdadeira a premissa B . O conceito segue a estrutura:

- premissa (regra): se x é A , então y é B ($A \rightarrow B$);

- premissa (fato): x é A ;
- conclusão: y é B .

Entretanto, em diversas situações práticas a lógica é adotada de forma aproximada. Considere a implicação "se o tomate está vermelho, então ele está maduro". Se na premissa é estabelecido que "o tomate está quase todo vermelho", então é possível inferir que "o tomate está quase maduro". Contudo, a estrutura lógica tradicional não permite essa conclusão.

Essa forma de raciocínio aproximado é sumarizada da forma a seguir:

- premissa (regra): se x é A , então y é B ($A \rightarrow B$);
- premissa (fato): x é A' ;
- conclusão: y é B' ,

sendo que A' e B' são próximos a A e B , respectivamente. Logo, a regra acima associa uma função de pertinência que mede o grau de veracidade (similaridade) da implicação. Esse procedimento de dedução lógica é conhecido como raciocínio nebuloso, raciocínio fuzzy, raciocínio aproximado ou *modus ponens* generalizado.

Para expressar conhecimentos nebulosos, assim como feito no exemplo acima, é muito comum o uso de elementos qualitativos ao invés de valores quantitativos. Elementos típicos incluem "quase todo", "mais ou menos", "alto", "não muitos", "médio", etc. Estas ideias são capturadas pela definição de variável linguística, que tem por característica assumir valores dentro de um conjunto de termos linguísticos, ou seja, palavras ou frases. Assim, ao invés de assumir instâncias numéricas, uma variável linguística assume instâncias linguísticas. Por exemplo, uma variável linguística *Temperatura* poderá assumir como valor um dos membros do conjunto $\{baixa, média, alta\}$. Para se atribuir um significado aos termos linguísticos, associa-se cada um desses a um conjunto nebuloso definido sobre um universo de discurso comum (GOMIDE; GUDWIN, 1994).

Na criação das regras nebulosas o processo de inferência, que gera as conclusões, deve estar baseado nos operadores lógicos *se-então*. A preposição *se* refere-se aos termos *antecedentes* e a proposição *então*, aos *consequentes*.

3.3 MODELO TAKAGI-SUGENO

O modelo Takagi-Sugeno (TS) consiste em um sistema de inferência capaz de descrever, de forma exata ou aproximada, sistemas dinâmicos não lineares por meio de um conjunto de sistemas dinâmicos lineares, localmente válidos, interpolados de forma suave, não linear e convexa. Em termos de modelagem, sabe-se que os modelos TS têm a capacidade de aproximar uniformemente funções quaisquer em um domínio compacto. Em outras palavras, modelos TS têm a capacidade de *aproximação universal* (MOZELLI, 2008).

Podemos representar um sistema dinâmico não linear por um modelo TS com as seguintes regras do tipo *se-então* (CAMPOS, 2015):

$$R_i : \text{Se } z_1 \in M_1^{k_1=\alpha_{i1}} \text{ e } z_2 \in M_2^{k_2=\alpha_{i2}} \text{ e } \dots \text{ e } z_n \in M_n^{k_n=\alpha_{in}},$$

$$\text{então } \begin{cases} \dot{\mathbf{x}} = \mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_{u_i} \mathbf{u}(t) + \mathbf{B}_{d_i} \mathbf{d}(t) \\ \mathbf{y} = \mathbf{C}_i \mathbf{x}(t) + \mathbf{D}_{u_i} \mathbf{u}(t) + \mathbf{D}_{d_i} \mathbf{d}(t) \end{cases}, \quad (3.3)$$

em que $\mathbf{x}(t) \in \mathbb{R}^m$ é o vetor de estados; $\mathbf{u}(t) \in \mathbb{R}^k$ é o vetor de entradas de controle; $\mathbf{d}(t) \in \mathbb{R}^z$ é o vetor de entradas de perturbação; $\mathbf{y} \in \mathbb{R}^l$ é o vetor de saídas controladas; as matrizes \mathbf{A}_i , \mathbf{B}_{u_i} , \mathbf{B}_{d_i} , \mathbf{C}_i , \mathbf{D}_{u_i} e \mathbf{D}_{d_i} são as matrizes que descrevem o comportamento do sistema e são de dimensões adequadas; R_i indica o número da regra; z_1 a z_n são as variáveis premissas do modelo; $M_j^{k_j}$ são os conjuntos nebulosos relacionados à variável z_j , e α_{ij} relaciona qual a conjunto fuzzy da variável z_j está associado à regra i (por exemplo, $\alpha_{21} = 3$ indica que o terceiro conjunto nebuloso da variável premissa z_1 é usado na segunda regra). Apesar de apenas as variáveis de estados estarem sendo utilizadas como variáveis premissas, tal modelo modelo permite a modelagem de qualquer sistema não linear afim nas entradas de controle e perturbação.

Considerando que cada conjunto $M_j^{\alpha_{ij}}$ corresponda a uma função de pertinência $w_j^{\alpha_{ij}}(z_j)$ com

$$0 \leq w_j^{\alpha_{ij}}(z_j) \leq 1,$$

a função de pertinência normalizada pode ser obtida de $w_j^{\alpha_{ij}}(z_j)$ por

$$\mu_j^{\alpha_{ij}}(z_j) = \frac{w_j^{\alpha_{ij}}(z_j)}{\sum_{\alpha_{ij}=1}^{r_j} w_j^{\alpha_{ij}}(z_j)},$$

com r_j sendo o número de conjuntos representados por M_j . Tais funções de pertinência possuem as seguintes propriedades:

$$\begin{aligned} \mu_j^{\alpha_{ij}}(z_j) &\geq 0; \\ \sum_{\alpha_{ij}=1}^{r_j} \mu_j^{\alpha_{ij}}(z_j) &= 1. \end{aligned} \quad (3.4)$$

As funções de pertinência normalizadas do modelo são obtidas usando o operador de interseção para conjuntos nebulosos:

$$h_i(\mathbf{z}) = \prod_{j=1}^n \mu_j^{\alpha_{ij}}(z_j) = \frac{\prod_{j=1}^n w_j^{\alpha_{ij}}(z_j)}{\sum_{k=1}^r \prod_{j=1}^n w_j^{\alpha_{kj}}(z_j)}. \quad (3.5)$$

Tais funções de pertinência normalizadas têm as seguintes propriedades:

$$\begin{aligned} h_i(\mathbf{z}) &\geq 0; \\ \sum_{i=1}^r h_i(\mathbf{z}) &= 1, \end{aligned}$$

em que r é o número de regras do modelo. O modelo inferido pode ser reescrito como

$$\begin{aligned}\dot{\mathbf{x}} &= \sum_{i=1}^r h_i(\mathbf{z}) [A_i \mathbf{x} + \mathbf{B}_{\mathbf{u}_i} \mathbf{u} + \mathbf{B}_{\mathbf{d}_i} \mathbf{d}], \\ \mathbf{y} &= \sum_{i=1}^r h_i(\mathbf{z}) [C_i \mathbf{x} + \mathbf{D}_{\mathbf{u}_i} \mathbf{u} + \mathbf{D}_{\mathbf{d}_i} \mathbf{d}].\end{aligned}\tag{3.6}$$

TRANSFORMAÇÃO DO PRODUTO TENSORIAL

Resumo do capítulo

Uma desvantagem do controle nebuloso é que não existe uma estrutura padronizada em relação ao projeto, otimização, redutibilidade e particionamento de um conjunto de regras nebulosas. A transformação do produto tensorial é uma técnica numérica que, dada uma representação do sistema na forma quasi-Linear a Parâmetros Variantes (qLPV ou quasi-LPV), em inglês quasi Linear Parameter Varying, encontra uma representação convexa equivalente. Tem aplicações interessantes na obtenção numérica de um modelo TS de um sistema, bem como na redução do número de regras de um determinado modelo. A base dessa técnica é a decomposição em valores singulares de alta ordem (HOSVD), ferramenta matemática que pode ser vista como a generalização da decomposição em valores singulares, que, por sua vez, é uma ferramenta matemática que pode ser usada para identificação de padrões de dados organizados de forma matricial.

Uma desvantagem do controle nebuloso é que não existe uma estrutura padronizada em relação ao projeto, otimização, redutibilidade e particionamento de um conjunto de regras nebulosas. Uma base de regras nebulosas pode conter componentes redundantes, de contribuição fraca ou totalmente inconsistentes. Além disso, em busca de uma boa aproximação, pode-se ficar tentado a atribuir excessivamente o número de conjuntos antecedentes, resultando em grandes bases de regras nebulosas e problemas em tempo de computação e espaço de armazenamento. Uma abordagem formal para extrair os elementos mais pertinentes de um determinado conjunto de regras é, portanto, altamente desejável (YAM; BARANYI; YANG, 1999).

A transformação do produto tensorial é uma técnica numérica que, dada uma representação do sistema na forma quasi-Linear Variante nos Parâmetros (qLPV ou quasi-LPV), em inglês *quasi Linear Parameter Varying*, encontra uma representação convexa equivalente. Tem aplicações interessantes na obtenção numérica de um modelo TS de um sistema, bem como a redução do número de regras de um determinado modelo (CAMPOS, 2015). Os procedimentos da técnica envolvem a discretização do modelo qLPV dado e, em seguida, usa-se a decomposição em valores singulares de alta ordem (HOSVD), em inglês *Higher Order Singular Value Decomposition*, para obter a estrutura em forma de produto tensorial (TP) única do modelo fornecido (BARANYI; YAM; VÁRLAKI, 2014). O ponto-chave da transformação do produto tensorial é o uso da HOSVD para encontrar, entre todos os valores amostrados, aqueles que realmente acrescentam informação ao modelo (CAMPOS, 2015).

A seguir serão apresentados os conceitos matemáticos necessários para o entendimento e desenvolvimento da técnica. Ao final do capítulo, é mostrado um exemplo que aborda o seu uso.

4.1 DECOMPOSIÇÃO EM VALORES SINGULARES DE TENSORES DE ALTA ORDEM

A decomposição em valores singulares (SVD), em inglês *singular value decomposition*, é um algoritmo numérico desenvolvido para minimizar erros computacionais envolvendo grandes operações matriciais (MOORE, 1986).

Teorema 4.1 (decomposição em valores singulares - SVD). (STEWART, 1993) *Toda matriz $\mathbf{A} \in \mathbb{R}^{I_1 \times I_2}$ pode ser reescrita como o produto:*

$$\mathbf{A} = \mathbf{U}_1 \cdot \mathbf{S} \cdot \mathbf{U}_2^T \quad (4.1)$$

Em que:

A matriz $\mathbf{U}_1 = [\mathbf{u}_{1,1} \mathbf{u}_{1,2} \dots \mathbf{u}_{1,I_1}] \in \mathbb{R}^{I_1 \times I_1}$ é uma matriz ortonormal cujas colunas, $\mathbf{u}_{1,i}$, são chamadas de vetores singulares à esquerda.

A matriz $\mathbf{U}_2 = [\mathbf{u}_{2,1} \mathbf{u}_{2,2} \dots \mathbf{u}_{2,I_2}] \in \mathbb{R}^{I_2 \times I_2}$ é uma matriz ortonormal cujas linhas, $\mathbf{u}_{2,i}$, são chamadas de vetores singulares à direita.

$\mathbf{S} \in \mathbb{R}^{I_1 \times I_2}$ é chamada de matriz de valores singulares. Ela possui as seguintes propriedades:

(a) pseudo-diagonalidade:

$$\mathbf{S} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_{\min(I_1, I_2)});$$

(b) ordenamento: os elementos que formam a matriz \mathbf{S} são chamados de valores singulares e são ordenados de tal forma que:

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(I_1, I_2)} \geq 0.$$

O número de valores singulares σ_i não nulos é igual ao posto de \mathbf{A} .

Pode-se escrever (4.1) em forma de somatória:

$$\mathbf{A} = \sum_{i=1}^r \sigma_i \mathbf{u}_{1i} \mathbf{u}_{2,i}, \quad (4.2)$$

em que, $r = \min(I_1, I_2)$, \mathbf{u}_{1i} são vetores coluna e $\mathbf{u}_{2,i}$ são vetores linha. A SVD pode ser interpretada como uma ferramenta de identificação de padrões de dados organizados em forma matricial, em que a matriz \mathbf{S} armazena a importância de cada padrão de maneira decrescente. O exemplo a seguir ilustra essa interpretação.

Exemplo 1: Reconhecimento de grupos em relação ao gosto por gêneros de filmes

A Figura 3 mostra uma tabela cujos elementos representam notas dadas por pessoas em relação ao quanto gostaram de um determinado filme. A nota zero indica que a pessoa não assistiu àquele filme, e as notas no intervalo $[1, 5]$ indicam, em ordem crescente, o quanto a pessoa gostou do filme.

Claramente, identificamos dois grupos de pessoas: aquelas que gostam de filmes de ficção científica (Matriz, Alien e Interstellar) e aquelas que gostam de filmes de romance (Casablanca e Titanic), representadas, respectivamente, pelas cores azul e verde (Figura 3).

Os dados da tabela podem ser representados matricialmente, conforme visto abaixo. Usando a SVD para encontrar padrões nesses dados, notamos que existem três padrões, sendo dois deles mais importantes (têm valores singulares muito maiores). O primeiro padrão refere-se ao grupo de pessoas que gostam de ficção científica; o segundo, pessoas que gostam de filmes de romance.

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix} = \begin{bmatrix} 0,13 & 0,02 & -0,01 \\ 0,41 & 0,07 & -0,03 \\ 0,55 & 0,09 & -0,04 \\ 0,68 & 0,11 & -0,05 \\ 0,15 & -0,59 & 0,65 \\ 0,07 & -0,73 & -0,67 \\ 0,07 & -0,29 & 0,32 \end{bmatrix} \times \begin{bmatrix} 12,4 & 0 & 0 \\ 0 & 9,5 & 0 \\ 0 & 0 & 1,3 \end{bmatrix} \times$$

$$\begin{bmatrix} 0,56 & 0,59 & 0,56 & 0,09 & 0,09 \\ 0,12 & -0,02 & 0,12 & -0,69 & -0,69 \\ 0,40 & -0,80 & 0,40 & 0,09 & 0,09 \end{bmatrix} = \begin{bmatrix} \mathbf{u}_{11} & \mathbf{u}_{12} & \mathbf{u}_{13} \end{bmatrix} \mathbf{S} \begin{bmatrix} \mathbf{u}_{2,1} \\ \mathbf{u}_{2,2} \\ \mathbf{u}_{2,3} \end{bmatrix}.$$

Nota-se que as matrizes $\mathbf{U}_1 \notin \mathcal{R}^{7 \times 7}$, $\mathbf{U}_2 \notin \mathcal{R}^{5 \times 5}$ e $\mathbf{S} \notin \mathcal{R}^{7 \times 5}$. Isso porque os valores singulares iguais a zero, e as respectivas colunas em \mathbf{U}_1 e linhas em \mathbf{U}_2 , foram descartados, resultando na chamada de *decomposição em valores singulares reduzida*. Outra observação é que a SVD não é capaz de relacionar os padrões encontrados com suas respectivas interpretações, ou seja, a interpretação acima sobre existir um grupo de pessoas que gosta de filmes de ficção científica e um que gosta de romance foi feita com base em um entendimento prévio dos dados (o entendimento que existiam filmes de ficção científica e de romance).

Escrito de outra forma, vemos que a matriz de dados pode ser reescrita como uma soma ponderada de matrizes, obtidas pela multiplicação dos vetores coluna pelos vetores linha,

feita pelos valores singulares.

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix} = 12,4 \begin{bmatrix} 0,0728 & 0,0767 & 0,0728 & 0,0117 & 0,0117 \\ 0,2296 & 0,2419 & 0,2296 & 0,0369 & 0,0369 \\ 0,3080 & 0,3245 & 0,3080 & 0,0495 & 0,0495 \\ 0,3808 & 0,4012 & 0,3808 & 0,0612 & 0,0612 \\ 0,0840 & 0,0885 & 0,0840 & 0,0135 & 0,0135 \\ 0,0392 & 0,0413 & 0,0392 & 0,0063 & 0,0063 \\ 0,0392 & 0,0413 & 0,0392 & 0,0063 & 0,0063 \end{bmatrix} +$$

$$9,5 \begin{bmatrix} 0,0024 & -0,0004 & 0,0024 & -0,0138 & -0,0138 \\ 0,0084 & -0,0014 & 0,0084 & -0,0483 & -0,0483 \\ 0,0108 & -0,0018 & 0,0108 & -0,0621 & -0,0621 \\ 0,0132 & -0,0022 & 0,0132 & -0,0759 & -0,0759 \\ -0,0708 & 0,0118 & -0,0708 & 0,4071 & 0,4071 \\ -0,0876 & 0,0146 & -0,0876 & 0,5037 & 0,5037 \\ -0,0348 & 0,0058 & -0,0348 & 0,2001 & 0,2001 \end{bmatrix} +$$

$$1,3 \begin{bmatrix} -0,0040 & 0,0080 & -0,0040 & -0,0009 & -0,0009 \\ -0,0120 & 0,0240 & -0,0120 & -0,0027 & -0,0027 \\ -0,0160 & 0,0320 & -0,0160 & -0,0036 & -0,0036 \\ -0,0200 & 0,0400 & -0,0200 & -0,0045 & -0,0045 \\ 0,2600 & -0,5200 & 0,2600 & 0,0585 & 0,0585 \\ -0,2680 & 0,5360 & -0,2680 & -0,0603 & -0,0603 \\ 0,1280 & -0,2560 & 0,1280 & 0,0288 & 0,0288 \end{bmatrix} .$$

Essa forma de escrita evidencia o porquê de, em diversas aplicações, a SVD reduzida ser usada, uma vez que as matrizes ponderadas por valores singulares nulos em nada agregam para reconstruir a matriz original.

	Matriz	Alien	Interstellar	Casablanca	Titanic
Lucas	1	1	1	0	0
Víctor	3	3	3	0	0
Renan	4	4	4	0	0
Guilherme	5	5	5	0	0
Luciano	0	2	0	4	4
Ariel	0	0	0	5	5
Eloisa	0	1	0	2	2

Figura 3 – Dados sobre opiniões de pessoas em relação aos filmes.

Um número crescente de problemas de processamento de sinal envolve a manipulação de quantidades cujos elementos são endereçados por mais de dois índices. Na literatura, esses

equivalentes de ordem superior de vetores (primeira ordem) e matrizes (segunda ordem) são chamadas de tensores de ordem superior, matrizes multidimensionais ou matrizes de múltiplos caminhos (LATHAUWER; MOOR; VANDEWALLE, 2000). Nesse cenário, usa-se a decomposição em valores singulares de alta ordem (HOSVD), técnica que pode ser interpretada como a generalização da SVD.

4.1.1 CONCEITOS PRELIMINARES

A forma com que os conceitos são apresentados é baseada nos trabalhos de (CAMPOS, 2015), (PETRES, 2006) e (BARANYI; YAM; VÁRLAKI, 2014).

O ponto inicial para a generalização da SVD para tensores de alta ordem é encontrar uma representação matricial para eles, de tal forma que operações tensoriais se transformem em operações matriciais. Essas matrizes são chamadas de *matrizes modo- n* ou *matrizes de desdobramento*:

Definição 4.1 (Matriz modo- n de um tensor \mathcal{A}). Considere um tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ de ordem N . A *matriz modo- n* , $\mathbf{A}_{(n)} \in \mathbb{R}^{I_N \times J}$, com $J = \prod_{k=1, k \neq n}^N I_k$, é uma representação matricial possível para o tensor. Tal matriz contém o elemento $a_{i_1 i_2 \dots i_N}$ do tensor na linha i_N e na coluna

$$(i_{n+1} - 1)I_{n+2}I_{n+3} \dots I_N I_1 I_2 \dots I_{n-1} + (i_{n+2} - 1)I_{n+3}I_{n+4} \dots I_N I_1 I_2 \dots I_{n-1} + \dots + (i_{N-1})I_1 I_2 \dots I_{n-1} + (i_1 - 1)I_2 I_3 \dots I_{n-1} + (i_2 - 1)I_3 I_4 \dots I_{n-1} + \dots + i_{n-1}.$$

Em outras palavras, essa definição mostra que o desdobramento de *modo- n* será tal que:

- O n -ésimo índice de cada elemento indica em qual linha ele estará na *matriz de modo- n* .
- As colunas são "desdobradas" na seguinte ordem: primeiro o $(n - 1)$ -ésimo elemento, seguido do $(n - 2)$ -ésimo elemento e assim por diante de forma circular.

A Figura 4 mostra a representação das matrizes de modo-1, modo-2 e modo-3 de um tensor de 3ª ordem.

É necessário que a *matriz de valores singulares* da HOSVD possua as mesmas propriedades apresentadas na SVD. Essa restrição requer o uso do conceito de *norma de Frobenius* de um tensor, que, por sua vez, requer o de *produto escalar* e *ortogonalidade* de tensores.

Definição 4.2 (Produto escalar entre tensores). O produto escalar $\langle \mathcal{A}, \mathcal{B} \rangle$ de dois tensores $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_n}$ é definido como:

$$\langle \mathcal{A}, \mathcal{B} \rangle \stackrel{\text{def}}{=} \sum_{i_1} \sum_{i_2} \dots \sum_{i_N} b_{i_1 i_2 \dots i_N} a_{i_1 i_2 \dots i_N} \quad (4.3)$$

Definição 4.3 (Ortogonalidade). Dois tensores são ditos ortogonais se seu produto escalar é igual a zero.

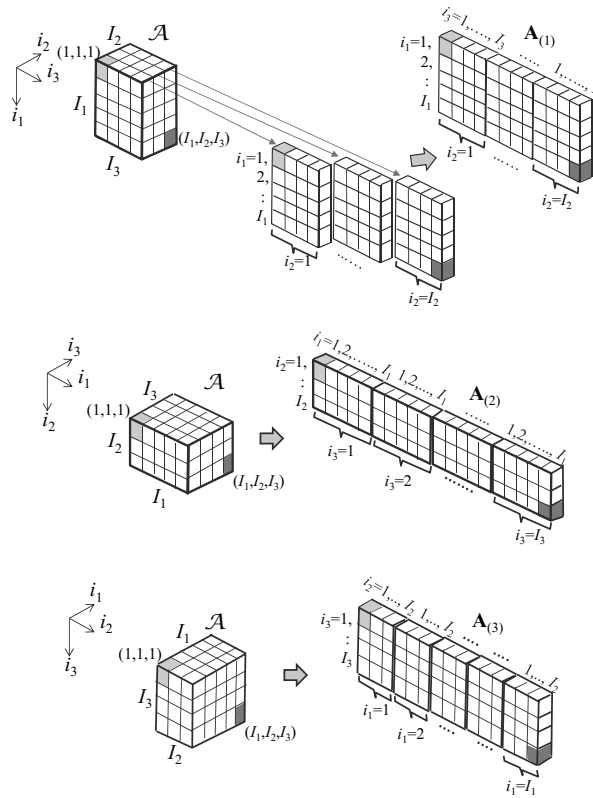


Figura 4 – Matrizes modo- n de um tensor de 3ª ordem.

Fonte : (BARANYI; YAM; VÁRLAKI, 2014)

Definição 4.4 (Norma de Frobenius). A norma de Frobenius de um tensor \mathcal{A} é dada por:

$$\|\mathcal{A}\| \stackrel{\text{def}}{=} \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle} \quad (4.4)$$

Por fim, vamos observar o que acontece no produto matricial $\mathbf{G} = \mathbf{U} \cdot \mathbf{F} \cdot \mathbf{V}^T$, envolvendo as matrizes $\mathbf{F} \in \mathbb{R}^{I_1 \times I_2}$, $\mathbf{U} \in \mathbb{R}^{J_1 \times I_1}$, $\mathbf{V} \in \mathbb{R}^{J_2 \times I_2}$ e $\mathbf{G} \in \mathbb{R}^{J_1 \times J_2}$. Para evitar trabalhar com "transposições generalizadas" para o caso tensorial, observamos que a relação entre \mathbf{U} e \mathbf{F} e entre \mathbf{V} (não \mathbf{V}^T) e \mathbf{F} são, na verdade, complemente semelhantes: da mesma forma que \mathbf{U} faz combinações lineares com as linhas de \mathbf{F} , \mathbf{V} faz combinações lineares com as colunas de \mathbf{F} ; da mesma forma que as colunas de \mathbf{F} são multiplicadas por \mathbf{U} , as linhas de \mathbf{F} são multiplicadas por \mathbf{V} ; da mesma forma que as colunas de \mathbf{U} estão associadas à imagem das colunas de \mathbf{G} , as colunas de \mathbf{V} estão associadas à imagem das linhas de \mathbf{G} . Essa relação é tipicamente denotada por $\mathbf{G} = \mathbf{F} \times_1 \mathbf{U} \times_2 \mathbf{V}$ (LATHAUWER; MOOR; VANDEWALLE, 2000), definida por:

Definição 4.5 (Produto modo- n). O produto modo- n de um tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ por uma matriz $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$ ($\mathcal{A} \times_n \mathbf{U}$) é um tensor pertencente a $\mathbb{R}^{I_1 \times I_2 \times \dots \times I_{n-1} \times J_n \times I_{n+1} \times \dots \times I_N}$ cujos elementos são dados por

$$(\mathcal{A} \times_n \mathbf{U})_{i_1 i_2 \dots i_{n-1} j_n i_{n+1} \dots i_N} \stackrel{\text{def}}{=} \sum_{i_n} a_{i_1 i_2 \dots i_{n-1} j_n i_{n+1} \dots i_N} u_{j_n i_n}$$

De forma compacta, o produto modo- n entre $\mathcal{A} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \cdots \times_N \mathbf{U}_N$ pode ser escrito como

$$\mathcal{A} \underset{n=1}{\otimes}^N \mathbf{U}_n.$$

Do ponto de vista computacional, o produto modo- n de um tensor por uma matriz, $\mathcal{A} = \mathcal{B} \times_n \mathbf{U}$, pode ser obtido encontrando primeiro a matriz de modo- n de \mathcal{B} , $\mathbf{B}_{(n)}$, computando o produto $\mathbf{A}_{(n)} = \mathbf{U}\mathbf{B}_{(n)}$, e então convertendo $\mathbf{A}_{(n)}$ para recuperar \mathcal{A} .

Essa operação satisfaz as seguintes propriedades:

Propriedade 4.1. (LATHAUWER; MOOR; VANDEWALLE, 2000) Dado o tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ e as matrizes $\mathbf{F} \in \mathbb{R}^{J_n \times I_n}$, $\mathbf{G} \in \mathbb{R}^{J_m \times I_m}$, com $n \neq m$,

$$(\mathcal{A} \times_n \mathbf{F}) \times_m \mathbf{G} = (\mathcal{A} \times_m \mathbf{G}) \times_n \mathbf{F} = \mathcal{A} \times_n \mathbf{F} \times_m \mathbf{G}.$$

Propriedade 4.2. (LATHAUWER; MOOR; VANDEWALLE, 2000) Dado o tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ e as matrizes $\mathbf{F} \in \mathbb{R}^{J_n \times I_n}$, $\mathbf{G} \in \mathbb{R}^{K_n \times J_n}$, temos que

$$(\mathcal{A} \times_n \mathbf{F}) \times_n \mathbf{G} = \mathcal{A} \times_n (\mathbf{G} \cdot \mathbf{F}).$$

4.1.2 TÉCNICA DA DECOMPOSIÇÃO EM VALORES SINGULARES DE ALTA ORDEM

A partir dos conceitos descritos, pode-se definir a decomposição em valores singulares de alta ordem.

Teorema 4.2 (HOSVD). (LATHAUWER; MOOR; VANDEWALLE, 2000) *Todo tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ pode ser escrito como o produto*

$$\mathcal{A} = \mathcal{S} \times_1 \mathbf{U}_{(1)} \times_2 \mathbf{U}_{(2)} \times_3 \cdots \times_N \mathbf{U}_{(N)} = \mathcal{S} \underset{n=1}{\otimes}^N \mathbf{U}_{(n)},$$

em que

1. $\mathbf{U}_{(n)} = (\mathbf{u}_{n,1} \ \mathbf{u}_{n,2} \ \dots \ \mathbf{u}_{n,I_n})$, $1, \dots, N$, é uma matriz ortonormal de dimensões $I_n \times I_n$ chamada de matriz singular de modo- n ;
2. $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, chamado de tensor de núcleo, é um tensor cujos subtensores $\mathcal{S}_{i_n=\alpha}$, obtidos fixando-se o n -ésimo índice em α , têm as seguintes propriedades:

- a) *toda-ortogonalidade: dois subtensores $\mathcal{S}_{i_n=\alpha}$ e $\mathcal{S}_{i_n=\beta}$ são ortogonais para todos os valores possíveis de n , α e β sujeitos a $\alpha \neq \beta$:*

$$\langle \mathcal{S}_{i_n=\alpha}, \mathcal{S}_{i_n=\beta} \rangle = 0, \quad \forall \alpha \neq \beta;$$

- b) *ordenação:*

$$\|\mathcal{S}_{i_n=1}\| \geq \|\mathcal{S}_{i_n=2}\| \geq \dots \geq \|\mathcal{S}_{i_n=I_n}\| \geq 0, \quad \forall n.$$

As normas de Frobenius $\|\mathcal{S}_{i_n=1}\|$, muitas vezes simbolizadas por $\sigma_i^{(n)}$, são os valores singulares de modo- n de \mathcal{A} , os vetores $\mathbf{u}_{n,i}$ são os vetores singulares de modo- n e as matrizes $\mathbf{U}_{(n)}$ são as matrizes singulares de modo- n .

A Figura 5 mostra um exemplo da HOSVD de um tensor de 3ª ordem. O cômputo dessa decomposição é realizado em dois passos. Inicialmente são calculadas as matrizes singulares de modo- n e em seguida é calculado o tensor de núcleo.

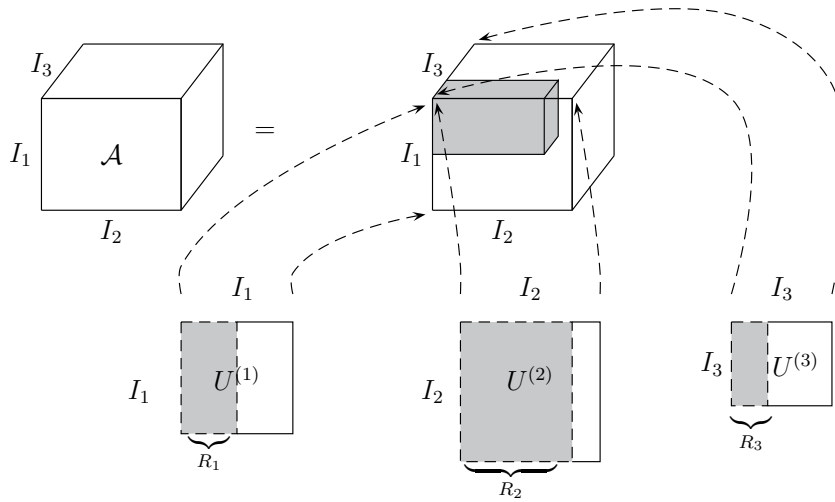


Figura 5 – Exemplo de HOSVD de um tensor de 3ª ordem.

Fonte : (BARANYI; YAM; VÁRLAKI, 2014)

As matrizes singulares de modo- n , $\mathbf{U}_{(n)}$, são as matrizes de vetores singulares esquerdos das matrizes de desdobramento, $\mathbf{A}_{(n)}$, do tensor \mathcal{A} . Sendo assim, cada matriz (e os respectivos valores singulares de modo- n) é encontrada a partir de uma decomposição de valores singulares (SVD) de cada matriz de desdobramento do tensor,

$$\mathbf{A}_{(n)} = \mathbf{U}_{(n)} \mathbf{\Theta}_{(n)} (\mathbf{V}_{(n)})^T.$$

Nessa etapa, as colunas das matrizes $\mathbf{U}_{(n)}$ correspondentes a valores singulares nulos podem ser descartadas.

Tendo as matrizes singulares, o tensor de núcleo pode ser encontrado fazendo-se

$$\mathcal{S} = \mathcal{A} \times_1 \mathbf{U}_{(1)}^T \times_2 \mathbf{U}_{(2)}^T \times_3 \dots \times_N \mathbf{U}_{(N)}^T = \mathcal{A} \underset{n=1}{\otimes}^N \mathbf{U}_{(n)}^T,$$

com o uso da Propriedade 4.2.

4.2 MODELOS TP

Considerando-se o seguinte modelo em espaço de estados:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}(\mathbf{p}(t))\mathbf{x}(t) + \mathbf{B}(\mathbf{p}(t))\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}(\mathbf{p}(t))\mathbf{x}(t) + \mathbf{D}(\mathbf{p}(t))\mathbf{u}(t)\end{aligned}\quad (4.5)$$

com entrada $\mathbf{u}(t) \in \mathbb{R}^k$, $\mathbf{y}(t) \in \mathbb{R}^l$ e o vetor de estados $\mathbf{x}(t) \in \mathbb{R}^m$. O sistema de matrizes

$$\mathbf{S}(\mathbf{p}(t)) = \begin{bmatrix} \mathbf{A}(\mathbf{p}(t)) & \mathbf{B}(\mathbf{p}(t)) \\ \mathbf{C}(\mathbf{p}(t)) & \mathbf{D}(\mathbf{p}(t)) \end{bmatrix} \in \mathbb{R}^{(m+k) \times (m+l)} \quad (4.6)$$

é um objeto de parâmetro variável, em que $\mathbf{p}(t) \in \Omega$ é um vetor de parâmetros N -dimensional variante no tempo limitado pelo hipercubo $\Omega = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_N, b_N] \subset \mathbb{R}^N$. Se o parâmetro $\mathbf{p}(t)$ não incluir nenhum elemento de $\mathbf{x}(t)$, (4.5) é um modelo Linear a Parâmetros Variantes (LPV). Se o parâmetro $\mathbf{p}(t)$ incluir algum elemento de $\mathbf{x}(t)$, (4.5) não é um modelo linear e sua denominação passa a ser modelo quasi-Linear a Parâmetros Variantes (qLPV).

A transformação do produto tensorial é uma técnica numérica, detalhada na próxima seção, que fornecerá aproximações para o sistema $\mathbf{S}(\mathbf{p})$ válidas dentro do subespaço limitado Ω . Tais aproximações convexas podem ser representadas através de um politopo de valores (não necessariamente pertencentes ao espaço imagem de $\mathbf{S}(\mathbf{p})$) e são capazes de representar o comportamento do modelo, cuja dinâmica está contida no sistema matricial $\mathbf{S}(\mathbf{p})$. Tal representação é equivalente a um modelo nebuloso Takagi-Sugeno. Sendo assim, é possível escrever (4.5) como um *modelo politópico*. Para isso, omitindo, por conveniência, as dependências temporais, considere o caso em que o sistema de matrizes (4.6) é expresso por

$$\mathbf{S}(\mathbf{p}) = \sum_{h=1}^H w_h(\mathbf{p})\mathbf{S}_h \quad (4.7)$$

para todo vetor parâmetro \mathbf{p} , isto é, $\mathbf{S}(\mathbf{p})$ pode ser dado como uma combinação linear dependente de parâmetro de um conjunto de sistemas de matrizes lineares invariantes no tempo (LIT), ou sistemas de vértice, $\mathbf{S}_h \in \mathbb{R}^{(m+k) \times (m+l)}$, ponderado por funções contínuas multivariáveis, $w_h(\mathbf{p})$, denominadas funções de peso ou funções de ponderação. O sistema (4.5) torna-se então

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \mathbf{y} \end{bmatrix} = \left[\sum_{h=1}^H w_h(\mathbf{p})\mathbf{S}_h \right] \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} \quad (4.8)$$

Impondo que as funções de ponderação multivariáveis $w_h(\mathbf{p})$ em (4.7) e (4.8) satisfaçam as seguintes condições:

$$w_h(\mathbf{p}) \in [0, 1] \quad (4.9)$$

e

$$\sum_{h=1}^H w_h(\mathbf{p}) = 1, \quad (4.10)$$

a combinação linear em (4.7) torna-se convexa. Nesse caso, dizemos que (4.8) é uma *representação politópica finita* do sistema (4.5).

Definição 4.6 (Modelo politópico de elementos finitos). O sistema (4.8), com $w_h(\mathbf{p})$ satisfazendo (4.9) e (4.10), é um modelo politópico de elementos finitos de (4.5).

Elementos finitos significa que o número H de sistemas de matrizes envolvidas é limitado ($H < \infty$). O modelo é politópico, visto que o sistema de matrizes $\mathbf{S}(\mathbf{p})$ encontra-se dentro de um politopo definido no espaço de parâmetros Ω , com os sistemas lineares invariantes no tempo (LIT), \mathbf{S}_h , constituindo os vértices do politopo.

O modelo TP (produto tensorial), ou modelos politópicos tipo TP de representação finita, é uma classe de modelos na forma (4.8) com a propriedade adicional de que as funções de ponderação multivariáveis $w_h(\mathbf{p})$ são decompostas como produtos de funções de ponderação de variáveis únicas p_n , $n = 1, \dots, N$. Esta propriedade leva à incorporação de uma estrutura de produto tensorial na representação do sistema resultante.

Definição 4.7 (Modelo TP). Um modelo politópico de elementos finitos na forma (4.8), com $w_h(\mathbf{p}) = w_{1,i_1}(p_1), w_{2,i_2}(p_2) \dots w_{N,i_N}(p_N)$, para todo parâmetro \mathbf{p} , é um modelo TP do sistema (4.5).

O sistema de matrizes $\mathbf{S}(\mathbf{p})$ correspondente a (4.7) torna-se

$$\mathbf{S}(\mathbf{p}) = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} \prod_{n=1}^N w_{n,i_n}(p_n) \mathbf{S}_{i_1,i_2,\dots,i_N}, \quad (4.11)$$

para qualquer parâmetro \mathbf{p} .

Na notação tensorial:

$$\mathbf{S}(\mathbf{p}) = \mathcal{S} \underset{n=1}{\otimes}^N \mathbf{w}_n(p_n), \quad (4.12)$$

em que o tensor $(N + 2)$ -dimensional $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N \times (m+k) \times (m+l)}$ é construído a partir de sistemas de vértices LIT, $\mathbf{S}_{i_1,i_2,\dots,i_N} \in \mathbb{R}^{(m+k) \times (m+l)}$, e o vetor linha $\mathbf{w}_n(p_n) \in \mathbb{R}^{1 \times I_n}$ contém como elementos as funções de ponderação contínuas $w_{n,i_n}(p_n) \in [0, 1]$, para $i_n = 1, \dots, I_n$ e $n = 1, \dots, N$. A função $w_{n,i_n}(p_n)$ é a i_n -ésima função de ponderação definida na enésima dimensão de Ω , e p_n é o enésimo elemento do vetor \mathbf{p} . O modelo TP é então:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \mathbf{y} \end{bmatrix} = \left(\mathcal{S} \underset{n=1}{\otimes}^N \mathbf{w}_n(p_n) \right) \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}. \quad (4.13)$$

Esse modelo exibe muito mais componentes em sua estrutura que o modelo politópico de elementos finitos. Observe que o sistema (4.13) permanece politópico, ou seja, as funções de ponderação $w_h(\mathbf{p}) = w_{1,i_1}(p_1)w_{2,i_2}(p_2) \dots w_{N,i_N}(p_N)$ ainda satisfazem as condições de (4.9) e (4.10) como nos modelos politópicos de elemento finito.

Para tornar a estrutura do modelo TP equivalente a estrutura de um modelo TS, desejamos uma caracterização mais precisa das condições politópicas (4.9) e (4.10) em termos de $w_{n,i_n}(p_n)$, $n = 1, \dots, N$ e $i = 1, \dots, I_n$, em vez de seu produto $w_h(\mathbf{p})$. Apresentamos dois tipos de modelos TP a seguir.

Definição 4.8 (Modelo do tipo TP de não negatividade (NN)). O modelo TP (4.13) é do tipo NN se as funções de ponderação satisfazem, para $\forall \mathbf{p} \in \Omega$,

$$\forall n, i, p_n : w_{n,i}(p_n) \geq 0.$$

Definição 4.9 (Modelo do tipo TP de soma normalizada (SN)). O modelo TP (4.13) é do tipo SN se as funções de ponderação satisfazem, para $\forall \mathbf{p} \in \Omega$,

$$\forall n, p_n : \sum_{i_n=1}^{I_n} w_{n,i_n}(p_n) = 1.$$

Juntas, as condições NN e SN implicam que as funções de ponderação satisfazem

$$\forall n, i, p_n : w_{n,i}(p_n) \in [0, 1], \quad (4.14)$$

$$\forall n, p_n : \sum_{i=1}^{I_n} w_{n,i}(p_n) = 1. \quad (4.15)$$

Notamos que a condição (4.14) implica diretamente em (4.9) e que (4.15) implica (4.10), já que:

$$\sum_{h=1}^H w_h(\mathbf{p}) = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} \prod_{n=1}^N w_{n,i_n}(p_n) = \prod_{n=1}^N \left(\sum_{i_n=1}^{I_n} w_{n,i_n}(p_n) \right) = 1,$$

mas o contrário não é verdadeiro. Isso implica que o politopo definido por (4.14) e (4.15) é mais restrito do que o definido por (4.9) e (4.10). A matriz do sistema $\mathbf{S}(\mathbf{p})$ de um modelo NN e SN do tipo TP fica, portanto, dentro de um politopo definido de maneira mais precisa pelos sistemas LIT de vértice $\mathbf{S}_{i_1, i_2, \dots, i_N}$, para $\mathbf{p} \in \Omega$, da estrutura do produto tensorial.

De outra perspectiva, as condições (4.14) e (4.15) refinam a representação da matriz do sistema como uma combinação convexa de funções de coeficiente multivariável de $\mathbf{p} \in \Omega$ para as funções de coeficiente de variável única $p_n \in [a_n, n_n]$, com $n = 1, \dots, N$. Para este efeito, damos uma caracterização especial para o modelo TP que satisfaz as condições NN e SN.

Definição 4.10 (Modelo TP convexo). O modelo TP (4.13) é convexo se for do tipo SN e NN.

Um exemplo de modelo TP convexo são os modelos nebulosos TS na forma:

$$\begin{aligned} \text{Se } p_1 \text{ é } M_{1,i_1} \text{ e } p_2 \text{ é } M_{2,i_2} \text{ e } \dots \text{ e } p_N \text{ é } M_{N,i_N}, \\ \text{então } \mathbf{S}_{i_1,i_2,\dots,i_N}, \end{aligned} \quad (4.16)$$

para $n = 1, \dots, N$, $i_n = 1, \dots, I_n$. Nesse caso, as variáveis premissas p_n , $n = 1, \dots, N$, constituem o parâmetro \mathbf{p} ; as funções de pertinência do antecedente M_{n,i_n} da variável nebulosa p_n constituem as funções de ponderação $w_{n,i_n}(p_n)$; e os consequentes das regras nebulosas constituem os elementos das matrizes LIT $\mathbf{S}_{i_1,i_2,\dots,i_N}$. Lembrando que essas matrizes armazenam as matrizes do sistema, fica fácil perceber que (4.16) é equivalente a (3.3).

4.3 TRANSFORMAÇÃO DO PRODUTO TENSORIAL

Como mostrado na seção anterior, é possível representar modelos TS a partir de modelos TP. Tal representação permite o uso de LMIs para a síntese de controladores, ou para análise do sistema. Para obter modelos TP, usamos a técnica da transformação do produto tensorial. A premissa básica usual para o uso da técnica de transformação do produto tensorial é de que a função para a qual estamos procurando uma representação convexa pode ser exatamente representada como a interpolação, de alguma ordem (seja constante por parte, linear por partes, quadrática por partes, ou alguma ordem mais alta), das amostras utilizadas. O que equivale a dizer que o erro de aproximação entre a função original e a função interpolada não é levado em conta nesse procedimento. Um requisito para garantir que obteremos uma representação convexa no final da transformação é que o esquema de interpolação apenas gere valores dados pela combinação convexa dos pontos de amostra (CAMPOS, 2015).

Em resumo, a técnica da transformação do produto tensorial pode ser dividida em quatro passos:

1. Amostragem: amostrar o sistema de matrizes (4.6) sobre o espaço hiper-retangular Ω e armazenar as matrizes discretas $\mathbf{S}^D(\mathbf{p}(t))$ dentro do tensor \mathcal{S}^D , em que o sobrescrito "D" significa discreto;
2. HOSVD: extrair o sistema de vértice LIT e as matrizes de peso por meio da aplicação da HOSVD no tensor \mathcal{S}^D . Esta etapa é especializada em encontrar o número mínimo de sistemas de vértices LIT necessários para representar o sistema de matrizes;
3. Representação convexa: para que a representação obtida seja semelhante a um modelo TS é necessária uma transformação para que os pesos, armazenados nas matrizes de peso, tenham características semelhantes às de funções de pertinência ((4.14) e (4.15));
4. Aproximação para as matrizes de peso: para que existam valores para pontos que não pertençam ao *grid*, é necessário encontrar funções contínuas que representem os valores

contidos nos vetores de pesos (colunas das matrizes de peso). Isso é feito por interpolações dos valores discretos.

A seguir, esses passos são apresentados em detalhes. No final, um exemplo prático de aplicação da técnica é mostrado.

4.3.1 DISCRETIZAÇÃO

Esta etapa converte numericamente o sistema de matrizes dependente de parâmetro do sistema fornecido, $\mathbf{S}(\mathbf{p})$, em uma representação tensorial. Para isso, $\mathbf{S}(\mathbf{p})$ é amostrada dentro do espaço de transformação Ω no qual esperamos que o sistema seja válido. As amostras são salvas em um tensor.

Definição 4.11 (Transformação do espaço Ω). Ω é um espaço hiper-retangular limitado onde o vetor parâmetro da matriz do sistema varia: $\mathbf{p} \in \Omega : [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_N, b_N]$. Na prática, Ω deve ser definido de acordo com o espaço de trabalho de \mathbf{p} determinado com base na consideração física do modelo. Devemos notar que as funções de ponderação do modelo TP resultante serão definidas ao longo dos intervalos $[a_n, b_n]$, $n = 1, \dots, N$, então o modelo TP resultante é interpretável apenas em Ω .

Definição 4.12 (*Grid* de discretização M). M denota um *grid* (grade) de discretização hiper-retangular definido em Ω . M_n denota o número de linhas do *grid* na n -ésima dimensão, com $n = 1, \dots, N$ e $a_n \leq g_{n,m_n} \leq b_n$, $m_n = 1, \dots, M_n$, denotam as localizações correspondentes das linhas do *grid*. Geralmente, as linhas do *grid* podem ser arbitrariamente localizadas nos intervalos; entretanto, (BARANYI; YAM; VÁRLAKI, 2014) sugerem definir uma localização equidistante, $g_{n,m_n} = a_n + \frac{b_n - a_n}{M_n - 1}(m_n - 1)$, se não houver nenhuma razão especial para fazer o contrário. Um ponto no *grid* em Ω é, portanto, definido pelo vetor de coordenadas

$$\mathbf{g}_{m_1, m_2, \dots, m_N} = \begin{bmatrix} g_{1, m_1} \\ \vdots \\ g_{N, m_N} \end{bmatrix}. \quad (4.17)$$

Definição 4.13 (Discretização do modelo qLPV). \mathcal{S}^D , em que o sobrescrito "D" denota "discreto", é o sistema de matrizes $\mathbf{S}(\mathbf{p}) \in \mathbb{R}^{(m+k) \times (m+l)}$ discretizado sobre o *grid* hiper-retangular M dentro de Ω . Isso significa que as entradas de \mathcal{S}^D são

$$\mathcal{S}_{m_1, m_2, \dots, m_N} = \mathbf{S}(\mathbf{g}_{m_1, m_2, \dots, m_N}),$$

em que $m_n = 1, \dots, M_n$ e $n = 1, \dots, N$. Assim, o tamanho de \mathcal{S}^D é $(M_1 \times M_2 \times \cdots \times M_N \times (m+k) \times (m+l))$.

Observe que podemos definir um *grid* de discretização tão denso quanto necessário para garantir que o tensor discretizado descreva a matriz do sistema com precisão suficiente em Ω . Especificamente, desejamos expressar \mathcal{S}^D de acordo com a forma (4.7):

$$\mathcal{S}^D = \mathcal{S} \bigotimes_{n=1}^N \mathbf{w}_n^D(p_n), \quad (4.18)$$

em que $\mathbf{w}_n^D(p_n)$ são valores das funções de ponderação $\mathbf{w}_n(p_n)$ (4.13) calculadas nos valores discretizados de $p_n = g_{n,m_n}$, $m_n = 1, \dots, M_n$, sobre a n -ésima dimensão do intervalo $[a_n, b_n]$.

4.3.2 APLICAÇÃO DO HOSVD

Definido o tensor \mathcal{S}^D , que armazena as amostras do sistema de matrizes a ser aproximado, utiliza-se a HOSVD para decompô-lo. Após a execução da técnica, tem-se

$$\mathcal{S}^D = \mathcal{S} \bigotimes_{n=1}^N \mathbf{U}_n. \quad (4.19)$$

O modelo qLPV, representando em Ω por \mathcal{S}^D , pode ser interpretado como uma soma ponderada de vários modelos lineares. Nesse caso, $\mathcal{S} \in \mathbb{R}^{I_1 \times \dots \times I_N \times (m+k) \times (m+l)}$ é um tensor que armazena esses modelos, e cada matriz de peso $\mathbf{U}_n \in \mathbb{R}^{M_n \times I_n}$ representa os pesos relativos à variável p_n do *grid* de amostras. Cada vetor coluna dessas matrizes representa uma função de peso diferente para essa variável.

É importante notar que a HOSVD é executada apenas para as N primeiras dimensões, já que as duas últimas são os dados (lembrando que cada dado é uma matriz, que é construída pelas matrizes do sistema). Comparando (4.19) com (4.18), temos:

- I_n é o número de funções de peso ao longo da dimensão n , para $n = 1, \dots, N$;
- O tensor $\mathcal{S} \in \mathbb{R}^{(I_1 \times I_2 \times \dots \times I_N \times (m+k) \times (m+l))}$ contém as matrizes $\mathbf{S}_{i_1, i_2, \dots, i_n} \in \mathbb{R}^{(m+k) \times (m+l)}$, $i_n = 1, \dots, I_n$, como elementos. As matrizes $\mathbf{S}_{i_1, i_2, \dots, i_N} \in \mathbb{R}^{(m+k) \times (m_l)}$ são os sistemas lineares invariantes no tempo (LIT);
- O i_n -ésimo vetor coluna \mathbf{u}_{n, i_n} da matriz $\mathbf{U}_n \in \mathbb{R}^{M_n \times I_n}$ determina uma função de ponderação discretizada $w_{n, i_n}(p_n)$ da variável p_n . O valor do m_n -ésimo elemento de \mathbf{u}_{n, i_n} , denotado como $\mathbf{u}_{n, i_n}(m_n)$, define o valor da i_n -ésima função de peso $w_{n, i_n}(p_n)$ no ponto $p_n = g_{n, m_n}$:

$$w_{n, i_n}(g_{n, m_n}) = u_{n, i_n}(m_n). \quad (4.20)$$

Conforme mencionado, o cálculo da HOSVD pode ser dividido em duas etapas (cálculo das matrizes singulares de modo- n e, em seguida, do tensor de núcleo). Durante a primeira etapa, é realizada uma decomposição de valores singulares de cada matriz de desdobramento do tensor encontrando-se as matrizes \mathbf{U}_n e os valores singulares de modo- n . Nesta etapa, cada coluna

$\mathbf{u}_{n,i_n}, i_n = 1, \dots, I_n$, encontrada corresponde a um vetor singular de modo- n . Mantendo apenas as colunas de valores singulares não nulos encontra-se uma representação exata para o tensor.

Retirando-se colunas com valores singulares não nulos, chega-se em uma representação de menor complexidade, porém com certo erro. Tal representação reduzida não têm a propriedade de ser a melhor representação reduzida (em termos de minimização do erro), como é o caso quando se utiliza tal redução para matrizes através da decomposição de valores singulares. Todavia, é possível mostrar que o erro de aproximação cometido por se retirar uma coluna com um valor singular de modo- n não nulo é limitado.

Propriedade 4.3. (Erro de aproximação limitado). (CUI; ZHANG; MA, 2016) Reescrevendo a representação do modelo TP (4.12) e a transformação dessa representação na forma política (4.7) para considerar o erro

$$\mathbf{S}(\mathbf{p}) \approx \mathcal{S} \bigotimes_{n=1}^N \mathbf{w}_n(p_n), \quad (4.21)$$

$$\mathbf{S}(\mathbf{p}) \approx \sum_{h=1}^H w_h(\mathbf{p}) \mathbf{S}_h, \quad (4.22)$$

a seguinte expressão é válida para o erro de aproximação ϵ

$$\epsilon = \left(\left\| \mathbf{S}(\mathbf{p}) - \sum_{h=1}^H w_h(\mathbf{p}) \mathbf{S}_h \right\|_{L_2} \right)^2 \leq \sum_k \sigma_k^2. \quad (4.23)$$

σ_k são os valores singulares diferentes de zero descartados durante a HOSVD.

Essa é, portanto, a etapa da técnica da transformação do produto tensorial que permite fazer uma escolha entre a complexidade final da representação (número de funções de peso para cada variável, que é equivalente ao número de colunas das matrizes \mathbf{U}_n) e a precisão da aproximação.

4.3.3 REPRESENTAÇÕES CONVEXAS

Nesse ponto, o uso da HOSVD permitiu a decomposição do tensor de amostras, \mathcal{S}^D , que representa o sistema de matrizes do sistema, $\mathbf{S}(\mathbf{p})$, dentro do intervalo de amostragem Ω , em uma soma de vários modelos lineares, armazenados no tensor \mathcal{S} , ponderados por funções de peso discretas, que são vetores salvos dentro das matrizes de peso \mathbf{U}_n .

Conforme mencionado, para que a representação obtida seja semelhante a um modelo TS, (3.6), é necessário que as funções de peso tenham características semelhantes às de funções de pertinência, (3.4), de um conjunto nebuloso. No caso da técnica da transformação do produto tensorial, as características das funções de peso são exigidas na forma de características das matrizes de peso \mathbf{U}_n , que por sua vez podem ser vistas como características do tensor de núcleo \mathcal{S} .

Seria ideal se pudéssemos definir diretamente os vértices do fecho convexo¹ definindo os sistemas de vértices LIT e depois computando as funções de ponderação de acordo para formar um fecho convexo limitado. Isso, no entanto, não é possível, porque os sistemas de vértices e as funções de ponderação são acoplados para efetuar uma aproximação aceitável do modelo qLPV. Não se pode mudar arbitrariamente um sem mudar o outro, pois isso poderia prejudicar a precisão do modelo TP (BARANYI; YAM; VÁRLAKI, 2014).

Para impor características sobre uma determinada matriz de peso, esta é multiplicada a direita por uma matriz de transformação quadrada. Para que tal transformação seja válida, é necessário que a matriz de transformação possua inversa. Considerando $\bar{\mathbf{U}}_n = \mathbf{U}_n \mathbf{T}_n$, tem-se que $\mathbf{U}_n = \bar{\mathbf{U}}_n \mathbf{T}_n^{-1}$. Utilizando as propriedades (4.1) e (4.2), é possível reescrever (4.19) como:

$$\mathcal{S}^D = \mathcal{S} \bigotimes_{n=1}^N \mathbf{U}_n = \mathcal{S} \bigotimes_{n=1}^N (\bar{\mathbf{U}}_n \mathbf{T}_n^{-1}) = \mathcal{S} \bigotimes_{n=1}^N \mathbf{T}_n^{-1} \bigotimes_{n=1}^N \bar{\mathbf{U}}_n = \bar{\mathcal{S}} \bigotimes_{n=1}^N \bar{\mathbf{U}}_n, \quad (4.24)$$

em que,

$$\bar{\mathcal{S}} = \mathcal{S} \bigotimes_{n=1}^N \mathbf{T}_n^{-1}. \quad (4.25)$$

A ideia aqui é que, embora a matriz \mathbf{U}_n contenha colunas ortonormais que podem não produzir nenhuma convexidade, podemos projetar transformações adequadas, \mathbf{T}_n , para injetar vários tipos de convexidade em $\bar{\mathbf{U}}_n$.

Na literatura, as características normalmente são definidas em relação às funções de peso (obtidas após a interpolação das colunas das matrizes de peso), que são geradas a partir das matrizes \mathbf{U}_n , matrizes essas que contém as funções de peso discretas obtidas após a HOSVD. Entretanto, como as transformações utilizadas para se obter as características desejadas são realizadas sobre as matrizes de peso, neste texto as definições são feitas a partir delas. A seguir são apresentadas as características normalmente utilizadas na literatura e o porquê elas são desejáveis.

4.3.3.1 Soma Normalizada e Não Negatividade

Definição 4.14 (Soma normalizada (SN)). Uma matriz \mathbf{U}_n é dita normalizada em soma se a soma de suas colunas, \mathbf{u}_{n,i_n} , $i_n = 1, \dots, I_n$, resulta em um vetor cujos componentes são todos iguais a um (representado por $\mathbf{1}$).

Definição 4.15 (Não Negatividade (NN)). Uma matriz \mathbf{U}_n é dita não negativa se nenhum dos seus elementos é negativo.

Quando todas as matrizes de peso são SN e NN, o modelo obtido equivale a um modelo TS, e as funções de peso (correspondentes às colunas das matrizes de peso) são equivalentes a funções de pertinência de um modelo TS. Sendo assim, sempre que se é desejável encontrar uma aproximação nebulosa TS, deseja-se que todas as matrizes de peso sejam SN e NN.

¹O fecho convexo $FC(X)$ de um conjunto $X \subset \mathcal{R}^n$ é a menor região convexa do \mathcal{R}^n que contém o conjunto X .

Propriedade 4.4. (NAGY et al., 2009) Para que a transformação para funções de peso do tipo SN seja possível, sem alterar o número de colunas, é necessário que o vetor $\mathbf{1}$ pertença ao espaço imagem da matriz de peso que se deseja transformar. Quando ele não pertence, é necessária a inclusão de uma nova coluna na matriz de modo que passe a pertencer. Nesse caso, a transformação SN tem uma resposta trivial que é dada pelo acréscimo de uma nova coluna dada por

$$\mathbf{1} - \sum_{i=1}^{I_n} \mathbf{u}_{n,i}.$$

Esse acréscimo é feito ou durante o cálculo da HOSVD, ou, para que a dimensão seja condizente, são inseridos zeros correspondentes a essa nova coluna no tensor de núcleo.

Para que a representação (4.19) seja convexa bastam as propriedades SN e NN. No entanto, representações com propriedades adicionais podem apresentar melhores resultados quando utilizadas para a síntese de controladores (CAMPOS; TÔRRES; PALHARES, 2011).

4.3.3.2 Normalizada e Próxima Normalizada

Definição 4.16 (Normalizada (NO)). Uma matriz \mathbf{U}_n é dita normalizada se ela é SN e NN e, além disso, os valores máximos de cada coluna são os mesmos e iguais a 1.

Quando as matrizes de peso do sistema são NO, o conjunto representado pelo politopo formado pelos vértices armazenados no tensor de núcleo equivale ao fecho convexo dos pontos amostrados utilizados para a técnica da transformação do produto tensorial. Entretanto, na maior parte dos casos, essa exigência pode tornar necessário um número muito grande de colunas, aumentando muito a complexidade da aproximação.

Definição 4.17 (Próxima Normalizada (CNO)). Uma matriz \mathbf{U}_n é dita próxima a normalizada se ela é SN e NN e, além disso, os valores máximos de cada coluna são próximos ou iguais a 1.

No caso da representação CNO, o conjunto representado pelo politopo não equivale mais ao fecho convexo. Porém nesse caso, busca-se por uma representação com um número certo de regras (colunas das matrizes de peso), que seja próxima a representação NO, de modo que o politopo seja próximo ao fecho convexo. Por isso, na maior parte dos casos, busca-se uma representação CNO ao invés de NO, pois normalmente buscam-se representações de complexidade reduzida.

Por buscar-se uma representação na qual o politopo seja próximo ao fecho convexo na representação CNO, neste trabalho utiliza-se uma definição alternativa para tal propriedade.

Definição 4.18 (Próxima Normalizada (CNO) - definição alternativa). Uma matriz \mathbf{U}_n é dita próxima a normalizada se ela é SN e NN e, além disso, o fecho convexo dos seus vetores linha tem o maior volume possível dentro do simplex unitário padrão (observação: não é possível

alargar o fecho convexo sem que as condições de SN e NN deixem de ser atendidas) (CAMPOS, 2015).

A partir dessa definição, temos que a transformação CNO será aquela que encontra o menor simplex² que cobre os vetores coluna da matriz de peso. Transformando esse simplex no simplex unitário³ temos uma matriz CNO.

4.3.3.3 Normalizada Relaxada e Normalizada Inversa

Definição 4.19 (Normalizada Relaxada (RNO)). Uma matriz U_n é dita normalizada relaxada se ela é SN e NN e, além disso, os valores máximos de cada coluna são os mesmos. Esses valores estão sempre no intervalo $[0, 1]$.

Definição 4.20 (Normalizada Inversa (INO)). Uma matriz U_n é dita normalizada inversa se ela é SN e NN e, além disso, os valores mínimos de cada coluna são os mesmos e iguais a zero.

Assim como a característica CNO, as características RNO e INO são formas relaxadas de se impor características sobre o politopo de modo que ele represente um conjunto menor do que quando se impõe apenas as características de SN e NN sobre as matrizes de peso.

É interessante notar que para matrizes de duas colunas, a condição INO é equivalente à condição NO. Além disso, a imposição de que todas as colunas da matriz de peso possuam o mesmo valor máximo (RNO) em conjunto com a imposição de que todas as colunas possuam um mínimo igual a zero (INO) faz com que as funções de peso tenham uma distribuição melhor, sendo que nenhuma função de peso causará uma importância maior para seu vértice correspondente.

A Figura 6 mostra a implicação geométrica da convexidade das funções de ponderação no fecho convexo resultante de $S(\mathbf{p})$. Considere $S(\mathbf{p})$ na forma TP (4.11) sendo mapeado de Ω para seu espaço de alcance. Como a combinação linear em (4.11) é convexa, os vértices do fecho convexo de $S(\mathbf{p})$ são, portanto, dados pelos sistemas de vértices LIT S_{i_1, \dots, i_N} . Em modelos TP do tipo NO, $S(\mathbf{p})$ será igual a cada um dos vértices, S_{i_1, \dots, i_N} , nos valores apropriados de $\mathbf{p} \in \Omega$. Isso ocorre quando p_n está em tal valor que a \bar{i}_n -ésima função de ponderação da n -ésima dimensão atinge o valor de 1 para $n = 1, \dots, N$. Como as funções de ponderação NO também são SN e NN, isso implica um valor zero para todas as outras funções de ponderação. A forma TP de $S(\mathbf{p})$ determina que $S(\mathbf{p}) = S_{\bar{i}_1, \dots, \bar{i}_N}$. Como \bar{i}_n vai de 1 a I_n , para $n = 1, \dots, N$, a imagem mapeada de $S(\mathbf{p})$ no espaço de alcance visitará cada um dos sistemas LIT de vértice S_{i_1, \dots, i_N} . Isso é ilustrado na Figura 6(a).

Para um modelo TP do tipo CNO, nem todas as funções de ponderação podem adquirir valor igual 1 para $p_n \in [a_n, b_n]$. Para essas funções de ponderação, o máximo que elas podem adquirir são valores próximos a 1, com outras funções de ponderação na mesma dimensão

²Em geometria, simplex é o politopo mais simples possível em qualquer espaço: um 0-simplex é um ponto, um 1-simplex é um segmento de linha, um 2-simplex é um triângulo, etc.

³ K -simplex unitário é o conjunto de pontos $x \in \mathcal{R}^K$ tal que para $1 \leq k \leq K$, $x_k > 0$ e $\sum_{k=1}^K x_k = 1$.

assumindo valores pequenos, mas diferentes de zero. Por outro lado, ainda é possível para um modelo TP do tipo CNO que algumas funções de ponderação adquiram os valores de 1 para $p_n \in [a_n, b_n]$, $n = 1, \dots, N$, levando $S(p)$ a ser mapeado exatamente em alguns vértices. A situação é ilustrada na Figura 6(b), mostrando que a imagem mapeada de $S(p)$ se desvia de alguns $S(p) = S_{\bar{i}_1, \dots, \bar{i}_n}$, embora seja exatamente igual a alguns outros. Nesse caso, o “grau” dos desvios, se ocorrerem, pode variar de vértice a vértice. Isso ocorre porque a convexidade do tipo CNO não exige que os valores máximos das funções de ponderação sejam os mesmos, apenas que sejam 1, ou próximos de 1. Portanto, a contribuição do vértice dominante em (4.11) varia de vértice para vértice e resulta em vários “graus” de desvio.

A Figura 6(c) mostra o caso RNO, onde exigimos que o valor máximo de todas as funções de ponderação na mesma dimensão tenham o mesmo valor próximo a 1. Em comparação com o caso CNO, vemos que $S(p)$ não passará por nenhum dos vértices do sistema LIT. Isso ocorre porque nenhuma função de ponderação pode adquirir o valor 1 no caso RNO, e então $S(p)$ certamente será dado pela combinação linear de pelo menos dois vértices. Além disso, o “grau” de desvios da imagem mapeada de $S(p)$ de vértice para vértice é mais uniforme no caso RNO. Isso ocorre porque a contribuição do vértice dominante para $S(p)$ é mais uniforme em todos os vértices, resultado do mesmo valor próximo a 1 obtido nas funções de ponderação correspondentes.

Finalmente, o caso do modelo TP do tipo INO apenas garante que alguns vértices não contribuirão para $S(p)$ em valores específicos de $p \in \Omega$, pois suas funções de ponderação correspondentes são de valor zero. Observe que o tipo NO é um caso especial do tipo INO (quando $S(p)$ é mapeado em um vértice específico, nenhum dos outros vértices está contribuindo).

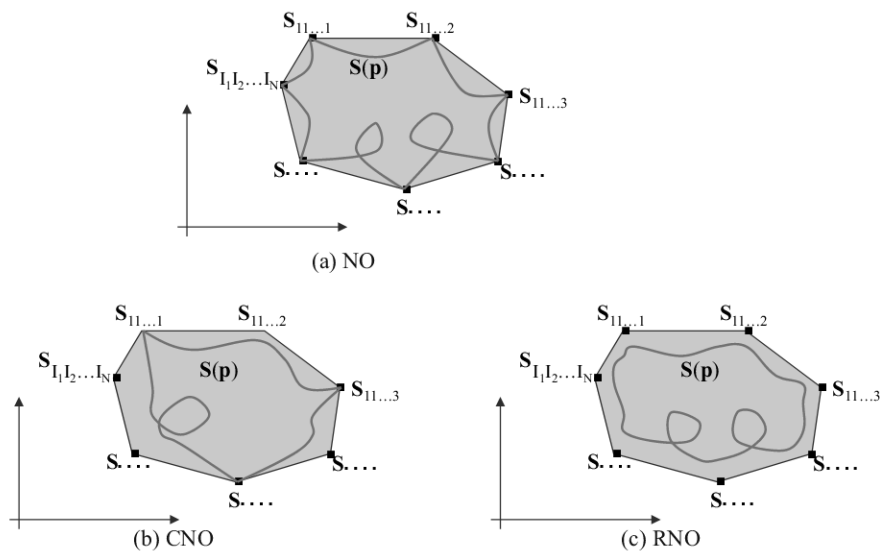


Figura 6 – Implicação geométrica das convexitades das funções de ponderação no fecho convexo resultante.

Fonte : (BARANYI; YAM; VÁRLAKI, 2014)

4.3.4 APROXIMAÇÃO PARA AS MATRIZES DE PESO

Tendo as matrizes de peso com as características desejadas é necessário que se encontre funções que substituam suas colunas, de modo que a aproximação encontrada possa ser utilizada para pontos que não pertençam ao *grid*. Ou seja, as funções de ponderação contínuas são geradas a partir das matrizes \mathbf{U}_n que contêm as funções de peso discretas.

Após aplicar as transformações convexas desejadas, podemos reescrever (4.19) como

$$\mathcal{S}^D = \mathcal{S} \otimes_{n=1}^N \mathbf{U}_n = \mathcal{S}^{CO} \otimes_{n=1}^N \mathbf{U}_n^{CO}, \quad (4.26)$$

em que "CO" indica convexo, ou seja, \mathcal{S}^{CO} e \mathbf{U}_n^{CO} são, respectivamente, \mathcal{S} e \mathbf{U}_n com as propriedades convexas desejadas. Vale ressaltar, que basta que as transformações NN e SN sejam atendidas para que haja convexidade.

O sistema de matrizes é aproximado pelo tensor de amostras ($\mathbf{S}(\mathbf{p}) \approx \mathcal{S}^D$) dentro do espaço de amostragem (Ω). Podemos reescrever (4.11) como

$$\begin{aligned} \mathbf{S}(\mathbf{p}) &= (\mathcal{S}^{CO} \otimes_{n=1}^N \mathbf{U}_n^{CO}) \otimes_{n=1}^N \mathbf{w}_n(p_n) \implies \\ \mathbf{S}(\mathbf{p}) &= \mathcal{S}^{CO} \otimes_{n=1}^N (\mathbf{w}_n(p_n) \mathbf{U}_n^{CO}) \implies \\ \mathbf{S}(\mathbf{p}) &= \mathcal{S}^{CO} \otimes_{n=1}^N \mathbf{w}_n^{CO}(p_n), \end{aligned} \quad (4.27)$$

com

$$\mathbf{w}_n^{CO}(p_n) = \mathbf{w}_n(p_n) \mathbf{U}_n^{CO}. \quad (4.28)$$

A equação (4.28) pode ser vista como uma operação de interpolação entre as linhas das matrizes de peso \mathbf{U}_n^{CO} usando as funções de ponderação $\mathbf{w}_n(p_n)$.

O exemplo a seguir foi retirado do trabalho de (BARANYI; YAM; VÁRLAKI, 2014). Para manter a coerência com os resultados obtidos pelos autores, usou-se, para esse exemplo, os códigos Matlab® criados por eles. Porém, esses códigos não foram usados no restante do trabalho.

Exemplo 2: TORA

O TORA, abreviatura para *Translational Oscillations with a Rotational Actuator*, em português, oscilações translacionais com um atuador rotacional, consiste em uma plataforma que pode oscilar sem amortecimento no plano horizontal sem efeito da gravidade. Na plataforma, uma massa excêntrica rotativa é acionada por um motor CC. Seu movimento aplica uma força à plataforma que pode ser usada para amortecer as oscilações translacionais (JANKOVIC; FONTAINE; KOKOTOVIĆ, 1996). Esse mecanismo pode ser descrito pelo sistema qLPV

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{p}(t))\mathbf{x} + \mathbf{B}(\mathbf{p}(t))\mathbf{u}.$$

As matrizes do sistema, dependentes dos parâmetros, são

$$\mathbf{A}(\mathbf{p}(t)) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ a_{21} & 0 & 0 & a_{24} \\ 0 & 0 & 0 & 1 \\ a_{41} & 0 & 0 & a_{44} \end{bmatrix}, \quad \mathbf{B}(\mathbf{p}(t)) = \begin{bmatrix} 0 \\ b_{21} \\ 0 \\ b_{41} \end{bmatrix},$$

em que, omitindo as dependências temporais de $\mathbf{p}(t)$ e $\mathbf{x}(t)$,

$$a_{21} = \frac{-1}{1-\epsilon^2 \cos^2(p_1)}, \quad a_{24} = \frac{\epsilon p_2 \sin(p_1)}{1-\epsilon^2 \cos^2(p_1)}, \quad a_{41} = \frac{\epsilon \cos(p_1)}{1-\epsilon^2 \cos^2(p_1)}, \quad a_{44} = \frac{-\epsilon^2 p_2(t) \cos(p_1) \sin(p_1)}{1-\epsilon^2 \cos^2(p_1)};$$

$$b_{21} = \frac{-\epsilon \cos(p_1)}{1-\epsilon^2 \cos^2(p_1)}, \quad b_{41} = \frac{1}{1-\epsilon^2 \cos^2(p_1)}; \quad \mathbf{p} = [p_1 \ p_2]^T = [x_3 \ x_4]^T; \quad \epsilon = 0, 2.$$

O sistema de matrizes é assim definido

$$\mathbf{S}(\mathbf{p}) = [\mathbf{A}(\mathbf{p}) \ \mathbf{B}(\mathbf{p})].$$

O intervalo de discretização, que será o mesmo intervalo no qual o modelo TP será válido, escolhido foi $p_1 = x_3 \in [-a, a]$, $p_2 = x_4 \in [-a, a]$, com $a = \frac{45}{180} \pi \frac{\text{rad}}{\text{s}}$. Ou seja, $\Omega = [-a, a] \times [-a, a]$. A densidade de discretização do *grid* foi definida como 137×137 . Então, o tensor que armazena as amostras dos modelos lineares matriciais é $\mathcal{S}^D \in \mathbb{R}^{137 \times 137 \times (4 \times 5)}$.

Usando o HOSVD em \mathcal{S}^D , obtemos os seguintes valores singulares: $\sigma_{11} = 280,99$, $\sigma_{12} = 5,59482$, $\sigma_{13} = 3,74984$, $\sigma_{14} = 0,0851335$ e $\sigma_{15} = 0,0365394$ na direção $p_1 = x_3$; $\sigma_{21} = 281,015$ e $\sigma_{22} = 5,59547$ na direção $p_2 = x_4$. Mantendo-se todos os valores singulares, o sistema do TORA pode ser representado pela combinação de 5×2 vértices LIT

$$\dot{\mathbf{x}} = \mathbf{S}(\mathbf{p}) \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = (\mathcal{S} \otimes_{n=1}^2 \mathbf{w}_n(p_n)) \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \sum_{i=1}^5 \sum_{j=1}^2 w_{1,i}(p_1) w_{2,j}(p_2) (\mathbf{A}_{i,j} \mathbf{x} + \mathbf{B}_{i,j} \mathbf{u}), \quad (4.29)$$

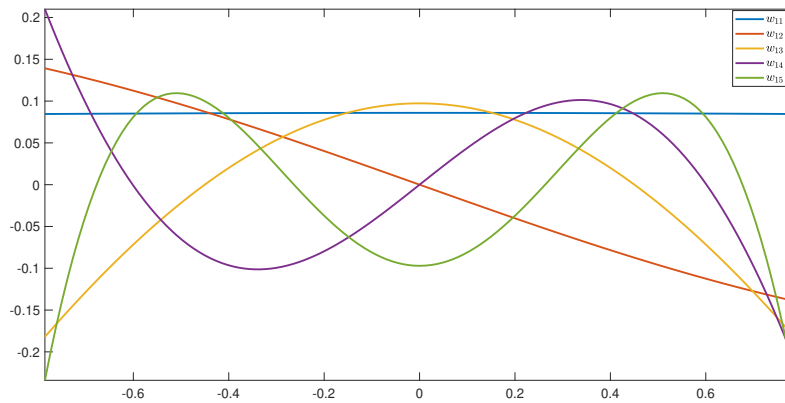
A Figura 7 mostra as funções de ponderação obtidas pelas interpolações dos vetores colunas $\mathbf{w}_{11} = \mathbf{w}_1(p_1) \in \mathbb{R}^{137 \times 1}$, $\mathbf{w}_{12} = \mathbf{w}_2(p_1) \in \mathbb{R}^{137 \times 1}$, $\mathbf{w}_{13} = \mathbf{w}_3(p_1) \in \mathbb{R}^{137 \times 1}$, $\mathbf{w}_{14} = \mathbf{w}_4(p_1) \in \mathbb{R}^{137 \times 1}$, $\mathbf{w}_{15} = \mathbf{w}_5(p_1) \in \mathbb{R}^{137 \times 1}$ e $\mathbf{w}_{21} = \mathbf{w}_1(p_2) \in \mathbb{R}^{137 \times 1}$, $\mathbf{w}_{22} = \mathbf{w}_2(p_2) \in \mathbb{R}^{137 \times 1}$. Esses vetores são as colunas das matrizes de peso $\mathbf{U}_1 \in \mathbb{R}^{137 \times 5}$ e $\mathbf{U}_2 \in \mathbb{R}^{137 \times 2}$, respectivamente. Observa-se que essas funções não obedecem às propriedades convexas (4.14) e (4.15) e, portanto, o modelo politópico acima não é um modelo convexo.

Mantendo-se dois valores singulares em cada direção, na etapa de aplicação da HOSVD, e aplicando a etapa de manipulação do fecho convexo às matrizes de peso, impondo-lhes

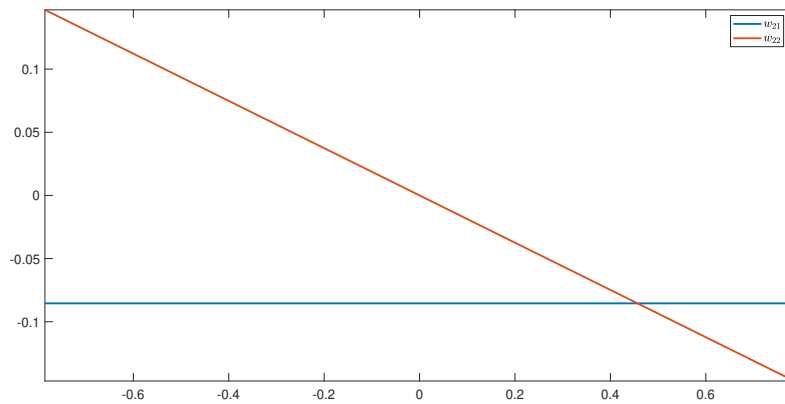
a propriedade CNO, chegamos a um modelo TP equivalente a um modelo TS com seis regras, uma vez que fomos forçados a adicionar outra regra para a primeira direção a fim de respeitar a Propriedade 4.4 da soma normalizada (SN). Nesse caso, o sistema do TORA pode ser representado pela combinação de 3×2 vértices LIT

$$\dot{\mathbf{x}} = (\mathcal{D}^{CO} \otimes_{n=1}^2 \mathbf{w}_n^{CO}(p_n)) \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \sum_{i=1}^3 \sum_{j=1}^2 w_{1,i}^{CO}(p_1) w_{2,j}^{CO}(p_2) (\mathbf{A}_{i,j} \mathbf{x} + \mathbf{B}_{i,j} \mathbf{u}), \quad (4.30)$$

em que $\mathcal{D}^{CO} \in \mathbb{R}^{3 \times 2 \times (4 \times 5)}$ indica que houve descarte de valores singulares diferentes de zero, e esse tensor já passou pela etapa de convexificação; $\mathbf{w}_n^{NO}(p_n) \in \mathbb{R}^{137 \times 1}$ são funções de peso obtidas pelas colunas de $\mathbf{U}_1 \in \mathbb{R}^{137 \times 3}$ e $\mathbf{U}_2 \in \mathbb{R}^{137 \times 2}$ depois dessas matrizes passarem pelo processo de convexificação e interpolação de seus valores. Essas funções podem ser vistas na Figura 8.

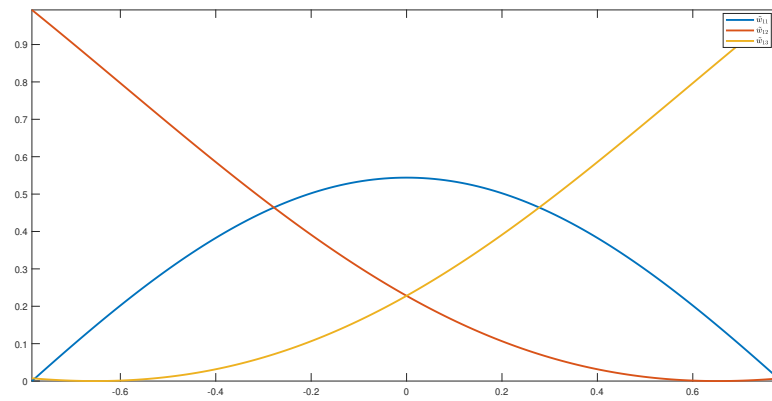


(a) Funções de ponderação na direção $x_3(t)$.

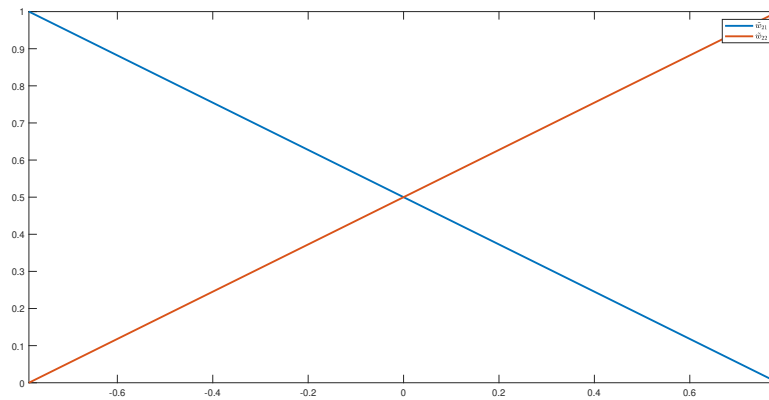


(b) Funções de ponderação na direção $x_4(t)$.

Figura 7 – Funções de ponderação obtidas após a aplicação do HOSVD.



(a) Funções de ponderação com propriedade CNO na direção $x_3(t)$.



(b) Funções de ponderação com propriedade CNO na direção $x_4(t)$.

Figura 8 – Funções de ponderação obtidas após a redução de regras e aplicação de propriedades de convexificação.

ESTABILIDADE DE SISTEMAS DINÂMICOS VIA DESIGUALDADES MATRICIAIS LINEARES

Resumo do capítulo

Desigualdade matricial linear (LMI), em inglês linear matrix inequality, é uma classe de problemas de otimização numérica. Para sistemas e controle, a importância da otimização de LMI decorre do fato de que uma ampla variedade de problemas de sistema e controle podem ser formulados como problemas de LMI. Lyapunov desenvolveu um método para investigar a estabilidade de sistemas não lineares baseado na ideia de encontrar funções com propriedades especiais. A grande valia do método de Lyapunov está no fato de que ele pode ser aplicado a qualquer sistema dinâmico. Ao escolher uma função candidata apropriada e se constatar que sua derivada ao longo do sistema é semidefinida negativa, garante-se então a estabilidade. Porém, apesar de elegante, o teorema de Lyapunov não nos fornece métodos para construir funções de Lyapunov. Uma alternativa é usar LMIs para se construir numericamente essas funções.

Problemas de otimização são geralmente classificados em duas classes: P e NP-difícil. A classe P refere-se a problemas nos quais o tempo necessário para resolver exatamente o problema pode sempre ser limitado por uma única função que é polinomial na quantidade de dados necessária para definir o problema. Esses problemas podem ser resolvidos em tempo polinomial. Embora as consequências exatas de um problema ser NP-difícil ainda sejam uma questão aberta fundamental na teoria da complexidade computacional, é geralmente aceito que um problema ser NP-difícil significa que sua solução não pode ser calculada em tempo polinomial no pior caso. Algoritmos práticos para problemas NP-difíceis existem e normalmente envolvem aproximação, heurísticas ou busca local. Determinar a classe do problema informa ao projetista que tipo de precisão e velocidade podem ser esperados pelos melhores algoritmos e que tipos de algoritmos investigar para fornecer soluções práticas para o problema (VANANTWERP; BRAATZ, 2000).

5.1 DESIGUALDADES MATRICIAIS LINEARES

Uma classe de problemas de otimização numérica chamada de programação semi-definida, que faz o uso de desigualdades matriciais lineares (LMIs), em inglês *linear matrix inequalities*, recebeu atenção significativa. Esses problemas de otimização podem ser resolvidos em tempo polinomial e, portanto, são tratáveis, pelo menos em um sentido teórico. Para sistemas e controle, a importância da otimização com LMI decorre do fato de que uma ampla variedade de problemas de sistema e controle podem ser formulados como problemas de LMI. Exceto em alguns casos especiais, esses problemas não têm soluções analíticas. No entanto, o ponto principal

é que, com o uso de LMIs, eles podem ser resolvidos numericamente de maneira eficiente. Portanto, reformular um problema de controle como um problema LMI é equivalente a encontrar uma "solução" para o problema original (TANAKA; WANG, 2001).

A vantagem mais significativa das LMIs é a facilidade de se especificar e combinar numericamente várias restrições de projeto, condições e objetivos. Uma vez formulados em termos de LMIs, um problema pode ser resolvido através do uso dos algoritmos de otimização (PETRES, 2006).

Definição 5.1 (Desigualdade matricial linear). Uma desigualdade matricial linear (LMI) tem a seguinte forma (BOYD et al., 1994)

$$\mathbf{F}(\mathbf{x}) \stackrel{\text{def}}{=} \mathbf{F}_0 + \sum_{i=1}^m x_i \mathbf{F}_i > 0, \quad (5.1)$$

em que $\mathbf{x} \in \mathbb{R}^m$ é um vetor de m variável reais chamadas de *variáveis de decisão*; as matrizes $\mathbf{F}_0 \dots \mathbf{F}_m$ são matrizes reais e simétricas, $\mathbf{F}_i = \mathbf{F}_i^T \in \mathbb{R}^{n \times n}$, $i = 0, \dots, m$. A desigualdade mostra que $\mathbf{F}(x)$ é definida positiva, ou seja, $\mathbf{u}^T \mathbf{F}(x) \mathbf{u} > 0$ para qualquer vetor não nulo $\mathbf{u} \in \mathbb{R}^n$.

A restrição (5.1) consiste em uma restrição convexa, isto é, o conjunto $\{\mathbf{x} | \mathbf{F}(\mathbf{x}) > 0\}$ é convexo. Então, pegando-se duas variáveis de decisão x_1 e x_2 pertencentes a esse conjunto e tomando $\alpha \in [0, 1]$

$$\mathbf{F}(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha \mathbf{F}(x_1) + (1 - \alpha) \mathbf{F}(x_2). \quad (5.2)$$

Definição 5.2 (Sistemas de LMIs). Um sistema de desigualdades lineares matriciais é um conjunto finito de LMIs

$$\mathbf{F}_1(\mathbf{x}) > 0, \mathbf{F}_2(\mathbf{x}) > 0, \dots, \mathbf{F}_k(\mathbf{x}) > 0. \quad (5.3)$$

Uma simples, porém essencial, propriedade das LMIs diz que todo sistema de LMIs pode ser reescrito como uma LMI. Ou seja, (5.3) pode ser reescrito como

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} \mathbf{F}_1(\mathbf{x}) & 0 & \dots & 0 \\ 0 & \mathbf{F}_2(\mathbf{x}) & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{F}_k(\mathbf{x}) \end{bmatrix} > 0. \quad (5.4)$$

Dados $\mathbf{F}, \mathbf{G}: \mathbb{V} \mapsto \mathbb{S}$, existem basicamente três problemas genéricos relacionados a LMIs (BOYD et al., 1994):

- **Factibilidade:** O teste para saber se existem ou não soluções $\mathbf{x} \in \mathbb{V}$ para $\mathbf{F}(\mathbf{x}) > 0$ é chamado de *problema de factibilidade*. A LMI é chamada de *factível* se existe tal \mathbf{x} ; caso contrário, a LMI é chamada de *infactível*.

- **Otimização:** Suponhamos $f : A \mapsto \mathbb{R}$ e $A = \{\mathbf{x} | \mathbf{F}(\mathbf{x}) > 0\}$. O problema descrito por

$$V_{opt} = \inf_{\mathbf{x} \in A} f(\mathbf{x})$$

é chamado de *problema de otimização com uma restrição LMI*. Esse problema envolve encontrar V_{opt} e, para qualquer valor arbitrário $\epsilon > 0$, o cálculo de uma *solução quase ótima* \mathbf{x} que satisfaça $\mathbf{x} \in A$ e $V_{opt} \leq f(\mathbf{x}) \leq V_{opt} + \epsilon$.

- **Problema de autovalor generalizado:** O *problema de autovalor generalizado* equivale a minimizar um escalar $\lambda \in \mathbb{R}$ sujeito a

$$\begin{cases} \lambda \mathbf{F}(\mathbf{x}) - \mathbf{G}(\mathbf{x}) > 0 \\ \mathbf{F}(\mathbf{x}) > 0 \\ \mathbf{G}(\mathbf{x}) > 0 \end{cases}$$

A Equação (5.1) apresenta a chamada forma padrão das LMIs. Contudo, LMIs normalmente são escritas de outro modo, no qual as variáveis são matrizes. Um exemplo é a desigualdade de Lyapunov (BOYD et al., 1994)

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} < 0, \quad (5.5)$$

em que $\mathbf{A} \in \mathcal{R}^{n \times n}$ e $\mathbf{P} = \mathbf{P}^T$ é a variável. Nesse caso, a LMI não está explicitamente na forma (5.1). Essa forma de escrita de LMIs é mais conveniente em grande parte das aplicações, pois existem softwares, como o YALMIP, que permitem escrever LMIs diretamente a partir de variáveis matriciais.

5.2 TEORIA DE LYAPUNOV

Contribuições fundamentais para a teoria da estabilidade de sistemas não lineares foram feitas pelo matemático russo Lyapunov no final do século XIX (BOYD et al., 1994). Lyapunov investigou a equação diferencial não linear

$$\frac{dx(t)}{dt} = f(x), \quad f(0) = 0. \quad (5.6)$$

Quando $f(0) = 0$, a equação tem solução $x(t) = 0$. Para garantir que uma solução existe e é única, é necessário fazer algumas suposições sobre $f(x)$. Uma suposição suficiente é $f(x)$ ser localmente Lipschitz, ou seja,

$$\|f(x) - f(y)\| \leq L\|x - y\|, \quad L > 0, \quad (5.7)$$

na vizinhança da origem. Lyapunov se interessou em investigar se a solução de (5.6) é estável. Para este propósito, ele introduziu o seguinte conceito de estabilidade (ASTRÖM; WITTENMARK, 2008):

Definição 5.3 (Estabilidade no sentido de Lyapunov). A solução $x(t) = 0$ da equação (5.6) é chamada *estável* se dado $\epsilon > 0$ existir $\delta(\epsilon)$ de modo que todas as soluções com condições iniciais

$$\|x(0)\| < \delta$$

possuam a propriedade

$$\|x(t)\| < \epsilon \text{ para todo } 0 \leq t \leq \infty.$$

Caso contrário, a solução é *instável*.

Alguns casos de estabilidade são introduzidos abaixo. Especificamente para este trabalho, a estabilidade exponencial é destacada, já que os controladores PDC projetados usam esse conceito.

Definição 5.4 (Estabilidade assintótica). A solução é chamada *assintoticamente estável* se ela for estável e δ pode ser encontrado de forma que todas as soluções com $\|x(0)\| < \delta$ tem a propriedade de $\|x(t)\| \rightarrow 0$ quando $t \rightarrow \infty$.

Definição 5.5 (Estabilidade global). Se a solução for assintoticamente estável para qualquer valor inicial, então ela é considerada *globalmente assintoticamente estável*.

Definição 5.6 (Estabilidade exponencial). Dizemos que uma solução é *exponencialmente estável* quando, além de ser assintoticamente estável, a sua convergência para a origem é exponencial.

Lyapunov desenvolveu um método para investigar a estabilidade baseado na ideia de encontrar funções com propriedades especiais. Para descrevê-las, primeiramente vamos introduzir o conceito de funções definidas positivas.

Definição 5.7 (Funções positivas definidas e semidefinidas positivas). Uma função contínua e diferenciável $V : U \mapsto \mathbb{R}$ é chamada de definida positiva se a região $U \subset \mathbb{R}^n$ contém a origem e

1. $V(0) = 0$;
2. $V(\mathbf{x}) > 0$.

A função é chamada *semidefinida positiva* se a condição 2, referente à desigualdade, da Definição 5.7 for substituída por $V(x) \geq 0$.

Uma função definida positiva possui curvas de nível que envolvem a origem. As curvas correspondentes a valores maiores da função englobam curvas que correspondem a valores menores. A situação no caso bidimensional é ilustrada na Figura 9. Se pudermos encontrar uma função para que o vetor velocidade, $dx/dt = f(x)$, sempre aponte para o interior das curvas de nível, então parece intuitivamente claro que uma solução que começa dentro de uma determinada curva de nível nunca pode passar para o lado de fora da mesma curva de nível. Nós temos o seguinte teorema:

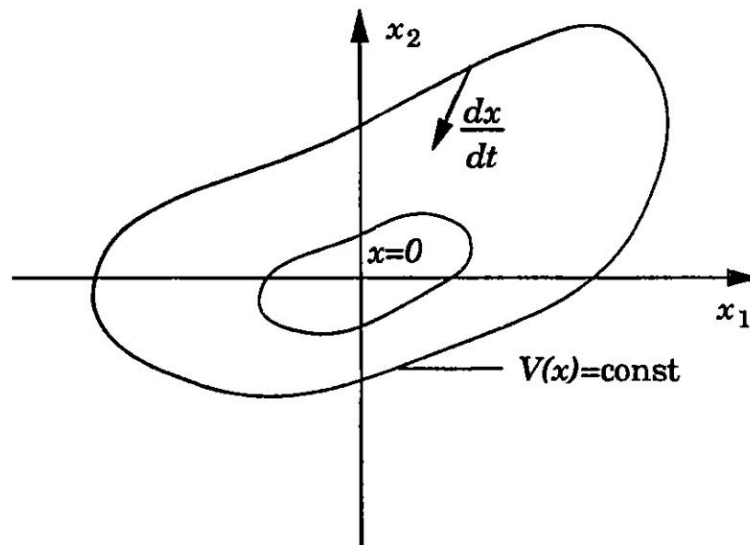


Figura 9 – Ilustração do método de Lyapunov para investigar estabilidade.

Fonte : (ASTRÖM; WITTENMARK, 2008)

Teorema 5.1 (Teorema da estabilidade de Lyapunov). *Se existir uma função $V : \mathbb{R}^n \mapsto \mathbb{R}$ definida positiva tal que sua derivada ao longo da solução de (5.6),*

$$\frac{dV}{dt} = \frac{\partial V^T}{\partial x} \frac{dx}{dt} = \frac{\partial V^T}{\partial x} f(x) = -W(x)$$

é semi-definida negativa, então a solução $x(t) = 0$ de (5.6) é estável. Se dV/dt é definida negativa, então a solução é assintoticamente estável. A função V é chamada de função de Lyapunov do sistema (5.6).

Caso

$$\frac{dV}{dt} < 0 \text{ e } V(x) \rightarrow \infty, \text{ quando } \|x\| \rightarrow \infty,$$

a solução é globalmente assintoticamente estável.

O conceito de estabilidade apresentado acima pode ser visto como uma extensão matemática de uma observação física: se a energia total de um sistema é dissipada continuamente, então esse sistema deve se acomodar em um ponto de equilíbrio. Nesse caso, a analogia que se faz é que a função de energia é dada por $V(x)$, sempre positiva, e sua derivada temporal ao longo das trajetórias, $\dot{V}(x) < 0$, representa a dissipação desta energia.

5.3 ANÁLISE DA ESTABILIDADE VIA LMIS

A grande valia do método de Lyapunov está no fato de que ele pode ser aplicado a qualquer sistema dinâmico. Ao escolher uma função candidata apropriada e se constatar que sua derivada ao longo do sistema é definida negativa, garante-se então a estabilidade. Porém, apesar de elegante, o teorema de Lyapunov não nos fornece métodos para construir funções de Lyapunov. Não existe um método universal para construir funções de Lyapunov para um sistema estável (ASTRÖM; WITTENMARK, 2008). Uma alternativa é usar LMIs para se construir numericamente essas funções, pois, conforme visto na seção anterior, elas permitem resolver problemas de factibilidade, permitindo constatar se a função candidata é solução ou não do problema.

5.4 PROGRAMAS PARA RESOLUÇÃO DE LMIS

Neste trabalho usaram-se:

- **YALMIP**: pacote gratuito para Matlab® para prototipagem rápida de problemas de otimização, que permite escrever as LMIs de uma forma simples e natural;
- **SeDuMi**: resolvidor para Matlab®, que permite resolver problemas de otimização com restrições lineares, quadráticas e semi-definidas.

CONTROLADOR PDC

Resumo do capítulo

Um tipo de controlador desenvolvido neste trabalho usa o conceito de projeto de compensação distribuída paralela (PDC). A ideia principal deste conceito é encontrar um controlador na mesma estrutura politópica do modelo TS inferido. Esta estrutura politópica combina os ganhos de realimentação LIT derivados dos sistemas LIT do modelo politópico. A lei de controle PDC pode ser vista como um ganho escalonado, em que controladores por realimentação de estados lineares tem sua contribuição ponderada por funções de pertinência, de forma que a saída seja uma soma convexa das contribuições individuais de cada um. Os ganhos de realimentação podem ser derivados por várias técnicas. Uma das técnicas de projeto mais poderosas, que é capaz de otimizar várias especificações de controle desejadas, é baseada em desigualdades matriciais lineares (LMIs) por meio de funções de Lyapunov.

A estrutura de projeto do controlador PDC, em inglês *parallel distributed compensation*, adotada neste trabalho consiste em duas etapas: a primeira, detalhada no Capítulo 4, consiste em encontrar um modelo politópico derivado do modelo qLPV equivalente a um modelo TS; a segunda, em gerar o controlador para o modelo TS.

A segunda etapa é baseada no conceito de projeto de *compensação distribuída paralela* (PDC). A ideia principal deste conceito é encontrar um controlador na mesma estrutura politópica do modelo. Esta estrutura politópica combina os ganhos de realimentação LIT derivados dos sistemas LIT do modelo politópico. Os ganhos de realimentação podem ser derivados por várias técnicas. Uma das técnicas de projeto mais poderosas, usada neste trabalho, que é capaz de otimizar várias especificações de controle desejadas, é baseada em desigualdades matriciais lineares (LMIs). Portanto, a lei de controle PDC é normalmente aplicada com LMIs. O projeto LMI desenvolvido sob a estrutura PDC requer que as funções de ponderação definam a combinação convexa, a saber o fecho convexo, dos sistemas LIT.

6.1 LEI DE CONTROLE

O controlador PDC foi inicialmente proposto por (WANG; TANAKA; GRIFFIN, 1995) para controle de sistemas com modelos TS. Sua lei de controle, denominada lei de controle PDC, oferece um procedimento para projetar um controlador nebuloso a partir de um modelo nebuloso TS fornecido. No projeto PDC, cada regra de controle é projetada a partir da regra correspondente de um modelo nebuloso TS. O controlador nebuloso projetado compartilha os mesmos conjuntos nebuloso com o modelo nas partes da premissa (TANAKA; WANG, 2001).

Para o modelo nebuloso (3.6), construímos o seguinte controlador nebuloso PDC

$$R_i : \text{Se } p_1(t) \in M_1^{k_1=\alpha_{i1}} \text{ e } p_2(t) \in M_2^{k_2=\alpha_{i2}} \text{ e } \dots \text{ e } p_n(t) \in M_n^{k_n=\alpha_{in}}, \quad (6.1)$$

$$\text{então } \mathbf{u}_i(t) = \mathbf{K}_i \mathbf{x}(t),$$

em que $i = 1, \dots, r$, \mathbf{K}_i são os ganhos do controlador, e as funções de pertinência do antecedente, $M_n^{i_n}$, e as variáveis nebulosas, p_n , são as mesmas do modelo TS (4.16) obtido a partir da transformação do produto tensorial.

A saída do controlador PDC, para r regras, é determinada pela somatória (NIEMANN et al., 1999)

$$\mathbf{u} = \sum_{j=1}^r h_j(\mathbf{x}(t)) \mathbf{K}_j \mathbf{x}(t). \quad (6.2)$$

A lei de controle na forma (6.2) é chamada de *lei de controle PDC*. A lei de controle PDC pode ser vista como um ganho escalonado, em que r controladores por realimentação de estados lineares tem sua contribuição ponderada por funções de pertinência, de forma que a saída seja uma soma convexa das contribuições individuais de cada um. A Figura 10 ilustra um caso de controle com dois parâmetros de dependência (p_1 e p_2), entradas do sistema nebuloso; com nove controladores por realimentação de estados K_i , $i = 1 \dots, 9$; e com três funções de pertinência trapezoidais associadas a cada um dos parâmetros: M_1^2 , M_1^2 e M_1^3 associadas ao parâmetro p_1 , e M_2^1 , M_2^2 e M_2^3 associadas ao parâmetro p_2 . Dois pontos de operações são mostrados, O_1 e O_2 . Nota-se que em O_1 , apenas o controlador por realimentação de estados K_1 está contribuindo para a saída de controle \mathbf{u} , enquanto em O_2 , a saída é uma soma de contribuições, ponderadas pelos valores sobre as funções de pertinência que estão ativas (M_2^2 , M_2^3 , M_1^1 e M_1^2), dos controladores K_1 , K_2 , K_4 e K_5 . Qualquer ponto de operação pode ser representado por uma soma convexa dos controladores por realimentação de estados, ou seja, os esses controladores são vértices de um politopo.

O projeto do PDC se resume a determinar os ganhos de realimentação locais nas partes consequentes. Assim, o PDC é simples e natural. Outras técnicas de controle não-linear requerem conhecimento especial e bastante complexo (TANAKA; IKEDA; WANG, 1998). Outra forma de visualizar a lei de controle PDC é partir do modelo TP de elementos finitos convexo

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \mathbf{y}(t) \end{bmatrix} = \mathcal{S}^{CO} \otimes_{n=1}^N \mathbf{w}_n^{CO}(p_n(t)) \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix}. \quad (6.3)$$

Lembrando que o modelo (6.3) é equivalente a um modelo TS, o que o controlador PDC fará é determinar, via LMIs (no caso deste trabalho) um ganho LIT de realimentação para cada componente LIT do modelo TP:

$$\mathcal{S}^{CO} \longrightarrow \mathcal{F}.$$

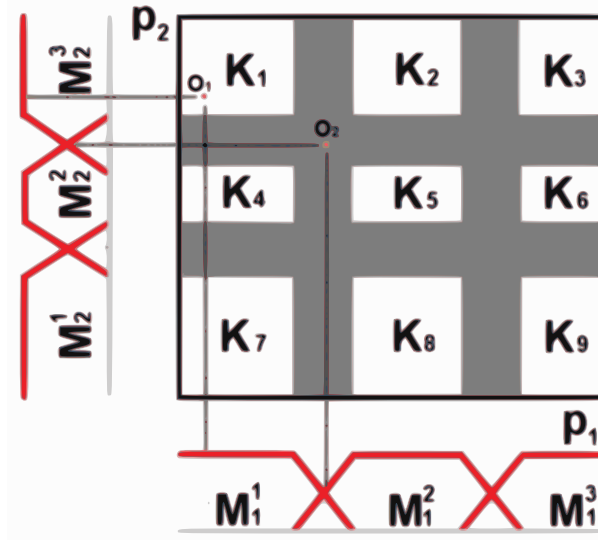


Figura 10 – Representação de pontos de operação de um controlador PDC.

A saída de controle pode, então, ser computada como

$$\mathbf{u}(t) = (\mathcal{F} \otimes_{n=1}^N \mathbf{w}_n^{CO}(p_n(t))) \mathbf{x}(t). \quad (6.4)$$

Um requisito de projeto imposto nesse trabalho é que o controlador tenha taxa de decaimento exponencial, requisito ligado à definição de estabilidade exponencial (Definição 5.6). A seguir, será apresentado um conjunto de LMIs que satisfaz esse requisito.

6.2 CONJUNTO DE LMIS PARA ESTABILIDADE ASSINTÓTICA

Desconsiderando entradas de perturbação e alimentação direta, o sistema dinâmico não linear representado por um modelo TS dado em (3.3) pode ser reescrito como

$$R_i : \text{Se } p_1 \in M_1^{k_1=\alpha_{i1}} \text{ e } p_2 \in M_2^{k_2=\alpha_{i2}} \text{ e } \dots \text{ e } p_n \in M_n^{k_n=\alpha_{in}},$$

$$\text{então } \begin{cases} \dot{\mathbf{x}} = \mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i \mathbf{u}(t) \\ \mathbf{y} = \mathbf{C}_i \mathbf{x}(t) \end{cases}, \quad (6.5)$$

em que $\mathbf{x} \in \mathbb{R}^m$ é o vetor de estados; $\mathbf{u} \in \mathbb{R}^l$ é o vetor de entradas de controle; $\mathbf{y} \in \mathbb{R}^k$ é o vetor de saídas controladas; as matrizes \mathbf{A}_i , \mathbf{B}_i e \mathbf{C}_i são as matrizes que descrevem o comportamento do sistema e são de dimensões adequadas; R_i indica o número da regra; p_1 a p_n são as variáveis premissas do modelo; $M_j^{k_j}$ são os conjuntos nebulosos relacionados à variável p_j ; e α_{ij} relaciona qual conjunto fuzzy da variável p_j está associado à regra i .

Sendo r o número de regras do modelo (6.5), ele pode ser reescrito como

$$\begin{aligned} \dot{\mathbf{x}} &= \sum_{i=1}^r h_i(\mathbf{p}) [\mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i \mathbf{u}(t)], \\ \mathbf{y} &= \sum_{i=1}^r h_i(\mathbf{p}) \mathbf{C}_i \mathbf{x}(t). \end{aligned} \quad (6.6)$$

É importante observar que (6.6) é equivalente a (3.6) para o cenário em que desconsideram-se as entradas de perturbação e alimentação direta. Para obtermos o sistema em malha fechada, substituímos a lei de controle PDC (6.2) na dinâmica do sistema (6.5):

$$\begin{aligned}\dot{\mathbf{x}} &= \sum_{i=1}^r h_i(\mathbf{p}) \left(\mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i \sum_{j=1}^r h_j(\mathbf{p}) \mathbf{K}_j \mathbf{x}(t) \right) \implies \\ \dot{\mathbf{x}} &= \sum_{i=1}^r \sum_{j=1}^r h_i(\mathbf{p}) h_j(\mathbf{p}) (\mathbf{A}_i + \mathbf{B}_i \mathbf{K}_j) \mathbf{x}(t).\end{aligned}\quad (6.7)$$

A partir desse ponto, para não carregar as notações, serão omitidas as dependências das funções.

Teorema 6.1 (Controlador com taxa de decaimento exponencial). (*TANAKA; WANG, 2001*) Dada a lei de controle PDC (6.4), equivalente a (6.2), o maior limitante inferior na taxa de decaimento pela função quadrática de Lyapunov é garantido pela solução do seguinte problema de minimização de autovalor generalizado em \mathbf{X} e γ

$$\begin{aligned}\max_{\mathbf{X}, \mathbf{Y}, \mathbf{M}_1, \dots, \mathbf{M}_r} \quad & \gamma \\ \text{Sujeito a } & \mathbf{X} > 0, \\ & \mathbf{X} \mathbf{A}_i^T + \mathbf{A}_i \mathbf{X} + \mathbf{M}_i^T \mathbf{B}_i + \mathbf{B}_i \mathbf{M}_i + 2\gamma \mathbf{X}_i < 0, \quad \forall i \\ & \mathbf{X} \mathbf{A}_i^T + \mathbf{A}_i \mathbf{X} + \mathbf{M}_i^T \mathbf{B}_i^T + \mathbf{B}_i \mathbf{M}_i + \mathbf{X} \mathbf{A}_j^T + \mathbf{A}_j \mathbf{X} + \mathbf{M}_i^T \mathbf{B}_j^T + \\ & \quad \mathbf{B}_j \mathbf{M}_i + 4\gamma \mathbf{X} < 0, \quad \forall i, j > i\end{aligned}\quad (6.8)$$

para $i < j \leq r$, exceto para o par (i, j) , em que $w_i(\mathbf{p})w_j(\mathbf{p}) = 0, \forall \mathbf{p}$, os ganhos de realimentação são determinados por $\mathbf{K}_i = \mathbf{M}_i \mathbf{X}^{-1}$, r é o número de regras e $\gamma > 0$.

Demonstração. Considere uma função de Lyapunov quadrática nos estados

$$V(\mathbf{x}) = \mathbf{x}^T \mathbf{P} \mathbf{x} > 0 \implies \mathbf{P} > 0, \quad (6.9)$$

em que $\mathbf{P} = \mathbf{P}^T$. A derivada dessa função deve ser tal que a taxa de decaimento seja exponencial. Como a derivada temporal de uma função exponencial é múltipla da própria função ($d(e^{-\gamma t})/dt = -\gamma e^{-\gamma t}$), a derivada da função de Lyapunov deve ter um termo que seja múltiplo dela mesma. Além disso, a exponencial deve ser convergente para que o decaimento seja limitado, justificando o sinal negativo que aparece em γ . Sendo assim, considere a seguinte derivada da função de Lyapunov:

$$\begin{aligned}\dot{V}(\mathbf{x}) &= \dot{\mathbf{x}}^T \mathbf{P} \mathbf{x} + \mathbf{x}^T \mathbf{P} \dot{\mathbf{x}} < -2\gamma V(\mathbf{x}) \implies \\ & \dot{\mathbf{x}}^T \mathbf{P} \mathbf{x} + \mathbf{x}^T \mathbf{P} \dot{\mathbf{x}} + 2\gamma V(\mathbf{x}) < 0 \implies \\ & \dot{\mathbf{x}}^T \mathbf{P} \mathbf{x} + \mathbf{x}^T \mathbf{P} \dot{\mathbf{x}} + 2\gamma (\mathbf{x}^T \mathbf{P} \mathbf{x}) < 0.\end{aligned}\quad (6.10)$$

Substituindo o valor de $\dot{\mathbf{x}}$ em malha fechada, dado em (6.7), em (6.10)

$$\sum_{i=1}^r \sum_{j=1}^r h_i h_j \mathbf{x}^T (\mathbf{A}_i \mathbf{P} + \mathbf{P} \mathbf{A}_i + \mathbf{K}_j^T \mathbf{B}_i^T \mathbf{P} + \mathbf{P} \mathbf{B}_i \mathbf{K}_j + 2\gamma \mathbf{P}) \mathbf{x} < 0. \quad (6.11)$$

Uma condição suficiente para garantir a validade da desigualdade (6.11) é

$$\sum_{i=1}^r \sum_{j=1}^r (\mathbf{A}_i \mathbf{P} + \mathbf{P} \mathbf{A}_i + \mathbf{K}_j^T \mathbf{B}_i^T \mathbf{P} + \mathbf{P} \mathbf{B}_i \mathbf{K}_j + 2\gamma \mathbf{P}) < 0.$$

O problema é que a desigualdade acima não é uma LMI, já que temos multiplicações entre as variáveis matriciais \mathbf{K} e \mathbf{P} , o que não configura linearidade. Usando uma mudança de variável no vetor da forma quadrática $\mathbf{x} = \mathbf{P}^{-1} \bar{\mathbf{x}}$ e duas modificações linearizantes: $\mathbf{X} = \mathbf{P}^{-1}$ e $\mathbf{M}_j = \mathbf{K}_j \mathbf{P}^{-1}$, podemos reescrever (6.11) como

$$\sum_{i=1}^r \sum_{j=1}^r h_i h_j \bar{\mathbf{x}}^T (\mathbf{X} \mathbf{A}_i^T + \mathbf{A}_i \mathbf{X} + \mathbf{M}_j^T \mathbf{B}_i + \mathbf{B}_i \mathbf{M}_j + 2\gamma \mathbf{X}) \bar{\mathbf{x}} < 0. \quad (6.12)$$

Uma condição LMI suficiente é

$$\sum_{i=1}^r \sum_{j=1}^r (\mathbf{X} \mathbf{A}_i^T + \mathbf{A}_i \mathbf{X} + \mathbf{M}_j^T \mathbf{B}_i + \mathbf{B}_i \mathbf{M}_j + 2\gamma \mathbf{X}) < 0.$$

Fazendo $\mathbf{G}_{ij} = (\mathbf{X} \mathbf{A}_i^T + \mathbf{A}_i \mathbf{X} + \mathbf{M}_j^T \mathbf{B}_i + \mathbf{B}_i \mathbf{M}_j + 2\gamma \mathbf{X})$, podemos reescrever (6.13) como

$$\sum_{i=1}^r \sum_{j=1}^r \mathbf{G}_{ij} < 0. \quad (6.13)$$

Usando o teorema que trata de uma condição suficiente para garantir estabilidade em somatórios nebulosos, mostrado e demonstrado no [Anexo A](#):

$$\begin{aligned} \mathbf{G}_{ii} &< 0, \\ \mathbf{G}_{ij} + \mathbf{G}_{ji} &< 0, \quad j > i \end{aligned}$$

chegamos às condições LMIs mostradas em (6.8).

□

O conjunto de LMIs (6.8) garante estabilidade com decaimento exponencial. Porém, o controlador proposto nesse trabalho deve conseguir lidar com o problema de rastreamento de trajetória. A próxima seção, trata da abordagem seguida neste trabalho para satisfazer essa condição de projeto.

6.3 RASTREAMENTO DE TRAJETÓRIA

Para o problema de rastreamento de trajetória, usou-se a abordagem em que o sistema é aumentado com um integrador. Para tal fim, considere um erro integral dado por

$$e_i(t) = \int_0^t y(\beta) - r(\beta) d\beta \quad (6.14)$$

em que $e_i(t)$ é o erro integral, $y(t)$ é o sinal da saída controlada do sistema em malha fechada e $r(t)$ é a referência desejada. Dessa forma, um sistema com matrizes aumentadas pode ser definido por

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{e}}_i(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0 \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{e}_i(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{D} \end{bmatrix} \mathbf{u}(t). \quad (6.15)$$

O controlador desejado neste trabalho é obtido usando-se as matrizes aumentadas (6.15) no conjunto de LMIs (6.8).

CONTROLADOR NEBULOSO ADAPTATIVO DIRETO

Resumo do capítulo

O controle adaptativo foi criado para realizar o controle em sistemas que apresentam variações ou incertezas paramétricas. Neste trabalho, são desenvolvidos controladores nebulosos adaptativos diretos capazes de resolver problemas de rastreamento de trajetória. Neles, sistemas nebulosos são empregados para aproximar um controlador ideal desconhecido. Os parâmetros ajustáveis dos sistemas nebulosos usados são atualizados usando um algoritmo de gradiente descendente, projetado para minimizar o erro entre o controlador ideal desconhecido e o controlador nebuloso. A análise de estabilidade do sistema em malha fechada é realizada usando uma abordagem de Lyapunov. Em particular, é mostrado que os erros de rastreamento são limitados e convergem para uma vizinhança da origem.

O controle adaptativo foi criado para realizar o controle em sistemas que apresentam variações ou incertezas paramétricas. Nele um controlador pode modificar seu comportamento de acordo com o sistema controlado com parâmetros que variam ou são inicialmente desconhecidos (JACOME, 2013).

Conforme mencionado na literatura, existem duas abordagens nebulosas adaptativas: a abordagem direta e a abordagem indireta. Dentro da abordagem adaptativa direta, a função do controlador ideal é estimada diretamente por um sistema de inferência nebulosa como um aproximador universal. Na indireta, a princípio, a função do sistema dinâmico é aproximada por alguns modelos nebulosos e então o controlador é calculado com base no sistema estimado. A principal vantagem da abordagem direta é que ela evita possíveis singularidades que podem ocorrer na abordagem indireta. Por outro lado, no adaptativo direto é necessário impor um maior número de restrições à planta. Em ambos os casos, a estabilidade robusta na presença de incertezas é uma condição necessária para o sistema em malha fechada (KALAB, 2019).

Neste trabalho são desenvolvidos controladores nebulosos adaptativos diretos capazes de resolver problemas de rastreamento de trajetória. Esses controladores usam o projeto de controle proposto por (LABIOD; GUERRA, 2007). Nele a lei adaptativa dos parâmetros nebulosos é projetada, baseada no método do gradiente descendente, para minimizar diretamente o erro entre o controlador ideal desconhecido e o controlador nebuloso. A estabilidade geral do sistema em malha fechada é estudada usando uma abordagem de Lyapunov. Uma nova abordagem usando diferenciador robusto e exato é proposta para estimar a derivada de um parâmetro da lei de adaptação que não pode ser obtida de forma analítica.

7.1 CONSIDERAÇÕES INICIAIS

O controlador adaptativo proposto por (LABIOD; GUERRA, 2007) foi projetado para o controle de sistemas multivariáveis (MIMO) não lineares. Esses pesquisadores consideram sistemas MIMO não lineares aqueles cuja dinâmica pode ser descrita por equações diferenciais na forma

$$\begin{aligned} y_1^{(r_1)} &= f_1(\mathbf{x}) + \sum_{j=1}^p g_{1j}(\mathbf{x})u_j \\ &\vdots \\ y_p^{(r_p)} &= f_p(\mathbf{x}) + \sum_{j=1}^p g_{pj}(\mathbf{x})u_j, \end{aligned} \quad (7.1)$$

em que $\mathbf{x} = [y_1, \dot{y}_1, \dots, y_1^{(r_1-1)}, \dots, y_p, \dot{y}_p, \dots, y_p^{(r_p-1)}]^T \in \mathcal{R}^n$, com $n = \sum_{i=1}^p r_i$, é o vetor de estado geral que é considerado disponível para medição, $\mathbf{u}(t) = [u_1(t), \dots, u_p(t)]^T \in \mathcal{R}^p$ é o vetor de entrada de controle, $\mathbf{y}(t) = [y_1(t), \dots, y_p(t)]^T \in \mathcal{R}^p$ é o vetor de saída, e $f_i(\mathbf{x})$ e $g_{ij}(\mathbf{x})$, $i, j = 1, \dots, p$, são funções não lineares desconhecidas. Reescrevendo (7.1) na forma matricial:

$$\mathbf{y}^{(r)}(t) = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}(t). \quad (7.2)$$

Observe que \mathbf{r} armazena a informação sobre o grau das equações diferenciais. Para o desenvolvimento do controlador, assumem-se as seguintes premissas:

Premissa 1. $\mathbf{G}(\mathbf{x})$ é simétrica definida positiva e limitada por $0 < \delta_0 \mathbf{I} \leq \mathbf{G}(\mathbf{x}) \leq \delta_1 \mathbf{I}_p$, em que $\mathbf{I}_p \in \mathbb{R}^{p \times p}$ é a matriz identidade, δ_0 e δ_1 são constantes positivas;

Premissa 2. As trajetórias desejadas $y_{d_i}(t)$, $i = 1, \dots, p$, são funções limitadas conhecidas com derivadas também limitadas e conhecidas, e assume-se que y_{d_i} é r_i vezes diferenciável.

Embora a restrição advinda da Premissa 1 restrinja a classe considerada de sistemas não lineares MIMO, muitos sistemas físicos, como os sistemas robóticos, cumprem essa propriedade (LABIOD; GUERRA, 2007). Além disso, mais a frente será mostrado que essa premissa cumpre o papel de facilitar o desenvolvimento das equações, mas que esse controlador pode ser usados para sistemas que não a cumprem.

O objetivo é desenvolver um controlador nebuloso adaptativo direto que faça com que a saída $\mathbf{y}(t)$ do sistema controlado em malha fechada siga uma trajetória desejada $\mathbf{y}_d(t) = [y_{d_1}(t), \dots, y_{d_p}(t)]^T$. Para isso, assume-se que cada componente do vetor de controle ideal $\mathbf{u}^*(t) = [u_1^*(t), \dots, u_p^*(t)]$ possa ser aproximado por meio de um sistema nebuloso (FS).

O controlador proposto por (LABIOD; GUERRA, 2007) usa a metodologia de controle não linear por modos deslizantes. Essa metodologia consiste em projetar uma lei de controle capaz de fazer com que todas as trajetórias desse sistema converjam para uma superfície definida no espaço de estados, chamada de superfície deslizante $S(t)$. A dinâmica desta superfície deve ser escolhida pelo projetista de modo que todas as trajetórias dentro da superfície $S(t)$ converjam

para os valores desejados (*set-points*). Após a trajetória atingir o interior da superfície deslizante, é dito que o sistema está operando em modo deslizante. Quando o sistema está em modo de deslizamento, é insensível à variações paramétricas e perturbações externas. Essa propriedade garante robustez a essa metodologia (AGOSTINHO, 2009).

A superfície deslizante, chamada por (LABIOD; GUERRA, 2007) de erro filtrado, é definida por:

$$\begin{aligned} s_1(t) &= \left(\frac{d}{dt} + \lambda_1\right)^{r_1-1} e_1(t), \lambda_1 > 0 \\ &\vdots \\ s_p(t) &= \left(\frac{d}{dt} + \lambda_p\right)^{r_p-1} e_p(t), \lambda_p > 0 \end{aligned} \quad (7.3)$$

em que $s_i(t) = 0$ representa a equação diferencial linear cuja solução implica que o erro de rastreamento $e_i(t) = y_{d_i}(t) - y_i(t)$ converge para zero com uma constante de tempo $(r_i - 1)/\lambda_i$. Além disso, a derivada $r_i - 1$ de $e_i(t)$ sempre converge para zero. Assim, o objetivo do controle passa a ser o projeto de um controlador para manter $s_i(t)$ em zero, $i = 1, \dots, p$ e, portanto, o problema de estabilização original do vetor r_i -dimensional $[e_i, \dots, e_i^{(r_i-1)}]^T$ é reduzido ao de manter o escalar $s_i(t)$ em zero. Ademais, os limites em $s_i(t)$ podem ser traduzidos diretamente em limites no erro de rastreamento. Especificamente, se tivermos $|s_i(t)| \leq \Phi_i$, em que Φ_i é uma constante positiva, podemos concluir que: $|e_i^j(t)| \leq 2^j \lambda_i^{(j-r_i+1)} \leq \Phi_i$, $j = 0, \dots, i_i - 1$, $i = 1, \dots, p$. Esses limites podem ser reduzidos aumentando os parâmetros λ_i (LABIOD; GUERRA, 2007).

Para simplificar a notação, as dependências temporais serão omitidas.

O método de inferência Takagi-Sugeno (TS) é usado para estimar \mathbf{u}^* . Anteriormente, esse método foi usado para descrever modelos TS (3.3). Aqui, ele será descrito de uma maneira generalizada.

O método de inferência TS mapeia cada uma das m entradas de um vetor $\mathbf{z} = [z_1, \dots, z_m]^T \in \Omega_{\mathbf{z}} \subset \mathbb{R}^m$, entrada do sistema nebuloso, em um escalar de saída y_f , em que $\Omega_{\mathbf{z}} = \Omega_{z_1} \times \dots \times \Omega_{z_m} \subset \mathbb{R}$ são os universos de discurso de cada variável z_i . Sendo M_i^k , $i = 1, \dots, m$, conjuntos nebulosos, temos, para N regras (R^k)

$$\begin{aligned} R^k : \text{ Se } z_1 \text{ é } M_1^k \text{ e } \dots \text{ e } z_m \text{ é } M_m^k, \\ \text{ então } y_f \text{ é } y_f^k (k = 1, \dots, N). \end{aligned} \quad (7.4)$$

A saída final do sistema, usando a fuzzificação singleton e o operador produto como operador de intersecção, é dada por

$$y_f(\mathbf{z}) = \frac{\sum_{k=1}^N \mu_k(\mathbf{z}) y_f^k}{\sum_{k=1}^N \mu_k(\mathbf{z})}. \quad (7.5)$$

Com

$$\mu_k(\mathbf{z}) = \prod_{i=1}^m \mu_{M_i^k}(z_i). \quad (7.6)$$

Sendo

$$h_k(\mathbf{z}) = \frac{\mu_k(\mathbf{z})}{\sum_{j=1}^N \mu_j(\mathbf{z})}, \quad k = 1, \dots, N \quad (7.7)$$

a saída pode ser reescrita de forma compacta

$$y_f(\mathbf{z}) = \mathbf{h}^T(\mathbf{z})\Theta, \quad (7.8)$$

em que $\Theta = [y_f^1, \dots, y_f^N]^T$ e $\mathbf{h}(\mathbf{z}) = [h_1(\mathbf{z}), \dots, h_N(\mathbf{z})]^T$.

7.2 LEI DE CONTROLE

Para o desenvolvimento da lei de controle é assumido, conforme visto anteriormente, que a lei de controle ideal \mathbf{u}^* possa ser aproximada por um sistema nebuloso na forma (7.8)

$$u_i^*(\mathbf{z}) = \mathbf{h}_i^T(\mathbf{z})\Theta_i^* + \epsilon_i(\mathbf{z}), \quad i = 1, \dots, p \quad (7.9)$$

em que $\mathbf{z} = [\mathbf{x}^T, \mathbf{s}^T]^T$ (as entradas do sistema nebuloso são estados e derivadas dos erros), $\epsilon_i(\mathbf{z})$ é o erro de aproximação, Θ_i^* o vetor de parâmetros ideal que minimiza a função $|\epsilon_i(\mathbf{z})|$ sobre o conjunto compacto Ω_z , e $\mathbf{h}_i(\mathbf{z})$ é um vetor de função nebulosa adequadamente especificado pelo projetista. Assume-se que o sistema nebuloso usado não viola a propriedade de aproximação universal no conjunto compacto Ω_z , que é considerado grande o suficiente para que a variável \mathbf{z} permaneça dentro dele sob controle em malha fechada. Portanto, é razoável supor que o erro de aproximação nebuloso é limitado para todo $\mathbf{z} \in \Omega_z$.

Escrito de forma matricial

$$\mathbf{u}^* = \mathbf{h}^T(\mathbf{z})\Theta^* + \boldsymbol{\epsilon}(\mathbf{z}). \quad (7.10)$$

Como o vetor de parâmetro ideal Θ^* é desconhecido, ele deve ser estimado por uma lei de adaptação adequada. Seja uma estimativa do vetor ideal e definida a lei de controle como a aproximação nebulosa adaptativa do controlador ideal (7.9), temos a seguinte lei de controle:

$$\mathbf{u}(\mathbf{z}) = \mathbf{h}^T(\mathbf{z})\Theta. \quad (7.11)$$

Definida a estrutura da lei de controle, é necessário projetar a lei de adaptação. A lei de adaptação proposta usa o método do gradiente.

7.3 LEI DE ADAPTAÇÃO

A lei de adaptação deve ser tal que o erro entre \mathbf{u}^* e \mathbf{u} seja o menor possível. Ou seja, a lei de adaptação será encontrada de um problema de otimização cuja função objetivo estará relacionada com a diferença entre \mathbf{u}^* e \mathbf{u} , e o método usado para solucionar esse problema de otimização será o do gradiente. Esse erro é definido por

$$\mathbf{e}_u = \mathbf{u}^* - \mathbf{u} = \mathbf{u}^* - \mathbf{h}^T(\mathbf{z})\Theta = \mathbf{h}^T(\mathbf{z})\tilde{\Theta} + \epsilon(\mathbf{z}), \quad (7.12)$$

em que $\tilde{\Theta} = \Theta^* - \Theta$. Considerando que a lei de adaptação inferida é igual à lei de adaptação ideal $\mathbf{e}_u = 0 \implies \mathbf{u}^* = \mathbf{u}$ (LABIOD; GUERRA, 2007)

$$\dot{\mathbf{s}} = -\mathbf{K}\mathbf{s} - \mathbf{K}_0 \tanh\left(\frac{\mathbf{s}}{\epsilon_0}\right), \quad (7.13)$$

em que $\mathbf{K} = \text{diag}[k_1, \dots, k_n]$, $\mathbf{K}_0 = \text{diag}[k_{0_1}, \dots, k_{0_n}]$, com $K_i > 0$ e $K_{0i} > 0$, para $i = 1, \dots, n$, ϵ_0 é uma pequena constante positiva e $\tanh(*)$ indica tangente hiperbólica.

A principal desvantagem de controladores baseados em modos deslizantes está na ocorrência do fenômeno denominado *chattering*, que são oscilações de alta frequência na informação de comando do sistema de controle. Entre os efeitos indesejados causados por este fenômeno estão: deterioração do desempenho do controlador; aumento de desgaste em partes mecânicas móveis; perdas por dissipação de calor em circuitos elétricos de potência (GUARACY; PINHEIRO, 2014). O termo ligado à tangente hiperbólica em (7.13) foi usada para mitigar esse efeito através de uma suavização das oscilações.

Considerando, agora, que $\mathbf{e}_u \neq 0$

$$\dot{\mathbf{s}} = -\mathbf{K}\mathbf{s} - \mathbf{K}_0 \tanh\left(\frac{\mathbf{s}}{\epsilon_0}\right) + \mathbf{G}(\mathbf{x})\mathbf{e}_u. \quad (7.14)$$

A função objetivo escolhida para problema de otimização é uma função de custo (deve ser minimizada) quadrática em \mathbf{e}_u , cujas medidas indicam a discrepância existente entre o controlador ideal e o controlador nebuloso atual, indicada por

$$J(\Theta) = \frac{1}{2}\mathbf{e}_u^T \mathbf{G}(\mathbf{x})\mathbf{e}_u = \frac{1}{2}(\mathbf{u}^* - \mathbf{h}^T(\mathbf{z})\Theta)^T \mathbf{G}(\mathbf{x})(\mathbf{u}^* - \mathbf{h}^T(\mathbf{z})\Theta) \quad (7.15)$$

O método do gradiente é usado para minimizar (7.15). Aplicando esse método, obtemos como lei adaptativa para os parâmetros Θ a seguinte equação diferencial de primeira ordem (LABIOD; GUERRA, 2007)

$$\dot{\Theta} = -\eta \nabla_{\Theta} J(\Theta), \quad (7.16)$$

em que η é um parâmetro constante positivo.

De (7.15), o gradiente de $J(\Theta)$ em relação a Θ é

$$\frac{\partial J(\Theta)}{\partial \Theta} = -\mathbf{h}(\mathbf{z})\mathbf{G}(\mathbf{x})\mathbf{e}_u. \quad (7.17)$$

Então, o algoritmo do gradiente torna-se

$$\dot{\Theta} = \eta \mathbf{h}(\mathbf{z})\mathbf{G}(\mathbf{x})\mathbf{e}_u. \quad (7.18)$$

Como assumimos \mathbf{u}^* desconhecido, o erro \mathbf{e}_u definido em (7.12) não pode ser encontrado. Isolando-se $\mathbf{G}(\mathbf{x})\mathbf{e}_u$ de (7.14)

$$\mathbf{G}(\mathbf{x})\mathbf{e}_u = \dot{\mathbf{s}} + \mathbf{K}\mathbf{s} + \mathbf{K}_0 \tanh\left(\frac{\mathbf{s}}{\epsilon_0}\right),$$

e substituindo esse valor em (7.18), obtem-se a seguinte lei de adaptação

$$\dot{\Theta} = \eta \mathbf{h}(\mathbf{z}) \left[\dot{\mathbf{s}} + \mathbf{K}\mathbf{s} + \mathbf{K}_0 \tanh\left(\frac{\mathbf{s}}{\epsilon_0}\right) \right]. \quad (7.19)$$

Todo o desenvolvimento até aqui considerou o erro de aproximação $\epsilon = 0$ (7.10). Ou seja, assumiu-se que o modelo nebuloso utilizado é capaz de representar perfeitamente a lei de controle ideal desconhecida, o que só seria verdade com um número infinito de regras. Intuitivamente, acontece um desvio dos parâmetros resultado da tentativa do algoritmo de aprendizagem de ajustar os parâmetros para corresponder a uma função para a qual não existe uma correspondência exata. É bem conhecido da literatura de controle adaptativo que, mesmo na presença de pequenos erros de aproximação, leis adaptativas, como a apresentada em (7.19), podem exibir um fenômeno denominado *deriva dos parâmetros*, no qual os parâmetros desviam indefinidamente de seus valores ótimos e, possivelmente, para o infinito (FARREL; POLYCARPOU, 2006). Quando pequenos erros de rastreamento são atingidos, é possível garantir que o algoritmo mantenha esse erro dentro de uma faixa pequena e limitada. Porém, como o controlador não para de adaptar seus parâmetros, e como a ação adaptativa é integral, não há garantia que $\tilde{\Theta}$ seja limitado. Uma forma simples de evitar esse fenômeno é chamada de *zona morta*, modificação da lei de adaptação que faz com que não seja realizada a adaptação dos parâmetros quando o erro de treinamento é muito pequeno. (LABIOD; GUERRA, 2007) não faz uso deste método. Ao invés disso, eles propõem o uso de uma lei de adaptação robusta obtida pelo uso da técnica modificação- σ , que introduz um termo em (7.19):

$$\dot{\Theta} = \eta \mathbf{h}(\mathbf{z}) \left[\dot{\mathbf{s}} + \mathbf{K}\mathbf{s} + \mathbf{K}_0 \tanh\left(\frac{\mathbf{s}}{\epsilon_0}\right) \right] - \eta \sigma \Theta, \quad (7.20)$$

em que σ é uma pequena constante positiva. O termo $\eta \sigma \Theta$ evita que os valores cresçam indefinidamente. Por exemplo, considere o caso em que Θ tenha atingido altos valores positivos, então esse termo torna-se altamente negativo, forçando os parâmetros a pararem de crescer.

Usando como ponto de partida para análise da convergência dos parâmetros uma função de

Lyapunov quadrática nos erros dos parâmetros

$$V_{\Theta} = \frac{1}{2\eta} \tilde{\Theta}^T \tilde{\Theta}, \quad (7.21)$$

é possível provar que o uso da modificação- σ garante que o vetor do erro dos parâmetros $\tilde{\Theta}$ é limitado (LABIOD; GUERRA, 2007). Após manipulações algébricas, mostra-se que a derivada da função de Lyapunov (7.21) é limitada por

$$\dot{V}_{\Theta} \leq -\frac{3}{4} \mathbf{e}_{\mathbf{u}}^T \mathbf{G}(\mathbf{x}) \mathbf{e}_{\mathbf{u}} - \frac{\sigma}{2} \|\tilde{\Theta}\|^2 + \epsilon^T(\mathbf{z}) \mathbf{G}(\mathbf{x}) \epsilon(\mathbf{z}) + \frac{\sigma}{2} \|\Theta^*\|^2. \quad (7.22)$$

Como os parâmetros que constituem Θ^* são constantes, e assume-se as funções $\epsilon(\mathbf{z})$ e $\mathbf{G}(\mathbf{x})$ limitadas, pode-se definir um limitante constante e positivo

$$\Psi = \sup_t (\epsilon(\mathbf{z})^T \mathbf{G}(\mathbf{x}) \epsilon(\mathbf{z})) + \frac{1}{2} \sigma \|\Theta^*\|^2, \quad (7.23)$$

em que $\alpha = \sigma\eta_0$ e $\eta_0 > 0$. Então, (7.22) pode ser reescrita como

$$\dot{V}_{\Theta} \leq -\alpha V_{\Theta} + \Psi - \frac{3}{4} \mathbf{e}_{\mathbf{u}}^T \mathbf{G}(\mathbf{x}) \mathbf{e}_{\mathbf{u}} \leq \alpha V_{\Theta} + \Psi \quad (7.24)$$

Com isso, é possível provar-se o seguinte teorema:

Teorema 7.1 (Limitante do erro dos parâmetros). (LABIOD; GUERRA, 2007) *Se a Premissa 1 for satisfeita e o erro de aproximação $\epsilon(\mathbf{z})$ em (7.9) for limitado $\|\epsilon(\mathbf{z})\| \leq \bar{\epsilon}$, em que $\bar{\epsilon}$ é uma constante positiva, então a lei de adaptação (7.20) garante*

1. *O vetor de erro dos parâmetros $\tilde{\Theta}$ é limitado e converge para o conjunto residual: $\Omega_{\Theta} = \{\tilde{\Theta} \mid \|\tilde{\Theta}\|^2 \leq 2\eta_0\Psi/\alpha\}$,*
2. *O erro estimado $\|\mathbf{e}_{\mathbf{u}}\|$ é limitado e $(\bar{\epsilon}^2 + \sigma)$ -vezes menor no sentido do erro quadrático médio.*

Demonstração. A Equação (7.24) implica que para $V_{\Theta} \geq \Psi/\alpha$, $\dot{V}_{\Theta} < 0$, e, portanto, V_{Θ} e $\tilde{\Theta}$ são limitados. Juntamente com o fato de que $\epsilon(\mathbf{z}) \in L_{\infty}$ e $\mathbf{h}(\mathbf{z}) \in L_{\infty}$, isso implica que $\mathbf{e}_{\mathbf{u}} \in L_{\infty}$ (7.12).

Ao integrar (7.24), podemos estabelecer que

$$\|\tilde{\Theta}\|^2 \leq \|\tilde{\Theta}(0)\|^2 e^{-\alpha t} + \frac{2\eta_0\Psi}{\alpha}, \quad (7.25)$$

o que implica que $\tilde{\Theta}$ converge para o conjunto residual

$$\Omega_{\Theta} = \left\{ \tilde{\Theta} \mid \|\tilde{\Theta}\|^2 \leq \frac{2\eta_0\Psi}{\alpha} \right\}. \quad (7.26)$$

O segundo item do Teorema 7.1 é provado através da integração de (7.21), usando-se o fato de que $V_{\Theta} \in L_{\infty}$, $0 < \delta_0 \mathbf{I}_p \leq \mathbf{G}(\mathbf{x}) \leq \delta_1 \mathbf{I}_p$ (Preposição 1) e $\|\epsilon(\mathbf{z})\| \leq \bar{\epsilon}$, chegando-se a seguinte integral limitada

$$\int_t^{t+T} \|\mathbf{e}_{\mathbf{u}}\|^2 d\tau \leq c_1(\bar{\epsilon}^2 + \sigma)T + c_2, \quad (7.27)$$

com $c_1 = 4/3(\delta_0) \max(\delta_1, \|\Theta^*\|/2)$ e $c_2 = 4/3(\delta_0) \sup_t (V_{\Theta}(t) - V_{\Theta}(t+T))$. A expressão (7.27) implica que $\|\mathbf{e}_{\mathbf{u}}\|$ é $(\bar{\epsilon}^2 + \sigma)$ -vezes menor no sentido do erro quadrático médio. \square

A derivada temporal do erro filtrado $\dot{\mathbf{s}}$ impõe um problema prático da implementação contínua da lei de adaptação (7.14), pois (7.14) pode ser reescrito da seguinte forma

$$\dot{\mathbf{s}} = \mathbf{K}\mathbf{s} - \mathbf{K}_0 \tanh\left(\frac{\mathbf{s}}{\epsilon_0}\right) + \mathbf{G}(\mathbf{x})(\mathbf{u}^* - \mathbf{u}), \quad (7.28)$$

e como assume-se \mathbf{u}^* e $\mathbf{G}(\mathbf{x})$ desconhecidos, fica evidente que não é possível o cálculo do valor de $\dot{\mathbf{s}}$. (LABIOD; GUERRA, 2007) propõem uma implementação digital do controlador adaptativo, de modo que a derivada de \mathbf{s} , (7.28), possa ser aproximada por uma diferença finita, obtida escolhendo-se um tempo de amostragem T_s pequeno (próximo a zero). Como a maioria das especificações, projetos e análise de comportamento dos processos, sistemas e plantas são feitas em tempo contínuo (MOREIRA; SILVA, 2014), e uma das estratégias para o desenvolvimento de controladores digitais é a aproximação discreta de controladores contínuos (CONCEIÇÃO; BECCARO; JUSTO, 2020), neste trabalho, propomos estimar o valor de $\dot{\mathbf{s}}$ com o uso de um diferenciador robusto e exato, tema do próximo capítulo. Além disso, propomos o uso de algoritmo genético, detalhado no Capítulo 9, para auxiliar na sintonia do controlador, já que ele possui um grande número de parâmetros, o que dificulta esse processo.

7.4 CONVERGÊNCIA DO ERRO DE RASTREAMENTO

A análise da convergência do erro de rastreamento é feita com o uso de funções de Lyapunov. Para isso, considera-se uma função de Lyapunov quadrática no erro filtrado

$$\mathbf{V}_s = \frac{1}{2} \mathbf{s}^T \mathbf{s}. \quad (7.29)$$

Derivando-se (7.29) em relação ao tempo e usando-se (7.14) e (7.12), temos

$$\dot{\mathbf{V}}_s = -\mathbf{s}^T \mathbf{K}\mathbf{s} - \mathbf{s}^T \mathbf{K}_0 \tanh\left(\frac{\mathbf{s}}{\epsilon_0}\right) + \mathbf{s}^T \mathbf{G}(\mathbf{x}) \left(\mathbf{h}^T(\mathbf{z}) \tilde{\Theta} + \epsilon(\mathbf{z})\right). \quad (7.30)$$

De (7.26), obtemos

$$\|\tilde{\Theta}(t)\| \leq \|\tilde{\Theta}(0)\| e^{-0,5\alpha t} + \sqrt{\frac{2\eta_0 \Psi}{\alpha}}. \quad (7.31)$$

Usando (7.31) e o fato de assumirmos $\mathbf{h}(\mathbf{z})$, $\boldsymbol{\epsilon}(\mathbf{z})$ e $\mathbf{G}(\mathbf{x})$ limitados, podemos escrever que

$$\left\| \mathbf{G}(\mathbf{x}) \left(\mathbf{h}^T(\mathbf{z}) \tilde{\boldsymbol{\Theta}} + \boldsymbol{\epsilon}(\mathbf{z}) \right) \right\| \leq \Psi_0 e^{-0,5\alpha t} + \Psi_1, \quad (7.32)$$

em que Ψ_0 e Ψ_1 são constantes positivas finitas.

Usando os resultados obtidos até aqui, e após manipulações algébricas, chega-se à seguinte expressão que mostra que a derivada temporal da função de Lyapunov (7.29) é limitada (LABIOD; GUERRA, 2007)

$$\dot{V}_s \leq \alpha_s V_s + p\Psi_0^2 e^{-\alpha t} + p\Psi_1 \kappa \epsilon_0, \quad (7.33)$$

com $\alpha_s = 2 \min_{1 \leq i \leq p} (K_i - 0,25)$, assumindo cada parâmetro escolhido como $K_i > 0,25$; p quantidade de entradas de controle; e $\kappa = 0,2785$. Nesse cenário, (LABIOD; GUERRA, 2007) enunciam o teorema sobre a convergência do erro de rastreamento para uma vizinhança da origem:

Teorema 7.2 (Convergência do erro de rastreamento). (LABIOD; GUERRA, 2007) *Considerando que as premissas 1 e 2 sejam atendidas e o erro de aproximação nebuloso (7.12) seja limitado, então a lei de controle (7.11) com a lei de adaptação (7.20) garantem que os sinais \mathbf{x} e \mathbf{u} sejam limitados, e limitação assintótica dos erros de rastreamento com: $\|e_i^j(t)\| \leq 2^j \lambda_i^{j-r_i+1} \Phi$, $j = 0, \dots, r_i - 1$, $i = 1, \dots, p$, com $\Phi = \sqrt{2p\Psi_1 \kappa \epsilon_0 / \alpha_s}$.*

Demonstração. A Equação (7.33) implica que para $V_s \leq (p\Psi_0^2 e^{-\alpha t} + p\Psi_1 \kappa \epsilon_0) / \alpha_s$, então $\dot{V}_s \leq 0$. Portanto, o vetor de erro de rastreamento filtrado \mathbf{s} é limitado, o que, junto com a restrição de que a trajetória desejada \mathbf{y}_d e suas derivadas são limitadas, implica que o vetor de estado \mathbf{x} é limitado. Além disso, como o termo $p\Psi_0^2 e^{-\alpha t}$, no lado direito de (7.33), vai para zero quando $t \rightarrow \infty$, podemos concluir que a função V_s será assintoticamente limitada por

$$V_s < \frac{p\Psi_1 \kappa \epsilon_0}{\alpha_s}, \quad (7.34)$$

e, conseqüentemente, o erro filtrado converge assintoticamente para o conjunto residual

$$\Omega_s = \left\{ \mathbf{s} \mid \|\mathbf{s}\| \leq \sqrt{2 \frac{p\Psi_1 \kappa \epsilon_0}{\alpha_s}} \right\}, \quad (7.35)$$

o que implica que s_i é assintoticamente limitado por

$$s_i \leq \left\{ \sqrt{2 \frac{p\Psi_1 \kappa \epsilon_0}{\alpha_s}} \right\}. \quad (7.36)$$

Podemos concluir que os erros de rastreamento e suas derivadas de tempo são assintoticamente limitados:

$$\|e_i^j(t)\| \leq 2^j \lambda_i^{j-r_i+1} \Phi, \quad j = 0, \dots, r_i - 1, \quad i = 1, \dots, p, \quad (7.37)$$

em que $\Phi = \sqrt{2p\Psi_1\kappa\epsilon_0/\alpha_s}$. É possível tornar os limitantes superiores dos erros de rastreamento os menores possíveis escolhendo apropriadamente os parâmetros de projeto ϵ_0 , k_i e λ_i , $i = 1, \dots, p$.

□

Segundo (LABIOD; GUERRA, 2007), a fim de simplificar o projeto de uma lei adaptativa que minimiza o erro entre o controlador ideal e o controlador nebuloso, a matriz de ganho de controle $\mathbf{G}(\mathbf{x})$ é assumida como sendo simétrica definida positiva. Essa suposição permite definir uma função de custo que mede a discrepância entre o controlador ideal e o controlador nebuloso, de forma que a lei de adaptação dos parâmetros resultante possa ser implementada. Por outro lado, ao assumir a matriz $\mathbf{G}(\mathbf{x})$ apenas regular e limitada, a maior dificuldade de projeto do controlador advém do fato de o sinal da função de custo (7.15) ser desconhecido. No entanto, os resultados permanecem válidos para sistemas com $\mathbf{G}(\mathbf{x})$ sendo uma matriz definida positiva (não necessariamente simétrica).

DIFERENCIADOR ROBUSTO E EXATO

Resumo do capítulo

A diferenciação de sinais em tempo real é uma questão que enfrenta problemas de ordem prática. A maioria dos diferenciadores, instrumentos usados para estimar a derivada de sinais, conhecidos é capaz de gerar estimativas próximas da derivada do sinal base, além de rejeitar razoavelmente ruídos de alta frequência. O diferenciador robusto e exato (RED) proposto neste trabalho pode fornecer, na ausência de ruído, a derivada exata. Na presença de ruído, o RED possui acurácia proporcional a raiz quadrada da magnitude do ruído.

A diferenciação de sinais em tempo real é uma questão que enfrenta problemas de ordem prática. A tarefa de gerar uma estimativa da derivada temporal de um sinal base é dificultada quando o sinal de entrada do diferenciador é corrompido por ruído de alta frequência, uma vez que esta classe de ruídos possui derivada com amplitudes elevadas. A maioria dos diferenciadores conhecidos é capaz de gerar estimativas próximas da derivada do sinal base, além de rejeitar razoavelmente ruídos de alta frequência. Para isso, eles utilizam uma largura de banda finita, de modo a filtrar as componentes de alta frequência do sinal de entrada. Deste modo, eles não conseguem fornecer uma resposta exata na ausência de ruído, sendo, portanto, diferenciadores robustos, mas não exatos (VIDAL, 2014).

Este trabalho utiliza o diferenciador robusto e exato baseado em modos deslizantes proposto por (LEVANT, 1998). Seu objetivo é que dado um sinal de entrada mensurável e localmente limitado $f(t)$, a saída u_{RED} do RED deve ser tal que $u_{RED} - \dot{f}(t) = 0$. Ou seja, que sua saída convirja para a derivada temporal do sinal de entrada. As equações que regem esse diferenciador são mostradas abaixo:

$$\dot{x}_{RED} = u_{RED} \quad (8.1)$$

$$\begin{aligned} u_{RED} &= u_{RED_1} - \kappa |x_{RED} - f(t)|^{\frac{1}{2}} \text{sign}(x_{RED} - f(t)) \\ \dot{u}_{RED_1} &= -\alpha \text{sign}(x_{RED} - f(t)), \quad \kappa, \alpha > 0. \end{aligned} \quad (8.2)$$

A condição de convergência desse dispositivo, apresentada no Teorema 8.1, usa o conceito de *constante de Lipschitz*. Essa constante está ligada à definição de funções de Lipschitz, uma importante classe de funções no campo de estudo da continuidade de funções.

Teorema 8.1 (Condição de convergência). *Seja $f(t)$ um sinal cuja derivada possua constante de Lipschitz $C > 0$, a observância das seguintes expressões é condição suficiente para a convergência de $u_{RED}(t)$ para $\dot{f}(t)$ (LEVANT, 1998):*

$$\alpha > C, \quad \kappa^2 \geq 4C \frac{\alpha + C}{\alpha - C} \quad (8.3)$$

Este diferenciador pode fornecer, na ausência de ruído, a derivada exata. Na presença de ruído, o RED possui acurácia proporcional a raiz quadrada da magnitude do ruído.

Teorema 8.2 (Acurácia na presença de ruídos). *Considere um sinal de entrada na forma $f(t) = f_0(t) + \nu(t)$, em que $\nu(t) \leq \delta_\nu$ é um ruído e $f_0(t)$ um sinal diferenciável cuja derivada $\dot{f}_0(t)$ possui uma constante de Lipschitz $C > 0$. Então, existe uma constante b dependente de $(\alpha - C)/\kappa^2$ e $(\alpha + C)/\kappa^2$ que após um tempo finito garante que desigualdade $|u_{RED} - \dot{f}_0(t)| < \kappa b \sqrt{\delta_\nu}$ seja satisfeita (LEVANT, 1998).*

Deve-se destacar que o estado do RED não pode escapar em tempo finito, dado que o sinal de entrada possui derivada segunda limitada (E.NUNES; HSU; LIZARRALDE, 2004).

Exemplo 3: Aplicação do RED

A Figura 11 apresenta as derivadas estimadas pelo RED e as derivadas reais de dois sinais na ausência de ruído $f(t) = 5t + \text{sen}(t)$, Figura 11(a), e $f(t) = 5t + \text{sen}(t) + 0,001\text{cos}(30t)$, Figura 11(b); e na presença de ruídos, figuras 11(c) e 11(d), respectivamente. Os parâmetros do RED usados foram $\alpha = 8$ e $\kappa = 6$, e os ruídos inseridos são brancos e com amplitudes de aproximadamente 0,001.

Nota-se que em cenários com a ausência de ruído, mesmo no segundo sinal, em que há uma parcela de alta frequência, esse diferenciador é capaz de estimar de forma exata a derivada do sinal. Na presença de ruído, a acurácia da estimação é proporcional à raiz quadrada da amplitude do ruído $\sqrt{0,001} = 0.0316$.

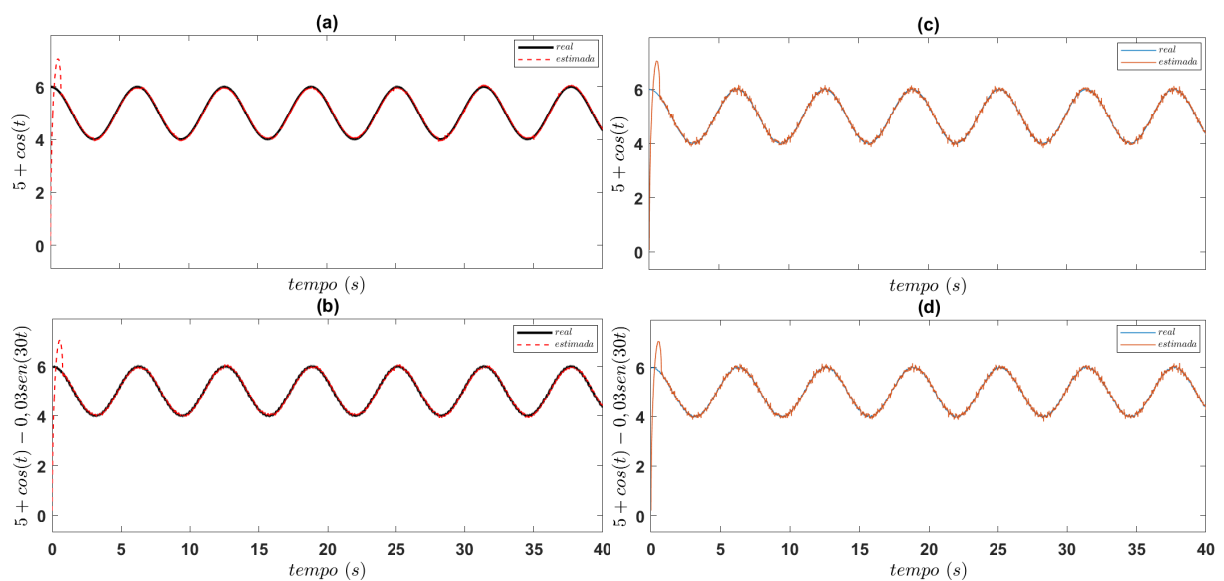


Figura 11 – Derivadas real e estimada pelo RED da função $f(t) = 5t + \text{sen}(t)$ na ausência (a) e na presença de ruído (c); derivadas real e estimada pelo RED da função $f(t) = 5t + \text{sen}(t) + 0,001\text{cos}(30t)$ na ausência (b) e na presença de ruído (d).

ALGORITMO GENÉTICO

Resumo do capítulo

A utilização de controladores nebulosos implica no ajuste de vários parâmetros e, à medida que a complexidade do processo aumenta, torna-se difícil estabelecer a configuração ideal deles. Neste trabalho, propomos o uso de algoritmo genético (AG) para contornar esse problema. Esses algoritmos são uma classe particular de algoritmos evolutivos, algoritmos inspirados na teoria da evolução natural das espécies, que usam técnicas inspiradas pela biologia evolutiva como hereditariedade, mutação, seleção natural e recombinação para encontrar soluções aproximadas em problemas de otimização e busca.

A dificuldade de encontrar algoritmos de controle analítico para plantas complexas contribuiu para a expansão do uso de algoritmos genéticos (AG) e da programação genética na engenharia de controle (HERRERA; LOZANO; VERDEGAY, 1995), e seu uso juntamente com a lógica nebulosa vem crescendo rapidamente no mundo industrial para resolver problemas do mundo real com eficácia (PLEROU, 2016).

A utilização de controladores nebulosos implica no ajuste de vários parâmetros e, à medida que a complexidade do processo aumenta, torna-se difícil estabelecer a configuração ideal deles (ARAUJO et al., 2003). Neste trabalho, propomos o uso de AG para contornar esse problema.

A computação evolutiva é constituída pelos métodos computacionais inspirados na teoria da evolução natural das espécies. Tais métodos são baseados em uma população de indivíduos sujeitos a avaliação, modificações e seleção. As instâncias dos algoritmos fundamentados em princípios evolutivos são chamadas de algoritmos evolutivos (MEDEIROS, 2006). Os algoritmos genéticos, classe particular de algoritmos evolutivos, usam técnicas inspiradas pela biologia evolutiva como hereditariedade, mutação, seleção natural e recombinação para encontrar soluções aproximadas em problemas de otimização e busca. Esses algoritmos são implementados como uma simulação de computador, em que uma população de representações abstratas de solução é selecionada em busca de soluções melhores. A evolução geralmente se inicia a partir de um conjunto de soluções criado aleatoriamente e é realizada por meio de gerações. A cada geração, a adaptação de cada solução na população é avaliada, alguns indivíduos são selecionados para a próxima geração, são recombinados e sofrem mutação para formar uma nova população. A nova população então é utilizada como entrada para a próxima iteração do algoritmo (MORAIS, 2006). Essas etapas podem ser vistas no fluxograma da Figura 12.

O AG desenvolvido neste trabalho, cuja a base está presente no Algoritmo 1, deve encontrar a melhor configuração (sintonia) dos parâmetros que minimize a função objetivo, que, nesse caso, é uma função de custo. A seguir, são apresentados os passos seguidos para desenvolvê-lo.

1. Definição da função objetivo: A função de custo $J(e, u)$ a ser otimizada pelo AG é

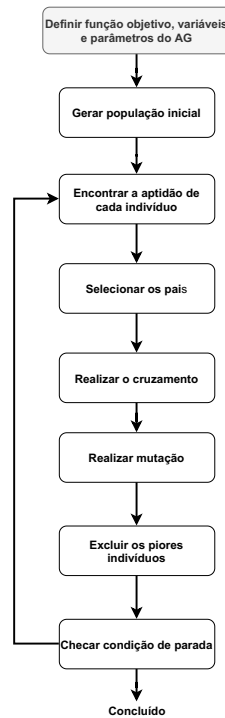


Figura 12 – Fluxograma de um AG.

Fonte : Adaptado de (HAUPT; HAUPT, 2004)

a raiz do erro quadrático médio (RMSE, em inglês) somado à raiz do sinal de controle quadrático médio,

$$J(\mathbf{e}, \mathbf{u}) = \left[\frac{1}{N} (\mathbf{e}^T \mathbf{e}) \right]^{1/2} + \left[\frac{1}{N} (\mathbf{u}^T \mathbf{u}) \right]^{1/2}, \quad (9.1)$$

sendo N o tamanho dos vetores coluna \mathbf{e} e \mathbf{u} . Os valores de \mathbf{e} e \mathbf{u} são obtidos por meio de simulação computacional usando o Simulink®. Como as escalas de \mathbf{e} e \mathbf{u} são diferentes, os valores foram normalizados, evitando que uma grandeza tenha importância maior que outra no processo de otimização.

- 2. Representação dos indivíduos:** Um indivíduo é uma possível solução para o problema a ser otimizado. No caso deste trabalho, sua representação computacional é dada por um vetor de números reais, chamado de cromossomo, em que cada posição deste vetor, chamada de gene, representa um parâmetro do controlador a ser otimizado.

Abaixo encontram-se exemplos de cromossomos para um cenário em que existem dois controladores nebulosos adaptativos diretos a serem sintonizados via AG:

$$cromossomo_1 = [K_1^1 \ K_2^1 \ K_{0_1}^1 \ K_{0_2}^1 \ \lambda_1^1 \ \lambda_2^1 \ \epsilon_{0_1}^1 \ \epsilon_{0_2}^1 \ \sigma_1^1 \ \sigma_2^1 \ \eta_1^1 \ \eta_2^1]$$

$$cromossomo_2 = [K_1^2 \ K_2^2 \ K_{0_1}^2 \ K_{0_2}^2 \ \lambda_1^2 \ \lambda_2^2 \ \epsilon_{0_1}^2 \ \epsilon_{0_2}^2 \ \sigma_1^2 \ \sigma_2^2 \ \eta_1^2 \ \eta_2^2]$$

⋮

$$cromossomo_N = [K_1^N \ K_2^N \ K_{0_1}^N \ K_{0_2}^N \ \lambda_1^N \ \lambda_2^N \ \epsilon_{0_1}^N \ \epsilon_{0_2}^N \ \sigma_1^N \ \sigma_2^N \ \eta_1^N \ \eta_2^N],$$

em que ν_i^j é o gene do cromossomo j que representa o parâmetro ν do controlador i (exemplo: K_1^2 é o parâmetro K do controlador 1 do cromossomo 2), e N é o tamanho da população.

3. **Inicialização da população:** A população é um conjunto de indivíduos. Sua inicialização foi feita de forma aleatória.

Realizaram-se simulações computacionais para definir os extremos dos intervalos em que os valores dos parâmetros não levam o sistema à instabilidade ou dinâmicas não desejadas. Posteriormente, valores aleatórios nesses intervalos foram gerados para compor os cromossomos dos indivíduos da população inicial.

4. **Avaliação da população:** Para cada indivíduo da população é executada uma simulação no Simulink® do sistema usando os valores dos parâmetros do controlador contidos em seu cromossomo. A avaliação usa o erro e o sinal de controle obtido na simulação em (9.1).

5. **Seleção dos pais:** A seleção dos pais que realizarão o cruzamento foi feita por torneio. Nele um número estipulado de indivíduos são sorteados aleatoriamente e os dois melhores indivíduos (menor valor $J(e, u)$) dentre eles são escolhidos para serem os indivíduos pais.

Quanto maior o número de indivíduos selecionados para participarem do torneio, mais elitista torna-se o algoritmo, já que a probabilidade dos melhores indivíduos serem selecionados é maior. Caso essa quantidade seja igual ao tamanho da população, os dois melhores indivíduos da população sempre serão selecionados como pais, o que pode provocar uma convergência precoce do algoritmo.

6. **Cruzamento:** O cruzamento recombina as características das soluções selecionadas (pais) para gerar outros indivíduos (filhos).

O operador de cruzamento usado tem os seguintes passos: um número aleatório (β) entre

0 e 1 é gerado. Os cromossomos dos filhos são gerados pelas equações:

$$\begin{aligned} cromossomo_{filho_1} &= \beta \cdot cromossomo_{pai_1} + \\ &\quad (1 - \beta) \cdot cromossomo_{pai_2}, \\ cromossomo_{filho_2} &= (1 - \beta) \cdot cromossomo_{pai_1} + \\ &\quad \beta \cdot cromossomo_{pai_2}, \end{aligned} \tag{9.2}$$

ou seja, os cromossomos dos filhos são uma combinação linear dos cromossomos dos pais.

7. **Mutação:** A mutação permite que algumas características dos indivíduos resultantes sejam alteradas para gerar mais variedade genética. Com certa porcentagem de acontecimento, cada filho pode sofrer uma alteração em seu cromossomo.

No operador escolhido, um gene aleatório do indivíduo é selecionado e seu valor é substituído por outro dentro do intervalo válido definido para o parâmetro do controlador que aquele gene representa.

8. **Exclusão dos piores indivíduos:** Os indivíduos filhos gerados são inseridos na população, e os N piores indivíduos desta população aumentada são excluídos, de forma que o tamanho da população permaneça sempre constante ao final de cada geração (iteração).
9. **Checagem da condição de parada:** A única condição, a priori, de parada utilizada neste trabalho foi atingir a quantidade máxima de gerações determinada.

Algoritmo 1: PSEUDO-CÓDIGO DO ALGORITMO GENÉTICO UTILIZADO NESTE TRABALHO

Entrada: quantidade de gerações (Q_g), probabilidade de mutação (P_m), tamanho da população (N), tamanho do torneio (T_t), função de custo (J)

Saída: melhor indivíduo encontrado

```
1 início
2   Gerar população inicial aleatoriamente
3   Avaliar indivíduos em relação a  $J$ 
4   repita
5       repita
6           Selecionar  $T_t$  indivíduos da população
7           Pegar os dois melhores dentre os selecionados para serem pais
8       até  $N/2$ ;
9       Realizar cruzamento dos pais
10      para cada filho faça
11          Gerar número aleatório entre 0 e 1
12          se número aleatório  $\leq P_m$  então
13              Realizar mutação
14          fim
15      fim
16      Avaliar filhos em relação a  $J$ 
17      Inserir filhos na população
18      Ordenar a população por aptidão
19      Manter os  $N$  melhores indivíduos
20      se critério de parada foi atingido então
21          Parar
22      fim
23      geração  $\leftarrow$  geração + 1
24  até geração  $\leq Q_g$ ;
25 fim
26 retorna Retornar o melhor indivíduo encontrado
```

Parte II

Projeto

Resumo do capítulo

Dois controladores PDC foram projetados: um obtido da síntese referente a um modelo qLPV sem simplificação, e outro da síntese de um modelo com a simplificação dos pequenos ângulos. As etapas de projeto seguidas foram: definição do modelo qLPV; utilização da transformação do produto tensorial com o uso do Matlab®; montagem das matrizes aumentadas do sistema usando os vértices LIT da etapa anterior; e utilização das matrizes aumentadas em LMIs, resolvidas com o uso do pacote SeDuMi. Duas abordagens de projeto também foram utilizadas para o controlador adaptativo: uma em que são utilizados três controladores, em que um deles é responsável pelo controle das malha de rolagem e arfagem, e outra usando quatro controladores. Utilizou-se o AG para auxiliar na sintonia dos parâmetros dos controladores adaptativos. A análise do desempenho de todos os controladores desenvolvidos foi feita com a utilização do Simulink®.

A análise do desempenho dos controladores desenvolvidos neste trabalho foi feita por meio de simulações. Eles foram implementados e simulados em ambiente computacional com o uso do programa Matlab®. Utilizou-se o Simulink®, ferramenta para modelagem, simulação e análise de sistemas dinâmicos, para simplificar a construção do sistema.

A Tabela 1 mostra os parâmetros do quadrotor usados neste trabalho. Nota-se que, como a modelagem foi realizada assumindo-se simetria da estrutura da aeronave, $I_{xx} = I_{yy}$.

Tabela 1 – Parâmetros do quadrotor.

Símbolo	Unidade	Valor	Descrição
m	kg	1,2	Massa total do quadrotor
l	m	0,31	Distância do centro de massa da hélice ao centro de massa do quadrotor
I_{xx}	$N \cdot m \cdot s^2$	$9,8 \cdot 10^{-3}$	Momento de inércia da aeronave em torno do eixo X
I_{yy}	$N \cdot m \cdot s^2$	$9,8 \cdot 10^{-3}$	Momento de inércia da aeronave em torno do eixo Y
I_{zz}	$N \cdot m \cdot s^2$	$16,3 \cdot 10^{-3}$	Momento de inércia da aeronave em torno do eixo Z

Fonte : Adaptado de (BENIGNO, 2015)

Os sinais de referência $y_d(t)$ apresentam parte senoidal e parte rampas com inclinações elevadas, assemelhando-se a degraus. Degraus não podem ser utilizados, pois não são deriváveis para todo $t \in \mathcal{R}$.

Nos intervalos de tempo $8 s \leq t \leq 12 s$ e $18 s \leq t \leq 22 s$ inseriram-se ruídos de processo e medição, respectivamente, ambos brancos e com amplitude de aproximadamente 0,16.

10.1 CONTROLADOR PDC

Para aplicar a técnica da transformação do produto tensorial é necessário escolher um modelo qLPV que represente matematicamente a aeronave. Essa escolha não é única, e diferentes modelos podem levar a resultados distintos na síntese de controladores. Duas abordagens de projeto foram desenvolvidas: uma usando um modelo sem simplificação; outra, um simplificado.

1. Modelo sem simplificação: obtido colocando-se (2.26) na forma de espaços de estados:

$$\begin{bmatrix} \dot{v}_z \\ \dot{z} \\ \dot{\omega}_\phi \\ \dot{\phi} \\ \dot{\omega}_\theta \\ \dot{\theta} \\ \dot{\omega}_\psi \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{35} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{53} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & A_{73} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_z \\ z \\ \omega_\phi \\ \phi \\ \omega_\theta \\ \theta \\ \omega_\psi \\ \psi \end{bmatrix} + \begin{bmatrix} B_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & B_{32} & B_{33} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & B_{52} & B_{53} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & B_{72} & B_{73} & B_{74} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}, \quad (10.1)$$

em que:

- $A_{35} = -\dot{\psi}c_\phi + \frac{I_{yy}-I_{zz}}{I_{yy}}(\dot{\psi} - \dot{\theta}s_\phi)c_\phi,$
- $A_{53} = \dot{\psi}t_\phi + \frac{\dot{\theta}}{c_\phi} - \frac{I_{yy}-I_{zz}}{I_{yy}}(\dot{\psi} - \dot{\theta}s_\phi)1/c_\phi,$
- $A_{73} = \dot{\psi}t_\phi + \frac{\dot{\theta}}{c_\phi} - \frac{I_{yy}-I_{zz}}{I_{yy}}(\dot{\psi} - \dot{\theta}s_\phi)t_\phi,$
- $B_{11} = c_\theta c_\phi / m,$
- $B_{32} = lc_\psi / I_{yy},$
- $B_{33} = -ls_\psi / I_{yy},$
- $B_{52} = ls_\psi / (I_{yy}c_\phi),$
- $B_{53} = lc_\psi / (I_{yy}c_\phi),$
- $B_{72} = ls_\psi t_\psi / I_{yy},$
- $B_{73} = lc_\psi t_\phi / I_{yy},$
- $B_{74} = 1 / I_{zz}.$

2. Modelo simplificado: obtido colocando-se (2.27), modelo com a simplificação dos pe-

quenos ângulos, na forma de espaço de estados:

$$\begin{bmatrix} \dot{v}_z \\ \dot{z} \\ \dot{\omega}_\phi \\ \dot{\phi} \\ \dot{\omega}_\theta \\ \dot{\theta} \\ \dot{\omega}_\psi \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{35} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{53} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & A_{73} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_z \\ z \\ \omega_\phi \\ \phi \\ \omega_\theta \\ \theta \\ \omega_\psi \\ \psi \end{bmatrix} + \begin{bmatrix} B_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & B_{32} & B_{33} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & B_{52} & B_{53} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & B_{72} & B_{73} & B_{74} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}, \quad (10.2)$$

com:

- $A_{35} = -\dot{\psi} + \frac{I_{yy} - I_{zz}}{I_{yy}}(\dot{\psi} - \dot{\theta}\phi),$
- $A_{53} = \dot{\psi}\phi + \dot{\theta} - \frac{I_{yy} - I_{zz}}{I_{yy}}(\dot{\psi} - \dot{\theta}\dot{\phi}),$
- $A_{73} = \dot{\psi}\phi + \dot{\theta} - \frac{I_{yy} - I_{zz}}{I_{yy}}(\dot{\psi} - \dot{\theta}\phi)\phi,$
- $B_{11} = c_\theta c_\phi / m,$
- $B_{32} = l / I_{yy},$
- $B_{33} = -l\psi / I_{yy},$
- $B_{52} = l\psi / I_{yy},$
- $B_{53} = l / I_{yy},$
- $B_{72} = l\dot{\psi}\psi / I_{yy},$
- $B_{73} = l\phi / I_{yy},$
- $B_{74} = 1 / I_{zz}.$

Uma **importante observação** é que o modelo simplificado foi usado apenas na etapa de aplicação da técnica da transformação do produto tensorial e, conseqüentemente, na síntese do controlador. O modelo da aeronave construído e simulado no Simulink® não possui qualquer tipo de simplificação.

A desvantagem crucial da técnica de transformação produto tensorial é que sua carga computacional aumenta de forma muito rápida com a densidade de discretização e a dimensionalidade do vetor de parâmetro (CUI; ZHANG; MA, 2016). Como a HOSVD está sendo usada como uma ferramenta de identificação de padrões, os dados que não variam ao longo do tensor de amostragem em nada agregam para essa aplicação. Sendo assim, com o intuito de reduzir o uso de memória, usaram-se dados vetoriais, em que $\mathbf{S}(\mathbf{p})$ é formado pelos elementos das matrizes \mathbf{A} e \mathbf{B} que variam no espaço de amostragem (elementos dependentes de parâmetros), em

vez de dados matriciais ($\mathbf{S}(\mathbf{p}) = [\mathbf{A} \ \mathbf{B}]$). Ou seja,

$$\mathbf{S}(\mathbf{p}) = \begin{bmatrix} A_{35} & A_{53} & A_{73} & B_{11} & B_{32} & B_{33} & B_{52} & B_{53} & B_{72} & B_{73} \end{bmatrix}^T,$$

no caso do modelo completo, e

$$\mathbf{S}(\mathbf{p}) = \begin{bmatrix} A_{35} & A_{53} & A_{73} & B_{11} & B_{33} & B_{52} & B_{72} & B_{73} \end{bmatrix}^T,$$

no caso do modelo simplificado.

O uso de dados vetoriais em vez de matriciais faz com que o tensor que armazena as amostras, \mathcal{S}^D , deixe de ter dimensão $I_1 \times I_2 \times \dots \times I_n \times (8 \times 12)$ e passe a ter dimensão $I_1 \times I_2 \times \dots \times I_n \times (10 \times 1)$ no caso completo; e $I_1 \times I_2 \times \dots \times I_n \times (8 \times 1)$, no simplificado.

A densidade do *grid* foi obtida por testes computacionais. Algumas execuções do algoritmo mostravam não ser necessários tantos pontos em uma determinada direção, já que o conjunto de funções de pertinência para ela apresentava apenas uma função de pertinência ou funções de pertinência lineares; outras execuções, que era necessário aumentar a densidade do *grid* em uma determinada direção, já que o conjunto de funções de pertinência para ela não apresentava aparente suavidade em uma ou mais função de pertinência.

Na etapa da HOSVD, foram mantidos em cada direção os valores singulares até mil vezes menores que o maior valor singular, e os tipos de fechos convexos usados foram o NN-SN e RNO-INO. Essas escolhas fazem parte das definições de projeto e são validadas através da análise da dinâmica do sistema controlado pelo controlador sintetizado pelo modelo TS obtido da decomposição do produto tensorial. Caso surja alguma dinâmica não desejada, essas escolhas podem ser mudadas.

Ao sinal de controle U_1 foi somado o termo $mg/(\cos(\theta)\cos(\phi))$, que tem o papel de compensar a força peso, diminuindo o tempo de resposta do controlador.

O esquema do projeto é mostrado na Figura 13. Em resumo:

1. Definiu-se o modelo qLPV do sistema;
2. Com o uso do Matlab®, utilizou-se a técnica da transformação do produto tensorial para encontrar as funções de pertinência e os sistemas LIT (vértices do politopo) que compõem o modelo TS;
3. Os sistemas LIT foram usados para montar as matrizes aumentadas do sistema, (6.15);
4. As matrizes aumentadas foram utilizadas para encontrar os ganhos de realimentação locais do controlador pela resolução, com o uso do SeDuMi, do sistema de LMIs (6.8);
5. Esses ganhos e as funções de pertinência compõem o controlador PDC, implementado no Simulink®, que estima o sinal de controle pelo sistema nebuloso (6.1). Esse sistema usa as funções de pertinência, os ganhos e as variáveis dependência (estados relacionados à atitude) para a inferência.

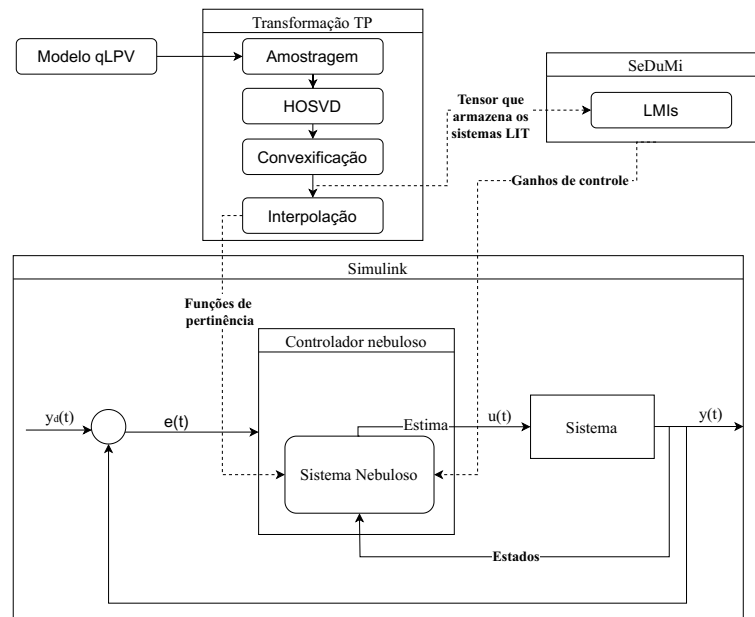


Figura 13 – Esquema da metodologia seguida com os controladores PDC.

10.2 CONTROLADOR NEBULOSO ADAPTATIVO DIRETO

Assim como no caso PDC, duas abordagens foram desenvolvidas para os controladores adaptativos diretos:

1. Primeiramente, foram desenvolvidos três controladores adaptativos diretos: um controlador para a altitude, um controlador para as malhas de rolagem (ϕ) e arfagem (θ), e um controlador para a malha de rolagem (ψ).
2. Na segunda abordagem, desenvolvida com o intuito de tentar reduzir o número de regras e com isso o número de parâmetros estimados, foram desenvolvidos quatro controladores, um para cada malha controlável da aeronave.

A derivada do erro filtrado \dot{s} de cada controlador foi estimada com o derivador robusto e exato (RED). Esse mecanismo também foi implementado no Simulink®, e seus parâmetros κ e α foram sintonizados por tentativa e erro.

Conforme mencionado anteriormente, devido à grande quantidade de parâmetros, uma boa sintonia manual de controladores nebulosos adaptativos torna-se um processo massante. Na tentativa de contornar esse problema, usou-se o AG. Os intervalos utilizados para gerar os valores dos genes dos cromossomos foram encontrados através de teste computacionais, que tinham o intuito de descobrir os valores aproximados dos limitantes inferiores e superiores de cada gene que não levavam o sistema à instabilidade ou a dinâmicas não desejadas.

A Figura 14 resume a metodologia seguida neste trabalho para o projeto dos controladores adaptativos. Esses controladores, que usam o método de inferência Takagi-Sugeno de ordem

zero para estimar o sinal de controle $u(t)$, foram implementados no Simulink®. Seus parâmetros são estimados e otimizados de maneira *off-line* com o uso do AG, cujo código, implementado em Matlab®, recebe os valores dos vetores e e u , necessários para o cálculo da função de custo (7.15), através das simulações no ambiente Simulink®.

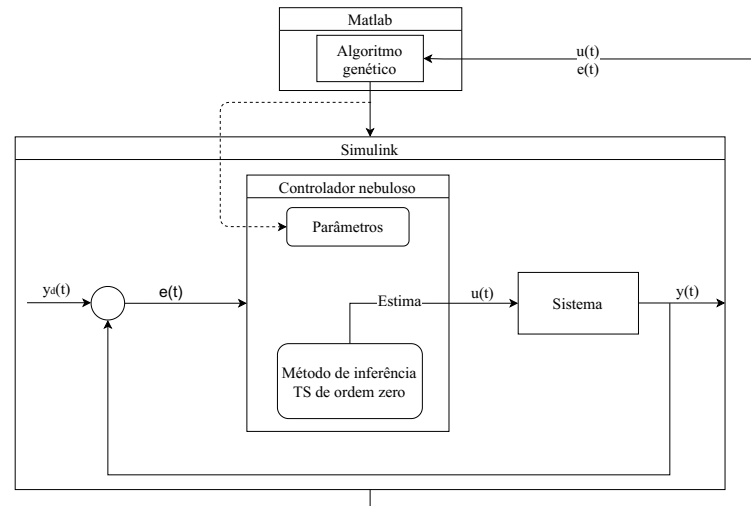


Figura 14 – Esquema da metodologia seguida com os controladores adaptativos.

Resumo do capítulo

O controlador PDC projetado usando o modelo qLPV simplificado apresentou baixos esforços de controle e eficácia no rastreamento de trajetória, apesar de ter demonstrado uma sensibilidade ao ruído de processo. Além disso, ele apresentou 6,5 menos regras que o controlador usando o modelo não simplificado. Não foi possível analisar o desempenho desse último controlador, pois houve problemas numéricos na simulação no Simulink®. O controle adaptativo feito com o uso de quatro controladores se mostrou mais eficiente, apresentando um menor número de regras e esforços de controle, e um melhor rastreamento de trajetória. O algoritmo genético demonstrou uma convergência precoce e mostrou-se ineficiente para otimizar os parâmetros sintonizados manualmente.

11.1 CONTROLADOR PDC

Na primeira abordagem, sem a simplificação do modelo (10.1), temos:

- Conforme descrito anteriormente, para poupar memória:

$$\mathbf{S}(\mathbf{p}) = \begin{bmatrix} A_{35} & A_{53} & A_{73} & B_{11} & B_{32} & B_{33} & B_{52} & B_{53} & B_{72} & B_{73} \end{bmatrix}^T ;$$

- Vetor de parâmetros dependência (\mathbf{p}): nota-se que o modelo qLPV é dependente de todos os estados da atitude

$$\mathbf{p} = [p_1 \ p_2 \ p_3 \ p_4 \ p_5 \ p_6]^T = [\omega_\phi \ \phi \ \omega_\theta \ \theta \ \omega_\psi \ \psi]^T,$$

em que $\omega_* = \dot{*}$;

- Construção do *grid*: o intervalo escolhido para a validade do modelo foi $p_1 \in [-7, 7]$, $p_2 \in [-\pi/4, \pi/4]$, $p_3 \in [-7, 7]$, $p_4 \in [-\pi/4, \pi/4]$, $p_5 \in [-7, 7]$, $p_6 \in [-\pi/4, \pi/4]$. Ou seja, $\Omega = [-7, 7] \times [-\pi/4, \pi/4] \times [-7, 7] \times [-\pi/4, \pi/4] \times [-7, 7] \times [-\pi/4, \pi/4]$. A densidade do *grid* obtida foi de $10 \times 20 \times 10 \times 20 \times 10 \times 10$;
- Discretização: após a etapa de discretização, em que o *grid* descrito acima foi usado, obteve-se o tensor $\mathcal{S}^D \in \mathbb{R}^{10 \times 20 \times 10 \times 20 \times 10 \times 10 \times (10 \times 1)}$, que armazena as amostras;
- HOSVD e geração do fecho convexo: a HOSVD foi executada apenas para as seis primeiras direções, já que a última representa os dados vetoriais. Vale observar que:

$$\mathbb{R}^{10 \times 20 \times 10 \times 20 \times 10 \times 10 \times (10 \times 1)} = \mathbb{R}^{10 \times 20 \times 10 \times 20 \times 10 \times 10 \times 10}.$$

A primeira forma de escrita foi adotada para salientar que os dados são vetoriais. A Figura 15 mostra as funções de pertinência obtidas após a convexificação e interpolação das funções de ponderação.

A quantidade de regras é calculada pela multiplicação da quantidade de funções de pertinência em cada direção $2 \cdot 5 \cdot 2 \cdot 2 \cdot 3 \cdot 3 = 360$.

A resolução do conjunto de LMIs (6.8) para valores de $\gamma < 1$ foi factível. Apesar disso, houve problemas numéricos na simulação do modelo no Simulink® causados pelos altos ganhos de realimentação encontrados, impossibilitando a validação do controlador. Devido ao alto número de regras, existe a possibilidade de ter ocorrido problemas numéricos na solução das LMIs.

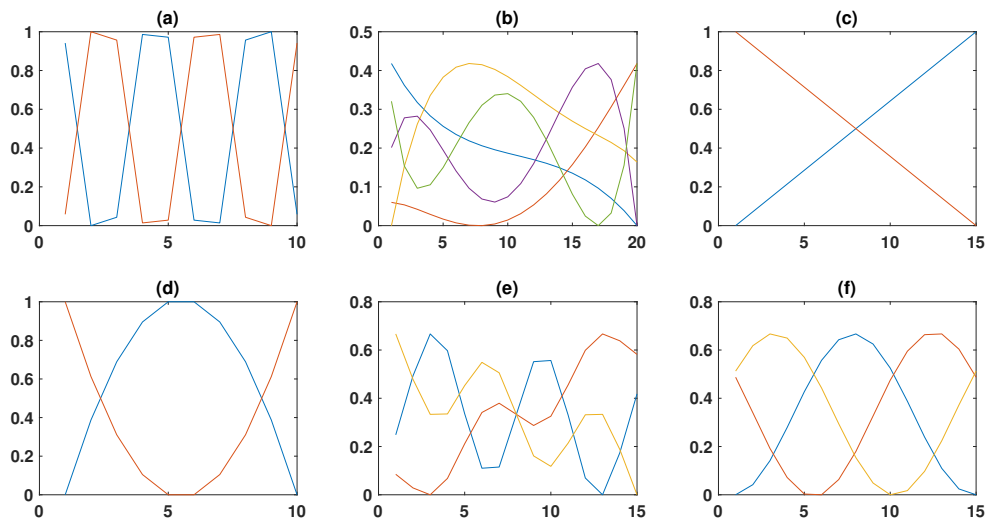


Figura 15 – Funções de pertinência do modelo sem simplificação; (a) $\mathbf{u}_{1,1}$ (azul) e $\mathbf{u}_{2,1}$ (vermelho); (b) $\mathbf{u}_{1,2}$ (azul), $\mathbf{u}_{2,2}$ (vermelho), $\mathbf{u}_{3,2}$ (amarelo), $\mathbf{u}_{4,2}$ (roxo) e $\mathbf{u}_{5,2}$ (verde); (c) $\mathbf{u}_{1,3}$ (azul) e $\mathbf{u}_{2,3}$ (vermelho); (d) $\mathbf{u}_{1,4}$ (azul) e $\mathbf{u}_{2,4}$ (vermelho); (e) $\mathbf{u}_{1,5}$ (azul), $\mathbf{u}_{2,5}$ (vermelho) e $\mathbf{u}_{3,5}$ (amarelo); (f) $\mathbf{u}_{1,5}$ (azul), $\mathbf{u}_{2,5}$ (vermelho) e $\mathbf{u}_{3,5}$ (amarelo).

Na abordagem em que o modelo (10.2) foi simplificado:

- Sistema de matrizes:

$$\mathbf{S}(\mathbf{p}) = \begin{bmatrix} A_{35} & A_{53} & A_{73} & B_{11} & B_{33} & B_{52} & B_{72} & B_{73} \end{bmatrix}^T;$$

- Vetor de parâmetros dependência (\mathbf{p}): nota-se que, mesmo após a simplificação, o modelo qLPV continuou dependente de todos os estados relacionados à atitude:

$$\mathbf{p} = [p_1 \ p_2 \ p_3 \ p_4 \ p_5 \ p_6]^T = [\omega_\phi \ \phi \ \omega_\theta \ \theta \ \omega_\psi \ \psi]^T.$$

Em alguns casos, após a simplificação, o modelo qLPV perde a dependência de algum parâmetro, diminuindo-se ainda mais o número de dimensões do tensor \mathcal{S}^D e, consequen-

temente, o custo computacional. Um exemplo seria um possível cenário de projeto de um controlador PDC apenas para as malhas da atitude, já que essas malhas passam a não depender de θ após a simplificação dos pequenos ângulos, conforme pode ser observado em (10.2).

- Construção do *grid*: o intervalo escolhido para a validade do modelo foi $p_1 \in [-7, 7]$, $p_2 \in [-\pi/4, \pi/4]$, $p_3 \in [-7, 7]$, $p_4 \in [-\pi/4, \pi/4]$, $p_5 \in [-7, 7]$, $p_6 \in [-\pi/4, \pi/4]$, ou seja, $\Omega = [-7, 7] \times [-\pi/4, \pi/4] \times [-7, 7] \times [-\pi/4, \pi/4] \times [-7, 7] \times [-\pi/4, \pi/4]$. A densidade do *grid* obtida foi de $10 \times 20 \times 10 \times 20 \times 10 \times 10$.
- Discretização: após a etapa de discretização, usando o *grid* descrito acima, obteve-se $\mathcal{S}^D \in \mathbb{R}^{10 \times 20 \times 10 \times 20 \times 10 \times 10 \times (8 \times 1)}$;
- HOSVD e geração do fecho convexo: a HOSVD foi executada apenas para as seis primeiras direções, já que a última representa os dados vetoriais. A Figura 16 mostra as funções de pertinência obtidas.

Sendo assim, a quantidade de regras para essa abordagem foi de $2 \cdot 3 \cdot 2 \cdot 1 \cdot 2 \cdot 2 = 48$, ou seja, 7,5 vezes menos regras que o resultado obtido na abordagem anterior.

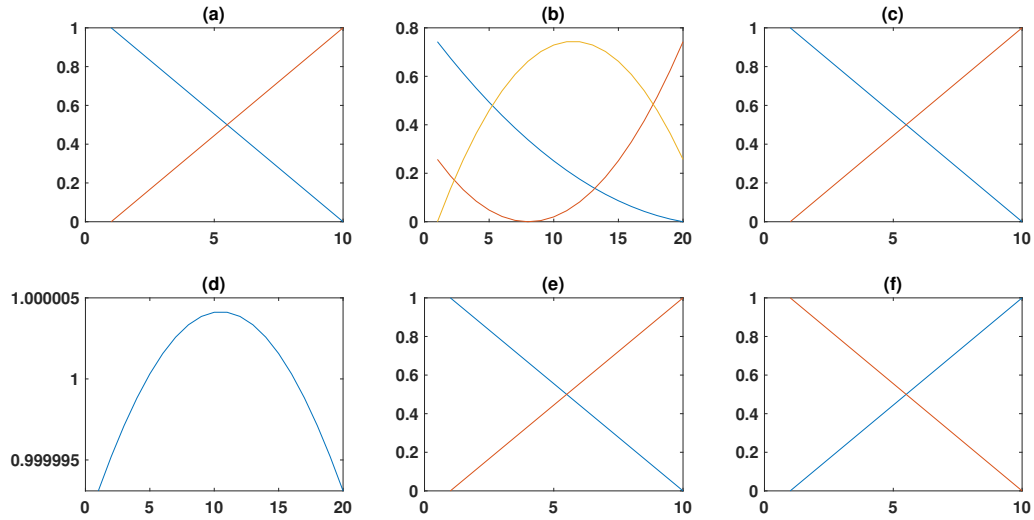


Figura 16 – Funções de pertinência do modelo simplificado; (a) $\mathbf{u}_{1,1}$ (azul) e $\mathbf{u}_{2,1}$ (vermelho); (b) $\mathbf{u}_{1,2}$ (azul), $\mathbf{u}_{2,2}$ (vermelho), $\mathbf{u}_{3,2}$ (amarelo); (c) $\mathbf{u}_{1,3}$ (azul) e $\mathbf{u}_{2,3}$ (vermelho); (d) $\mathbf{u}_{1,4}$ (azul); (e) $\mathbf{u}_{1,5}$ (azul) e $\mathbf{u}_{2,5}$ (vermelho); (f) $\mathbf{u}_{1,5}$ (azul) e $\mathbf{u}_{2,5}$ (vermelho).

Utilizou-se $\gamma = 0, 1$ para a resolução do conjunto de LMIs (6.8), maior valor testado para o qual a solução deu factível. A Figura 17 mostra a resposta do sistema utilizando o controlador PDC encontrado. Ressalta-se que, apesar do projeto do controlador ter utilizado o modelo simplificado, a simulação foi realizada com o modelo completo. Nota-se que o controlador mostrou uma sensibilidade ao ruído de processo e uma robustez ao ruído de medição. Ainda assim, ele conseguiu lidar muito bem com o problema de rastreamento de trajetória. A defasagem entre a

saída e a referência é causada pela ação integral advinda do uso do integrador para compor o sistema aumentado.

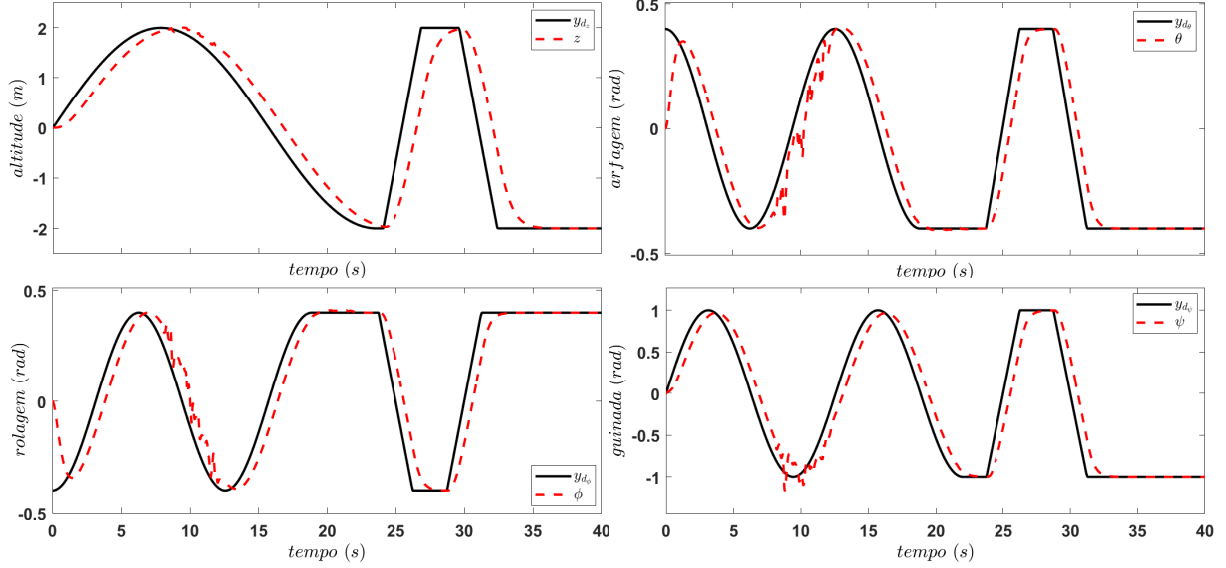


Figura 17 – Rastreamento de trajetória com o controlador PDC.

Os esforços de controle foram baixos, atingindo picos pequenos. Os maiores picos foram atingidos quando o sistema foi submetido ao ruído de processo. Mesmo nesse caso, os picos não tiveram valores muito altos. Nota-se que em regime permanente, U_1 tem aproximadamente o valor do peso do quadrotor (mg), conforme mostrado na Figura 18, o que já era esperado.

11.2 CONTROLADORES ADAPTATIVOS DIRETOS

Os valores dos parâmetros dos derivadores robustos e exatos, usados para estimar o valor de \dot{s} , são $\kappa = \sqrt{10}$ e $\alpha = 11$.

As funções de pertinência associadas a cada uma das variáveis premissas z_i dos sistemas nebulosos (FSs) são iguais e do tipo gaussianas:

$$\begin{aligned} \mu_{M_i^1}(z_i) &= \exp\left(-\frac{1}{2}\left(\frac{z_i+1,25}{0,6}\right)^2\right), \\ \mu_{M_i^2}(z_i) &= \exp\left(-\frac{1}{2}\left(\frac{z_i}{0,6}\right)^2\right), \\ \mu_{M_i^3}(z_i) &= \exp\left(-\frac{1}{2}\left(\frac{z_i-1,25}{0,6}\right)^2\right). \end{aligned}$$

O tipo gaussiano foi usado pela sua facilidade de implementação, pois é definido apenas por dois valores (média e desvio padrão), e por ser amplamente usado em sistemas nebulosos. Os valores da média e desvio padrão foram retirados do trabalho de (LABIOD; GUERRA, 2007) e, como apresentaram resultado satisfatório, foram mantidos.

As equações diferenciais que regem todas as malhas controladas do quadrotor são de segunda ordem, ou seja, os elementos do vetor r (7.2) são iguais 2, definindo o vetor de erro

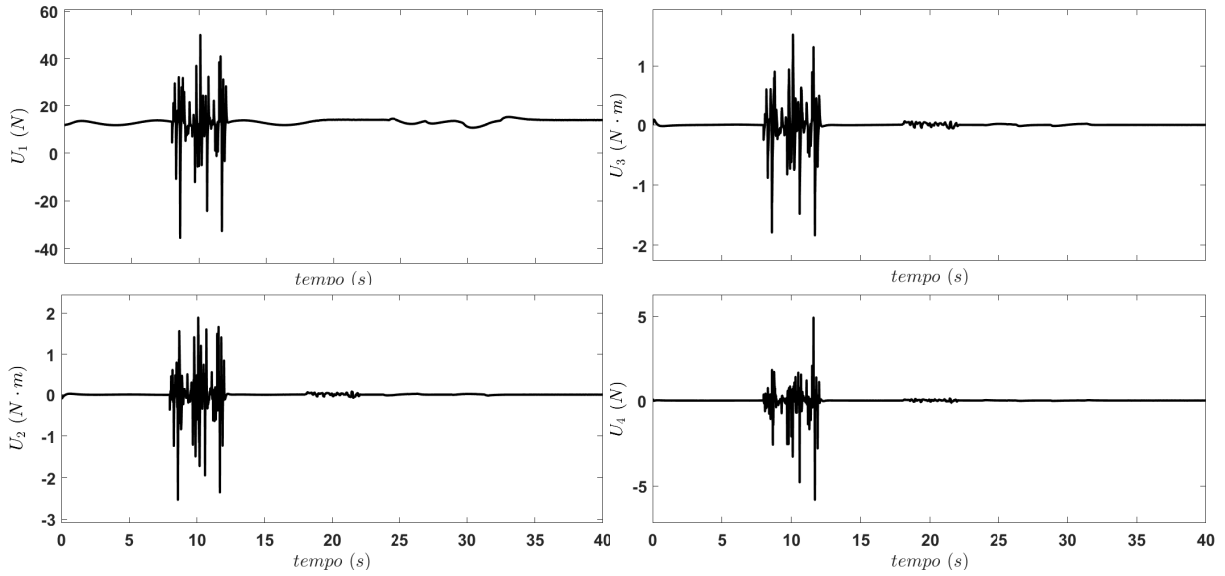


Figura 18 – Sinais de controle do controlador PDC.

filtrado (superfície deslizando) como

$$s(t) = \frac{de(t)}{dt} + \lambda e(t). \quad (11.1)$$

As tabelas 2 e 3 mostram, respectivamente, os valores obtidos manualmente para os parâmetros dos controladores da abordagem cujo controle da aeronave é feito com três controladores adaptativos e da abordagem cujo controle é feito com quatro.

Tabela 2 – Parâmetros usados na abordagem com três controladores adaptativos.

Controlador	Entradas do FS (\mathbf{z})	\mathbf{K}	\mathbf{K}_0	η	ϵ_0	σ	λ
Malha z	$[e_z(t) \dot{e}_z(t) \phi \theta]^T$	30	40	60	0,01	0,001	30
Malha ϕ e θ	$[e_\phi(t) \dot{e}_\phi(t) e_\theta(t) \dot{e}_\theta(t)]^T$	$[12 \ 12]^T$	$[40 \ 40]^T$	50	0,1	0,5	$[14 \ 14]^T$
Malhas ψ	$[e_z(t) \dot{e}_z(t) \phi \theta]^T$	80	60	130	0,01	0,001	20

Tabela 3 – Parâmetros usados na abordagem com quatro controladores adaptativos.

Controlador	Entradas do FS (\mathbf{z})	\mathbf{K}	\mathbf{K}_0	η	ϵ_0	σ	λ
Malha z	$[e_z(t) \dot{e}_z(t)]^T$	30	40	50	0,01	0,001	10
Malha ϕ	$[e_\phi(t) \dot{e}_\phi(t)]^T$	12	12	4	0,01	0,001	12
Malha θ	$[e_\theta(t) \dot{e}_\theta(t)]^T$	12	10	14	0,01	0,001	10
Malhas ψ	$[e_\psi(t) \dot{e}_\psi(t)]^T$	12	10	14	0,01	0,001	8

Apesar de utilizar um controlador a mais, a segunda abordagem entrega um número menor de regras. O número de regras para cada controlador é obtido por

$$n_r = n_{fp}^{n_z}, \quad (11.2)$$

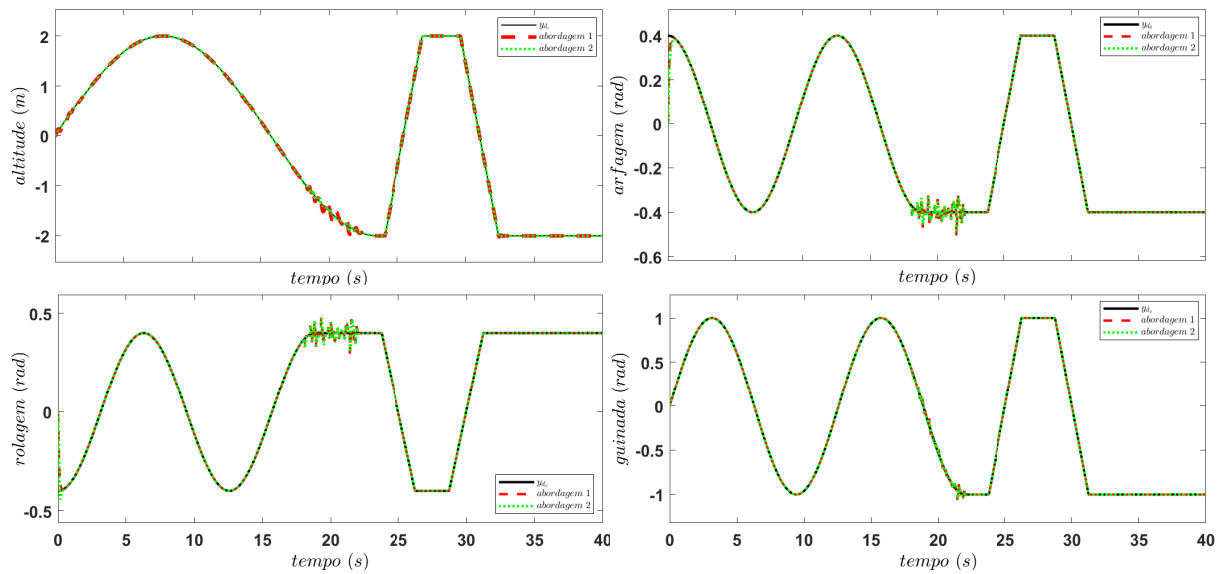


Figura 19 – Comparação entre as saídas duas abordagens adaptativas.

em que n_{fp} é o número de funções de pertinência e n_z é o número de entradas do sistema nebuloso. Na primeira abordagem, o número de regras é igual a $3^4 + 3^4 + 3^4 = 243$, enquanto na segunda, esse valor cai para $3^2 + 3^2 + 3^2 + 3^2 = 36$. A segunda abordagem possui 6, 5 vezes menos regras que a primeira, o que diminui significativamente o número de parâmetros a ser estimados pelo SN, tornando o controle muito mais rápido e com custo computacional menor. Além disso, a segunda abordagem apresentou um desempenho de rastreamento ligeiramente melhor (Figura 19) e a amplitude dos sinais de controle levemente mais baixos (Figura 20). As oscilações de alta frequência observadas nos gráficos das figuras 19 e 20 são decorrentes do fenômeno de *chattering*, característico do controle por modos deslizantes.

Como a segunda abordagem mostrou-se mais atrativa, optou-se por usar o algoritmo genético apenas nela. Primeiramente, tentou-se utilizar o AG com os seguintes parâmetros: tamanho da população igual a 40, quantidade máxima de gerações igual a 50, quantidade de indivíduos participantes do torneio igual a 6 e a probabilidade de mutação igual a 0,1. O único critério de parada usado foi atingir o número máximo de gerações estipulado. Porém, devido a uma convergência precoce, tentou-se aumentar a diversidade na população aumentando o tamanho da população para 50 indivíduos e a probabilidade de mutação para 0,3. A probabilidade de mutação passou a aumentar em 0,1 caso a aptidão média da população não se alterasse por 4 gerações consecutivas. Adicionou-se outro critério de parada: a aptidão média não mudar no intervalo de 8 gerações consecutivas. A Figura 21 mostra que, apesar de uma pequena melhora alcançada no segundo algoritmo, ambos algoritmos convergiram prematuramente. A Tabela 4 mostra os parâmetros obtidos pelo segundo AG.

O desempenho no rastreamento de trajetória dos controladores usando os parâmetros encontrados pelo AG foi praticamente o mesmo que o obtido pelos controladores sintonizados manualmente, conforme visto na Figura 22. O controlador com sintonia realizada pelo AG exibiu o sinal de controle U_1 ligeiramente menor, mas apresentou picos maiores no sinal U_2 (Figura

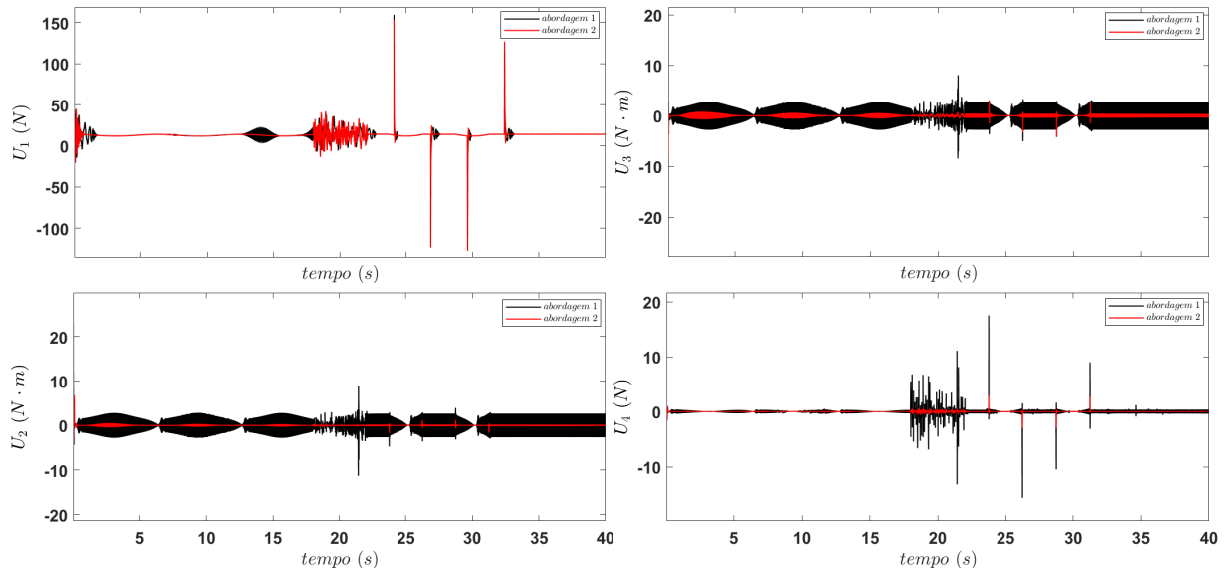


Figura 20 – Comparação entre as entradas de controle das duas abordagens adaptativas.

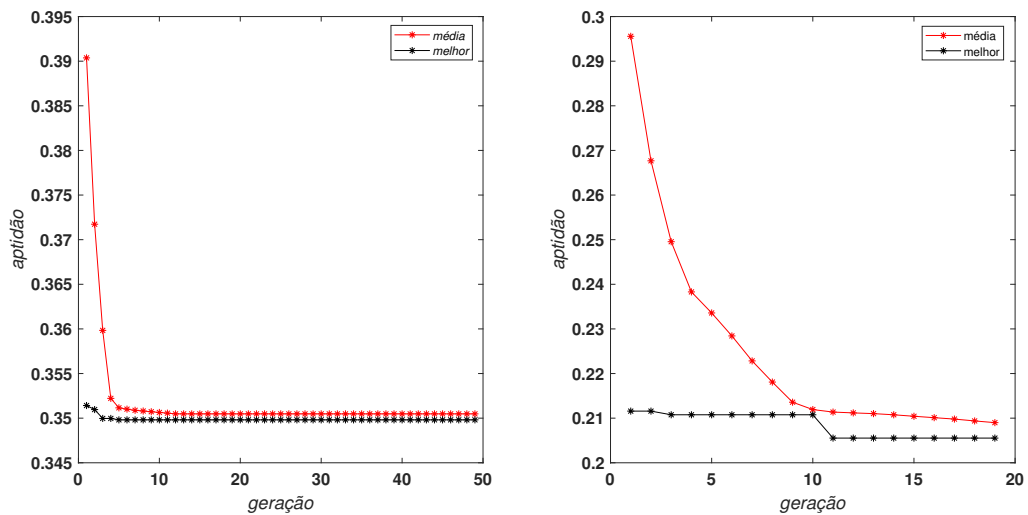


Figura 21 – Comparação entre as convergências dos AG implementados; com probabilidade de mutação fixa (esquerda); com probabilidade de mutação variável e população inicial expandida (direita).

Tabela 4 – Parâmetros encontrados pelo AG no projeto com quatro controladores adaptativos.

Controlador	Entradas do FS (\mathbf{z})	\mathbf{K}	\mathbf{K}_0	η	ϵ_0	σ	λ
Malha z	$[e_z(t) \dot{e}_z(t)]^T$	16,3643	9,7608	26,9306	0,0393	0,0024	14,5037
Malha ϕ	$[e_\phi(t) \dot{e}_\phi(t)]^T$	20,7641	15,7804	9,8974	0,0951	0,0085	14,3628
Malha θ	$[e_\theta(t) \dot{e}_\theta(t)]^T$	19,6833	25,0371	12	0,0611	0,0023	14,4165
Malhas ψ	$[e_\psi(t) \dot{e}_\psi(t)]^T$	13,3689	29,0203	14,0617	0,0878	0,0089	7,0279

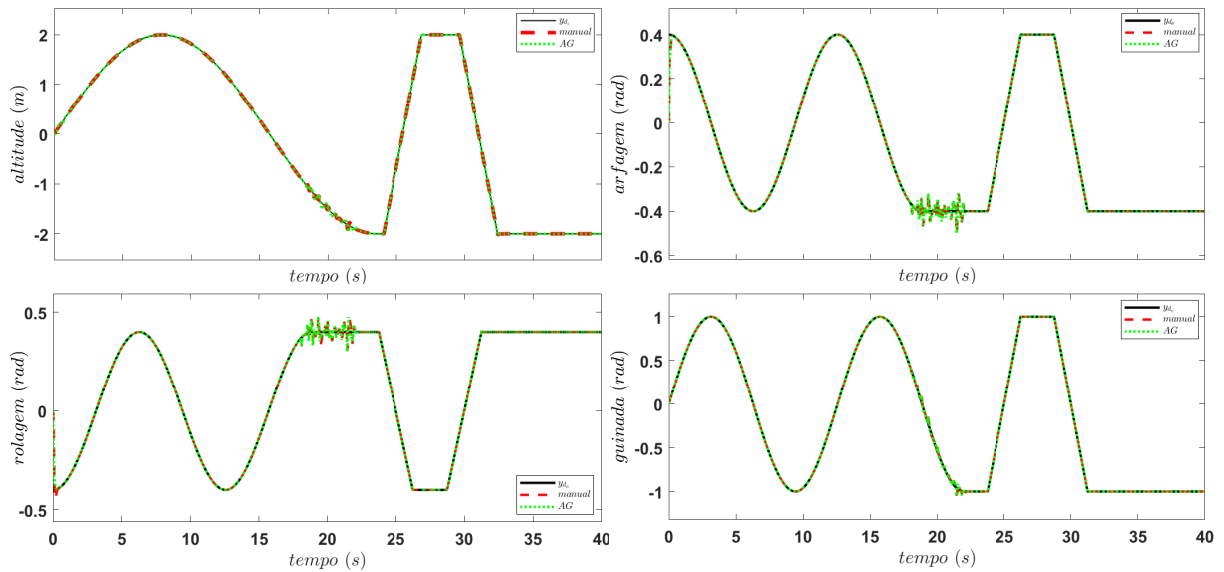


Figura 22 – Comparação entre as saídas dos controladores com sintonia manual e com sintonia via AG.

23). Essa proximidade no desempenho das sintonias pode ser explicada pelo fato de que houve uma nítida perda de sensibilidade dos controladores adaptativos em relação às variações de seus parâmetros no controle feito por quatro controladores. No controle feito por três controladores, a sensibilidade às variações dos parâmetros foi maior, talvez pelo fato de que o controlador responsável pelas malhas de rolagem e arfagem é MIMO (múltiplas entradas e múltiplas saídas) ao invés de ser MISO (múltiplas entradas e uma saída).

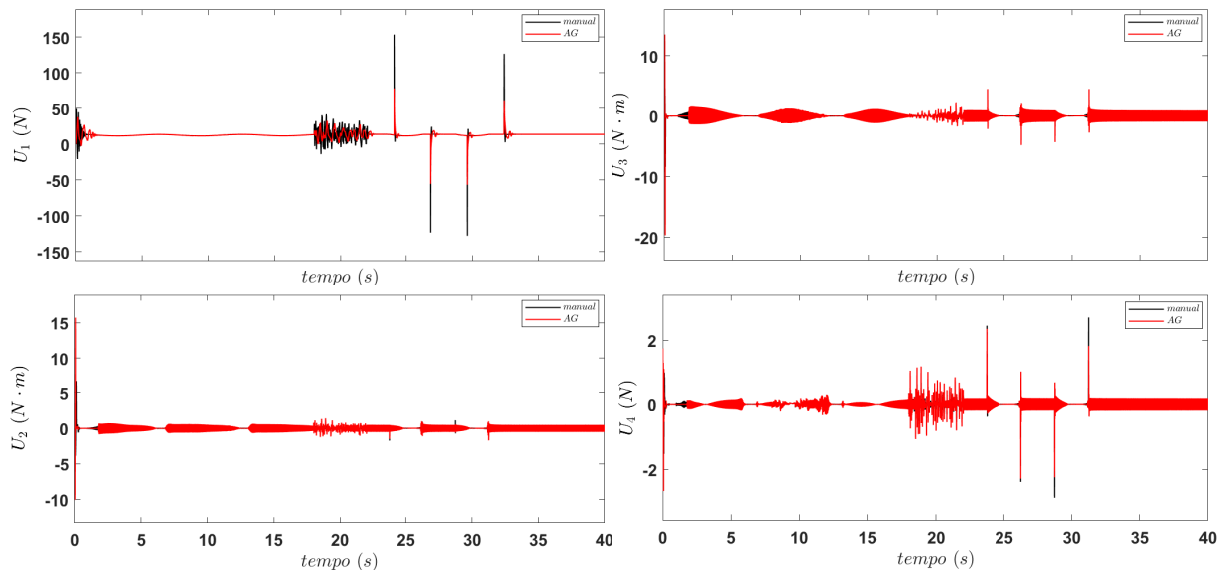


Figura 23 – Comparação entre as entradas de controle dos controladores com sintonia manual e com sintonia com o uso de AG.

Parte III

Discussões

CONCLUSÃO

Com exceção do controlador PDC sintetizado à partir do modelo qLPV completo, o restante dos controladores projetados conseguiram lidar com o problema de rastreamento de trajetória, mostrando desempenhos satisfatórios. Apesar de ter apresentado um melhor desempenho de saída, exibindo maior rejeição geral a ruídos e um melhor rastreamento de trajetória, o controle adaptativo direto possui uma implementação mais complexa, demonstra maiores esforços de controle e tem maior custo computacional, referente à etapa de adaptação, quando comparado ao controle PDC. Além disso, o projeto de controlador PDC é mais simples, envolvendo menos parâmetros de ajuste.

O controle da aeronave feita por três controladores adaptativos foi preterido ao controle feito por quatro controladores. Por apresentar maior sensibilidade à variação de seus parâmetros, sua sintonia é mais complicada. Além do mais, ele apresentou um número maior de regras, o que implica em mais parâmetros a serem estimados pelo SN e, conseqüentemente, repostas mais lentas e maior custo computacional. Ele apresentou também pior rastreamento de trajetória e maiores esforços de controle.

O uso do AG mostrou-se ineficaz no auxílio da sintonia dos controladores desenvolvidos neste trabalho. Além de possuir alto custo computacional, o seu emprego não encontrou parâmetros que provocassem melhoras significativas nos desempenhos dos controladores.

A simplificação do modelo para aplicação da técnica da decomposição do produto tensorial é uma alternativa bastante atraente para o projeto de controladores PDC para o quadrotor estudado. Ela permitiu diminuir o número de regras do modelo TS e o custo computacional no uso da técnica, além de apresentar um bom rastreamento de trajetória.

A adoção de dados vetoriais em detrimento de dados matriciais na técnica da decomposição do produto tensorial foi muito eficiente para diminuição do custo computacional intrinsecamente associado à técnica.

O uso do diferenciador robusto e exato mostrou-se uma alternativa eficaz para a implementação em tempo contínuo do controlador adaptativo direto proposto por (LABIOD; GUERRA, 2007).

TRABALHOS FUTUROS

Por tratar de temas amplos, diversas propostas podem ser formuladas para trabalhos futuros.

Existe a possibilidade de obter o modelo qLPV, usado na decomposição do produto tensorial, através da linearização em velocidades ou da linearização utilizando jacobianas.

Pretende-se aplicar diferentes condições LMIs, como as que utilizam custos H-infinito, e condições que considerem incertezas politópicas e distúrbios, o que, provavelmente, diminuirá a sensibilidade do controlador PDC à ruídos.

Modificações no AG poderão ser feitas para otimizar parâmetros de controladores adaptativos. Dentre elas, citam-se abordagens menos elitistas, tanto na seleção quanto na exclusão dos indivíduos, e a aplicação do AG na otimização das funções de pertinência, em que os cromossomos seriam compostos pelos pontos notáveis (média e desvio padrão) das funções gaussianas. Há também a possibilidade de otimizar as funções de pertinência de maneira adaptativa pelo método do gradiente (o que complicaria um pouco a análise da lei de adaptação, porque o modelo utilizado não será mais linear nos parâmetros a serem estimados). Novas propostas de normalização dos dados, diferentes funções de custo e uma implementação multiobjetivo do algoritmo poderão ser desenvolvidas.

Um tópico muito interessante é o abordado no trabalho de (CUI; ZHANG; MA, 2016). Eles propõem modificações no algoritmo clássico da técnica da transformação do produtor tensorial, que se baseiam em minimizar o número de pontos discretizados de tal forma que o algoritmo seja capaz de extrair o máximo de informação possível do modelo usando um menor número de pontos. Essas modificações objetivam a diminuição do alto custo computacional da técnica.

Uma outra possibilidade é a implementação de controles PDC via utilização da técnica da decomposição do produto tensorial baseado em superfícies deslizantes (TAKARICS, 2011).

Há a possibilidade de modificar o controlador adaptativo direto usando modos-deslizantes de ordem superior.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABRO, G.; ZULKIFLI, S.; ASIRVADAM, V.; ALI, Z. Model-free-based single-dimension fuzzy smc design for underactuated quadrotor uav. *Actuators*, v. 10, 2021. Citado na pág. 16.
- AGOSTINHO, A. Controle por modos deslizantes aplicado a sistemas de posicionamento dinâmico. *Dissertação (Mestrado) apresentada à Escola Politécnica da Universidade de São Paulo*, São Paulo, 2009. Citado na pág. 71.
- AL-MAHTURI, A.; SANTOSO, F.; GARRATT, M.; ANAVATTI, S. Nonlinear altitude control of a quadcopter drone using interval type-2 fuzzy logic. *IEEE Symposium Series on Computational Intelligence*, Bangalore, p. 236—241, 2018. Citado na pág. 16.
- ARAÚJO, F.; FONSECA, C.; MAITELLI, A.; MEDEIROS, A. de. Algoritmo genético para otimização de um controlador nebulosos para suspensão de vibrações. *Simpósio Brasileiro de Automação Inteligente*, Bauru, 2003. Citado na pág. 81.
- ASTRÖM, K.; WITTENMARK, B. *Adaptive Control*. New York: Dover Publications, INC, 2008. Citado na pág. 60, 61, 62.
- BARANYI, P.; YAM, Y.; VÁRLAKI, P. *Tensor Product Model Transformation in Polytopic Model-Based Control*. New York: CRC Press, 2014. Citado na pág. 34, 38, 39, 41, 46, 49, 52, 53.
- BASRI, M.; HUSAIN, A.; DANAPALASINGAM, K. A hybrid optimal backstepping and adaptive fuzzy control for autonomous quadrotor helicopter with time-varying disturbance. *Aerospace Engineering*, Institution of Mechanical Engineers, v. 229, 2015. Citado na pág. 16.
- BENIGNO, T. Modelagem matemática e controle de atitude e posição do quadrotor. *Dissertação (Mestrado) apresentada à Universidade Federal Rural do Semi-Árido*, Mossoró, 2015. Citado na pág. 16, 25, 26, 27, 87.
- BOUABDALLAH, S. Design and control of quadrotors with application to autonomous flying. *Tese (Doutorado) apresentada à École Polytechnique Fédérale de Lausanne*, Lausanne, 2007. Citado na pág. 24.
- BOYD, S.; GHAOUI, L.; FERON, E.; BALAKRISHNAN, V. *Linear Matrix Inequalities in System and Control Theory*. Philadelphia: Society for Industrial and Applied Mathematics, 1994. v. 15. Citado na pág. 58, 59.
- BRESCIANI, T. Modelling, identification and control of a quadrotor helicopter. *Dissertação (Mestrado) apresentada à Lund University*, Lund, 2008. Citado na pág. 20.
- CAMPOS, V. Takagi-sugeno in a tensor product approach-exploiting the representation. *Tese (Doutorado) apresentada à Universidade Federal de Minas Gerais*, Belo Horizonte, 2015. Citado na pág. 32, 34, 38, 45, 51.
- CAMPOS, V.; TÔRRES, L.; PALHARES, R. Revising the tp model transformation: Interpolation and rule reduction. *Asian Journal of Control*, v. 10, n. 2, p. 392—401, 2011. Citado na pág. 50.
- CASTILLO, P.; LOZANO, R.; DZUL, A. *Modelling and Control of Mini-Flying Machines*. London: Advances in Industrial Control, 2005. Citado na pág. 25.

CONCEIÇÃO, F.; BECCARO, W.; JUSTO, J. Ensinando métodos de discretização para projeto de sistemas de controle: Comparação de controladores pid em sistemas embarcados. *Journal of Applied Instrumentation and Control*, p. 8–15, 2020. Citado na pág. 76.

CUI, J.; ZHANG, K.; MA, T. An efficient algorithm for the tensor product model transformation. *International Journal of Control, Automation and Systems*, p. 1205–1212, 2016. Citado na pág. 48, 89, 104.

E.NUNES; HSU, L.; LIZARRALDE, F. Controle por realimentação de saídas para rastreamento exato de sistemas incertos. *Brazilian Conference on Automatic Control*, Gramado, 2004. Citado na pág. 80.

EVANS, P. Rotations and rotation matrices. *Acta Crystallographica*, p. 1355–1359, 2001. Citado na pág. 23.

FARREL, J.; POLYCARPOU, M. *Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches*. New York: John Wiley & Sons, Inc, 2006. Citado na pág. 74.

GOMIDE, F.; GUDWIN, R. Modelagem, controle, sistemas e lógica fuzzy. *SBA Controle & Automação*, v. 4, n. 3, p. 97–115, 1994. Citado na pág. 31.

GUARACY, F.; PINHEIRO, C. Sintonia de controlador por modos deslizantes para um sistema de nível. *XX Congresso Brasileiro de Automática*, Belo Horizonte, p. 2556–2563, 2014. Citado na pág. 73.

HAUPT, R.; HAUPT, S. *Practical Genetic Algorithms*. New York: John Wiley & Sons, Inc, 2004. v. 2. Citado na pág. 82.

HERRERA, F.; LOZANO, M.; VERDEGAY, J. Tuning fuzzy logic controllers by genetic algorithms. *International Journal of Approximate Reasoning*, New York, v. 12, p. 299—315, 1995. Citado na pág. 81.

JACOME, I. Controle adaptativo por modelo de referência e estrutura variável discreto no tempo. *Dissertação (Mestrado) apresentada à Universidade Federal do Rio Grande do Norte*, 2013. Citado na pág. 69.

JANKOVIC, M.; FONTAINE, D.; KOKOTOVIĆ, P. Tora example: Cascade- and passivity-based control designs. *IEEE Transactions on Control System Technology*, v. 4, n. 3, p. 292–297, 1996. Citado na pág. 53.

KALAB, A. A robust direct adaptive fuzzy control for a class of uncertain nonlinear mimo systems. *Soft Computing*, v. 23, p. 9747–9759, 2019. Citado na pág. 69.

KHALIFA, A.; FANNI, M.; RAMADAN, A.; ABO-ISMAIL, A. Modeling and control of a new quadrotor manipulation system. *First International Conference on Innovative Engineering Systems*, Alexandria, p. 1126—1131, 2013. Citado na pág. 27.

LABIOD, S.; GUERRA, T. Direct adaptive fuzzy control for a class of mimo nonlinear systems. *International Journal of Systems Science*, v. 38, n. 8, p. 665–675, 2007. Citado na pág. 69, 70, 71, 73, 74, 75, 76, 77, 78, 96, 103.

- LATHAUWER, L.; MOOR, B.; VANDEWALLE, J. A multilinear singular value decomposition. *Society for Industrial and Applied Mathematics*, v. 21, n. 4, p. 1253–1278, 2000. Citado na pág. 38, 39, 40.
- LEEL, H.; KIM, H. Robust control of a quadrotor using takagi-sugeno fuzzy model and an lmi approach. *International Conference on Control, Automation and Systems*, Korea, 2014. Citado na pág. 16.
- LEVANT, A. Robust exact differentiation via sliding mode technique. *Automatica*, v. 34, n. 3, p. 379–384, 1998. Citado na pág. 79, 80.
- LUUKKONEN, T. Modelling and control of quadcopter. *Dissertação (Mestrado) apresentada à School of Science*, Espoo, 2011. Citado na pág. 22, 26.
- MEDEIROS, A. de. Modelagem de sistemas dinâmicos não lineares utilizando sistemas fuzzy, algoritmo genéticos e funções de base ortonormal. *Dissertação (Mestrado) à Universidade Estadual de Campinas*, Campinas, 2006. Citado na pág. 81.
- MICHAEL, N.; MELLINGER, Q.; LINDSEY, Q.; KUMAN, V. The grasp multiple micro-uav test bed: Experimental evaluation of multirobot aerial control algorithms. *IEEE Robotics & Automation Magazine*, p. 57—65, 2010. Citado na pág. 27.
- MIZUMOTO, M.; TANAKA, K. Fuzzy sets and their operations. *Information and Control*, v. 48, p. 30–48, 1981. Citado na pág. 30.
- MOORE, C. Application of singular value decomposition to the design, analysis, and control of industrial processes. *American Control Conference*, p. 643–650, 1986. Citado na pág. 35.
- MORAIS, E. Desenvolvimento de um controlador híbrido fuzzy-pid para estabilização de um voo do tipo quadricóptero na realização de um voo autônomo vertical. *Dissertação (Mestrado) apresentada à Universidade Federal Rural do Semi-Árido*, Mossoró, 2017. Citado na pág. 16, 20.
- MORAIS, J. de. Algoritmo diferencial evolutivo aplicado à identificação de sistemas dinâmicos não-lineares. *Tese (Doutorado) apresentada à Universidade Federal de Uberlândia*, Uberlândia, 2006. Citado na pág. 81.
- MOREIRA, N.; SILVA, F. Automação da discretização de controladores pid e filtros digitais. *Revista Ciência e Tecnologia*, v. 17, n. 31, p. 12–22, 2014. Citado na pág. 76.
- MOZELLI, L. Controle fuzzy para sistemas takagi-sugeno: Condições aprimoradas e aplicações. *Dissertação (Mestrado) apresentada à Universidade Federal de Minas Gerais*, Belo Horizonte, 2008. Citado na pág. 30, 31.
- NAGY, S.; PETRES, Z.; BARANYI, P.; HASHIMOTO, H. Computational relaxed tp model transformation: Restricting the computational to subspaces of the dynamic model. *Asian Journal of Control*, v. 11, n. 5, p. 551–566, 2009. Citado na pág. 50.
- NIEMANN, D.; LI, J.; WANG, O.; TANAKA, K. Parallel distributed compensation for takagi-sugeno fuzzy models: New stability conditions and dynamic feedback design. *14th World Congress of IFAC*, p. 5374–5379, 1999. Citado na pág. 64.
- NUNEZ, I. O uso de controladores fuzzy adaptativos, implementados em microcontroladores, no controle de vibrações de sistemas mecânicos. *Dissertação (Mestrado) à Universidade de Uberlândia*, Natal, 2001. Citado na pág. 30.

- PAULA, J. Desenvolvimento de um vant do tipo quadricóptero para obtenção de imagens aéreas em alta definição. *Dissertação (Mestrado) apresentada à Universidade Federal do Paraná, Curitiba*, 2012. Citado na pág. 21.
- PETRES, Z. Polytopic decomposition of linear parameter-varying models by tensor-product model transformation. *Tese (Doutorado) apresentada à Universidade de Budapeste, Budapeste*, 2006. Citado na pág. 38, 58.
- PFEIFER, E. Projeto e controle de um uav quadricóptero. *Dissertação (Mestrado) apresentada à Escola Politécnica da Universidade de São Paulo, São Paulo*, 2013. Citado na pág. 22.
- PLEROU, A. Fuzzy genetic algorithms: Fuzzy logic controllers and genetics algorithms. *Global Journal for research analysis*, v. 5, p. 497—500, 2016. Citado na pág. 81.
- RABAH, M.; ROHAN, A.; HAN, Y.; KIM, S. Design of fuzzy-pid controller for quadcopter trajectory-tracking. *International Journal of Fuzzy Logic and Intelligent Systems*, v. 18, n. 3, p. 204—2013, 2018. Citado na pág. 16.
- RESENDE, D.; SILVA, A.; LIMA, E.; SENA, A. de. Desenvolvimento de um controlador híbrido fuzzy-pid e supervisor para estabilização de drone quadricóptero. *II Congresso Internacional de Gestão e Tecnologias*, Teresina, 2018. Citado na pág. 16.
- ROCKENBACH, L. Sistema quadricóptero: uma aplicação de fusão de sensores e fundamentos de controle. *Trabalho de Conclusão de Curso apresentado à Universidade Tecnológica do Paraná, Toledo*, 2018. Citado na pág. 28.
- SAMPAIO, L.; OLIVEIRA, M.; IGNACIO, A. Lógica nebulosa: Aplicações e tendências. *Simposio de Pesquisa Operacional e Logística da Marinha*, Rio de Janeiro, 2007. Citado na pág. 29.
- SHIRZADEH, M.; AMIRKHANI, A.; TORK, N.; TAGHAVIFAR, H. Trajectory tracking of a quadrotor using a robust adaptive type-2 fuzzy neural controller optimized by cuckoo algorithm. *ISA Transactions*, 2020. Citado na pág. 16.
- SIMÕES, M.; SHAW, I. *Controle e Modelagem Fuzzy*. São Paulo: Blucher, 2007. Citado na pág. 29.
- STEWART, G. On the early history of the singular value decomposition. *Society for Industrial and Applied Mathematics*, v. 35, n. 4, p. 551—566, 1993. Citado na pág. 35.
- TAKARICS, B. Parallel distributed compensation based sector sliding mode control of takagi-sugeno type polytopic models. *EEE Jubilee International Symposium on Applied Machine Intelligence and Informatics*, Slovakia, 2011. Citado na pág. 104.
- TANAKA, K.; IKEDA, T.; WANG, H. Fuzzy regulators and fuzzy observers: Relaxed stability conditions and lmi-based designs. *IEEE Transactions on Fuzzy Systems*, v. 6, n. 2, p. 250—265, 1998. Citado na pág. 64.
- TANAKA, K.; WANG, H. *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*. New York: John Wiley & Sons, Inc, 2001. Citado na pág. 58, 63, 66, 110.
- TSAI, S.; CHANG, Y.; LIN, H.; CHANG, L. Design and implementation of integral backstepping sliding mode control for quadrotor trajectory tracking. *Processes*, v. 9, n. 11, 2021. Citado na pág. 21.

VANANTWERP, J.; BRAATZ, R. A tutorial on linear and bilinear matrix inequalities. *Journal of Process Control*, v. 10, p. 363–385, 2000. Citado na pág. 57.

VIDAL, P. Rastreamento global e exato de sistemas incertos usando diferenciadores exatos não-lineares. *Projeto (Graduação) apresentado à Universidade Federal do Rio de Janeiro*, Rio de Janeiro, 2014. Citado na pág. 79.

WANG, H.; TANAKA, K.; GRIFFIN, M. Parallel distributed compensation for nonlinear systems by takagi-sugeno model. *Proceedings of 1995 IEEE International Conference on Fuzzy Systems*, Yokohama, p. 97–115, 1995. Citado na pág. 63.

WANG, P.; MAN, Z.; CAO, Z.; ZHENG, J. Dynamics modelling and linear control of quadcopter. *International Conference on Advanced Mechatronic Systems*, Melbourne, p. 498—503, 2016. Citado na pág. 27.

YACEF, F.; BOUHALI, O.; KHEBBACHE, H.; BOUDJEMA, F. Takagi-sugeno model of quadrotor modelling and control using nonlinear state feedback controller. *International Journal of Control Theory and Computer Modelling*, v. 2, n. 3, 2012. Citado na pág. 16.

YAM, Y.; BARANYI, P.; YANG, C. Reduction of fuzzy rule base via singular value decomposition. *IEE Transaction on Fuzzy System*, v. 7, n. 2, p. 120–132, 1999. Citado na pág. 34.

YOUNES, Y.; DRAK, A.; NOURA, H.; RABHI, A.; HAJJAJI, A. Model-free control of a quadrotor vehicle. *International Conference on Unmanned Aircraft Systems*, Orlando, p. 1126—1131, 2014. Citado na pág. 27.

ZADEH, L. Fuzzy sets. *Information and Control*, p. 338–353, 1965. Citado na pág. 30.

Considere a seguinte condição de síntese para sistemas TS

$$\sum_{i=1}^r \sum_{j=1}^r h_i h_j \mathbf{G}_{ij} \geq 0, \quad (1)$$

em que \mathbf{G}_{ij} são matrizes formadas pelos parâmetros do sistema e as variáveis de decisão do problema. Uma condição suficiente para garantir a validade da desigualdade acima é dada por (TANAKA; WANG, 2001)

Teorema .1 (Condição para somatório nebuloso). *Uma condição suficiente para para que expressão (1) seja válida, com $i, j \in 1, \dots, r$, é*

$$\begin{aligned} \mathbf{G}_{ii} &> 0, \\ \mathbf{G}_{ij} + \mathbf{G}_{ji} &> 0, \quad j > i \end{aligned}$$

Demonstração. Podemos reescrever (1) como

$$\sum_{i=1}^r \sum_{j=1}^r h_i h_j \mathbf{G}_{ij} = \sum_{i=1}^r h_i^2 \mathbf{G}_{ii} + \sum_{i=1}^{r-1} \sum_{j=i+1}^r h_i h_j (\mathbf{G}_{ij} + \mathbf{G}_{ji})$$

□