

Notas de Aula
Controle Usando Sistemas Nebulosos

Prof. Víctor Costa da Silva Campos
Prof. Leonardo Amaral Mozelli

Belo Horizonte, 8 de junho de 2022

“If only I had 5 lives!
Then I could be from 5 different
towns
and stuff myself full on 5
different types of food
and have 5 different jobs...
and I could...
fall in love with the same
person 5 times.”

Inoue Orihime, *Bleach*, 237,
Good Bye, Halcyon Days

Resumo

Estas notas de aula apresentam uma introdução a alguns tópicos sobre controle usando sistemas nebulosos, sendo dividida em quatro unidades.

A primeira unidade apresenta o básico sobre lógica nebulosa e sistemas nebulosos, além de apresentar a forma “clássica” de se projetar controladores usando sistemas nebulosos (a abordagem em que o controlador funciona como um sistema especialista, traduzindo conhecimento descrito de forma linguística para algo numérico e realizável). A referência principal utilizada nesta parte é Wang [1999](#).

O restante do conteúdo apresentado lida com maneiras formais de se provar o comportamento de controladores fuzzy. Para isso, a segunda unidade apresenta, principalmente, a teoria de estabilidade de Lyapunov, bem como algumas estratégias de controle de sistemas não lineares. A referência principal utilizada nesta parte é Khalil [2002](#).

A terceira unidade apresenta estratégias de controle de sistemas não lineares representados por modelos Takagi-Sugeno (TS) sintonizadas por meio de desigualdades matriciais lineares (LMIs - do inglês *Linear Matrix Inequalities*).

Por fim, a quarta unidade faz uso de sistemas nebulosos como aproximadores universais, e estende as técnicas de controle de sistemas não lineares para sistemas com dinâmica desconhecida. A ideia aqui é mostrar alguns fundamentos de controle adaptativo baseado em aproximação, e modificações robustas para as leis de adaptação. A referência principal utilizada nesta parte é Farrel e Polycarpou [2006](#).

Conteúdo

Lista de Figuras	vi
Notação	ix
1 Introdução	1
I Sistemas Nebulosos	4
2 Conjuntos Nebulosos e Operações Básicas	5
3 Operações em Conjuntos Nebulosos	19
4 Relações Nebulosas	29
5 Princípio da Extensão	38
6 Variáveis Linguísticas e Regras Nebulosas Se-Então	44
7 Lógica Nebulosa	51
8 Banco de Regras e Sistemas de Inferência Nebulosa	55
9 Fuzzificação e Defuzzificação (<i>Fuzzification and Defuzzification</i>)	60
10 Exemplo de Controle Utilizando Sistemas Nebulosos	65
11 Sistemas Takagi-Sugeno: Definições e Aproximadores Universais	79
12 Aproximação usando Modelos TS: Abordagem Linear nos Parâmetros	86
II Sistemas Não-Lineares	92
13 Sistemas Dinâmicos Não Lineares em Espaço de Estados	93

14	Estabilidade de Sistemas Dinâmicos Não-Lineares	101
15	Sistemas Dinâmicos Não Lineares em Tempo Discreto	108
16	<i>Boundedness e Ultimate Boundedness</i>	115
17	Feedback Linearization	118
18	Backstepping	124
19	Lista de Exercícios - Sistemas Não Lineares	132
III Controle Usando Desigualdades Matriciais Lineares		135
20	Sistemas Lineares e Desigualdades Matriciais Lineares	136
21	Desigualdades Matriciais Lineares Revisitadas	143
22	Não-Linearidade de Setor	157
23	Controle PDC	173
24	Observadores e o princípio da separação	189
25	Funções de Lyapunov Fuzzy	201
IV Controle Adaptativo Baseado em Aproximação		216
26	Controle Adaptativo Baseado em Aproximação	217
27	Controle Adaptativo por Aproximação - Lidando com o erro de aproximação e seguimento de trajetória	222
28	Modificações robustas das leis de adaptação	227
29	Filtros de geração de trajetória e feedback linearization (Input-State) com aproximação adaptativa	232
30	Backstepping com aproximação adaptativa	238
31	Controle Adaptativo por Aproximação - Caso discreto	243
32	Controle Adaptativo por aproximação - Mínimos Quadrados Recursivo	250
Bibliografia		258

A	Lemas interessantes para manipulações de Desigualdades Matriciais Lineares	261
B	Somatórios Fuzzy	267
C	Transformação do Produto Tensorial	270

Lista de Figuras

1.1	Representação dos elementos que compõem um sistema nebuloso.	1
1.2	Representação dos elementos que compõem um sistema de controle adaptativo utilizando um aproximador universal.	3
2.1	Diagrama Venn Crisp	6
2.2	Conjunto números reais menores que 2	6
2.3	Conjunto números naturais menores que 2	7
2.4	Conjuntos nebulosos para dias quentes e frios	9
2.5	União dos dias frios e quentes	15
2.6	Interseção dos dias frios e quentes	16
3.1	Exemplo de uma função de pertinência guassiana.	20
3.2	Complementos de Sugeno	20
3.3	Complementos de Yager	21
3.4	Exemplo de duas funções de pertinência guassianas.	23
3.5	Exemplos de s-normas	23
3.6	Exemplos de t-normas	25
3.7	Comparação do mapeamento $t(\alpha, \beta)$ para diferentes t-normas.	26
4.1	Exemplo de relação no \mathbb{R}^2	30
4.2	Diagrama Sagittal Crisp	31
4.3	Projeção e Extensão Cilíndrica	34
5.1	Conjunto nebuloso “ x é aproximadamente zero”.	41
5.2	Conjunto nebuloso “ y é aproximadamente zero”.	41
6.1	Exemplo de variável linguística	45
6.2	Conjunto nebuloso “ v é devagar”.	48
6.3	Conjunto nebuloso “ a é grande”.	48
6.4	Implicação de Dienes-Rescher para Se “ v é devagar” então “ a é grande”.	48
6.5	Implicação de Lukasiewicz para Se “ v é devagar” então “ a é grande”.	49
6.6	Implicação de Zadeh para Se “ v é devagar” então “ a é grande”.	49
6.7	Implicação de Mamdani do mínimo para Se “ v é devagar” então “ a é grande”.	49

6.8	Implicação de Mamdani do produto para Se “ v é devagar” então “ a é grande”.	50
7.1	Inferência utilizando uma função $y = f(x)$.	52
7.2	Inferência intervalar, utilizando uma função intervalar.	53
9.1	Exemplo de defuzzificador por centro de gravidade.	63
9.2	Exemplo de defuzzificador de média central. Neste exemplo, por coincidência, o valor defuzzificado coincidiu com o valor médio do segundo conjunto nebuloso.	63
9.3	Exemplo dos defuzzificadores de máximo	64
10.1	Sistema de dois tanques.	65
10.2	Comportamento do sistema de dois tanques em equilíbrio	69
10.3	Resposta ao degrau do sistema de dois tanques	69
10.4	Diagrama de simulação do sistema de dois tanques	70
10.5	Outro diagrama de simulação do sistema de dois tanques	71
10.6	Conjuntos nebulosos para nível alto e baixo	72
10.7	Conjuntos nebulosos para abertura da válvula	72
10.8	Comportamento do sistema de dois tanques com um controlador nebuloso	75
10.9	Conjuntos nebulosos para a variação de abertura da válvula	75
10.10	Comportamento do sistema de dois tanques com um controlador nebuloso integral	78
10.11	Diagrama de simulação do sistema de dois tanques em malha fechada.	78
11.1	Exemplo de aproximação de função utilizando uma função constante por partes.	82
11.2	Exemplo de aproximação de função utilizando uma função linear por partes.	83
11.3	Exemplo de aproximação de função utilizando funções de pertinência gaussianas.	83
13.1	Pêndulo simples	95
13.2	Exemplo de ponto de equilíbrio estável. Uma trajetória que se inicia na região δ se mantém na região ϵ . A curva em preto representa um exemplo de trajetória.	97
13.3	Exemplo de ponto de equilíbrio instável. As curvas em preto representam exemplos de trajetória do sistema. Não existe uma região δ tal que se o sistema se inicia nela ele permanece na região ϵ .	97
13.4	Exemplo de ponto de equilíbrio assintoticamente estável. Uma trajetória que se inicia na região δ se mantém na região ϵ . Além disso, a trajetória converge assintoticamente para o ponto de equilíbrio. A curva em preto representa um exemplo de trajetória.	98

20.1 Sistema 2 Equações	140
20.2 teste	141
20.3 teste	141
21.1 Região Factível Primeira LMI	145
22.1 Setores da função seno	160
22.2 Funções de pertinência utilizadas no exemplo	161
22.3 Funções de pertinência utilizadas no exemplo	163
22.4 Funções de pertinência utilizadas no exemplo	163
22.5 Plano de fase original do exemplo	171
22.6 Plano de fase normalizado do exemplo	171
C.1 $f(x, y) = xy(1 + \text{sen}(x) \cos(y))$, utilizada no exemplo C.2.	276
C.2 funções de peso obtidas no exemplo C.3.	277
C.3 funções de peso SN-NN obtidas no exemplo C.4.	279
C.4 funções de peso CNO obtidas no exemplo C.5.	282
C.5 funções de peso RNO-INO obtidas no exemplo C.6.	283
C.6 Erro quadrático de aproximação ao se utilizar as matrizes de peso do exemplo C.6.	284
C.7 diferentes funções de peso obtidas no exemplo C.7.	287

Notação

a	Escalar
\mathbf{a}	Vetor
A	Matriz
\mathcal{A}	Tensor
\mathbf{a}_i	i -ésimo vetor coluna da matriz A
a_{ij}	Elemento da linha i , coluna j da matriz A
$a_{i_1 i_2 \dots i_N}$	Elemento da posição (i_1, i_2, \dots, i_N) do tensor \mathcal{A}
$\mathcal{A}_{(n)}$	Matriz de modo- n do tensor \mathcal{A}
$A \otimes B$	Produto de Kronecker entre as matrizes A e B
$\langle \cdot, \cdot \rangle$	Operador de produto escalar (ponto)
\times_n	Produto de modo- n entre uma matriz e um tensor
$\ \cdot \ $	Operador de norma
$\mathcal{S} \times_{n=1}^N U^{(n)}$	Notação curta para $\mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \dots \times_N U^{(N)}$
$\mathbf{1}$	Vetor cujos componentes são todos iguais a um
A^T	Transposta da matriz A
$*$	Elementos transpostos dentro de uma matriz simétrica
$P > 0$	Indica que a matriz P é definida positiva
$P \geq 0$	Indica que a matriz P é semi-definida positiva
$P < 0$	Indica que a matriz P é definida negativa
$P \leq 0$	Indica que a matriz P é semi-definida negativa
$P \succ 0$	Indica que todos os elementos da matriz P são positivos
$P \succeq 0$	Indica que todos os elementos da matriz P são não-negativos
$P \prec 0$	Indica que todos os elementos da matriz P são negativos
$P \preceq 0$	Indica que todos os elementos da matriz P são não-positivos

Aula 1

Introdução

O objetivo principal dessa disciplina é apresentar noções básicas de controle usando sistemas nebulosos a estudantes de pós-graduação. Especificamente, apresentam-se três aspectos dos sistemas nebulosos: modelagem de representações linguísticas; aproximação universal; e combinação convexa de representações lineares. Para cada aspecto, indica-se como utilizá-lo em sistemas de controle.

Unidade I - Conjuntos e Sistemas Nebulosos

Na unidade I, iremos aprender sobre conjuntos e lógica nebulosa e como descrever sistemas a partir de um conjunto de regras nebulosas (*fuzzy*) se-então e uma máquina de inferência.

O objetivo final dessa unidade é, dada uma representação linguística de um sistema ou controlador, escrever um sistema nebuloso que modele esta descrição.

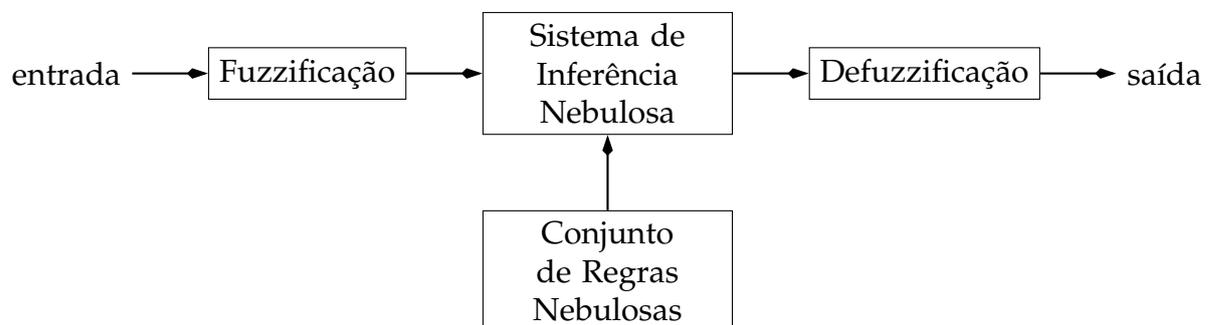


Figura 1.1: Representação dos elementos que compõem um sistema nebuloso.

Como veremos na Unidade I, sistemas nebulosos normalmente são mapeamentos não lineares entre a entrada e a saída. Dessa forma, a teoria de sistemas lineares, normalmente, não é suficiente para estudar o comportamento destes sistemas.

Unidade II - Introdução à Estabilidade e Controle de Sistemas Não-Lineares

Na Unidade II, iremos estudar um pouco sobre a teoria de estabilidade segundo Lyapunov e técnicas de controle usando funções de Lyapunov e “*Feedback Linearization*” (saída e estado).

O objetivo principal dessa parte é fundamentar a estabilidade e o controle de sistemas não lineares para que possamos utilizar esta teoria nas outras partes. Além disso, oferecer aos alunos um primeiro contato à técnica “*Feedback Linearization*” que de certa forma pode ser vista como uma generalização do método de síntese direta.

“Por alto”, considerando que temos um sistema de primeira ordem descrito por

$$\dot{x} = f(x) + g(x)u,$$

a ideia do “*Feedback Linearization*” é escolher uma lei de controle

$$u = \frac{1}{g(x)} (-f(x) + v),$$

que “cancela” a dinâmica original do sistema, e faz com que ele seja descrito por

$$\dot{x} = v.$$

Nas Unidades III e IV, focamos nossos estudos em um tipo particular de sistemas nebulosos, os sistemas Takagi-Sugeno (TS). Como veremos adiante, tais sistemas possuem características muito interessantes (podem ser vistos como uma combinação convexa de vários subsistemas lineares e são aproximadores universais - podem aproximar arbitrariamente qualquer função dentro de um conjunto compacto).

Unidade III - Controle baseado em otimização convexa

Na Unidade III, estudaremos como modelar problemas de controle como problemas de otimização sujeitos à desigualdades matriciais lineares. Começaremos com sistemas lineares incertos e estenderemos às condições destes sistemas para sistemas Takagi-Sugeno.

O objetivo dessa unidade é ter um primeiro contato com LMIs e entender alguns conceitos fundamentais.

Por exemplo, veremos que podemos verificar a estabilidade de um sistema TS se existir uma matriz $P = P^T$ tal que

$$P > 0, \\ A_i^T P + P A_i < 0, \quad \forall i$$

sendo A_i modelos lineares, localmente válidos, que servem como subsídio para modelar um sistema não linear.

Unidade IV - Controle Adaptativo Baseado em Aproximação

Na Unidade IV, iremos estudar uma forma específica de controle adaptativo que faz uso de aproximadores universais para controlar sistemas não lineares.

O objetivo principal dessa unidade é estudar o uso de modelos TS adaptados de forma a aproximar uma função não linear em malha fechada e usar esta aproximação na lei de controle.

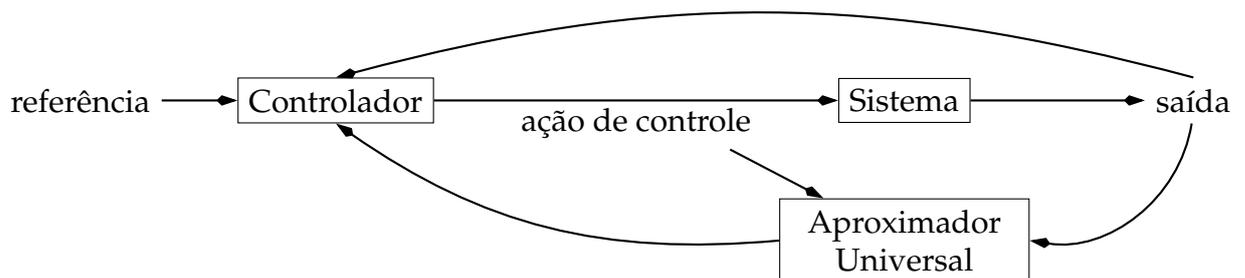


Figura 1.2: Representação dos elementos que compõem um sistema de controle adaptativo utilizando um aproximador universal.

Unidade I

Sistemas Nebulosos

Aula 2

Conjuntos Nebulosos e Operações Básicas

Nesta aula, temos um primeiro contato com os conjuntos nebulosos. São apresentadas as definições dos conjuntos nebulosos, bem como algumas operações básicas que podem ser realizadas sobre o conjunto.

Iniciamos nossa discussão nesta aula lembrando um pouco sobre a teoria de conjuntos clássica.

Definição 2.1: Universo de Discurso

O **universo de discurso** é o conjunto que contém todos os elementos de interesse para um dado contexto ou aplicação.

Na teoria de conjuntos clássica/*Crisp*, temos que um conjunto A , contido em um universo de discurso U ($A \subset U$), pode ser definido de três formas:

- No *método da lista*, um conjunto é definido pela listagem de todos os seus elementos. Este método só pode ser usado para o caso de conjuntos finitos;
- No *método da regra*, um conjunto é definido por meio de propriedades que devem ser satisfeitas por seus elementos. Normalmente, conjuntos descritos por este método são descritos como

$$A = \{x \in U \mid x \text{ atende algum requisito}\};$$

- No *método da função indicadora/pertinência*, define-se uma função indicadora do conjunto A , denotada como $\mu_A(x)$, tal que

$$\mu_A(x) = \begin{cases} 1, & \text{se } x \in A \\ 0, & \text{se } x \notin A \end{cases}$$

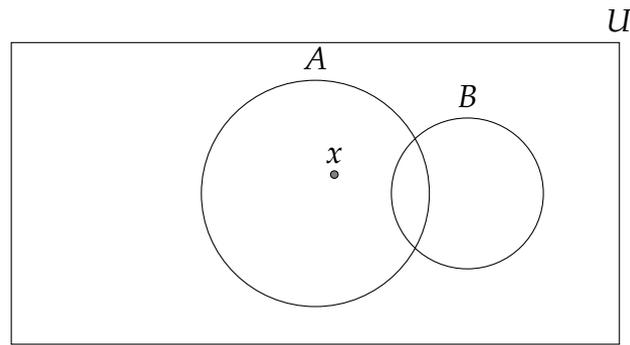


Figura 2.1: Diagrama de Venn representando dois conjuntos, A e B , em um universo de discurso U . O elemento x pertence ao conjunto A , mas não pertence a B .

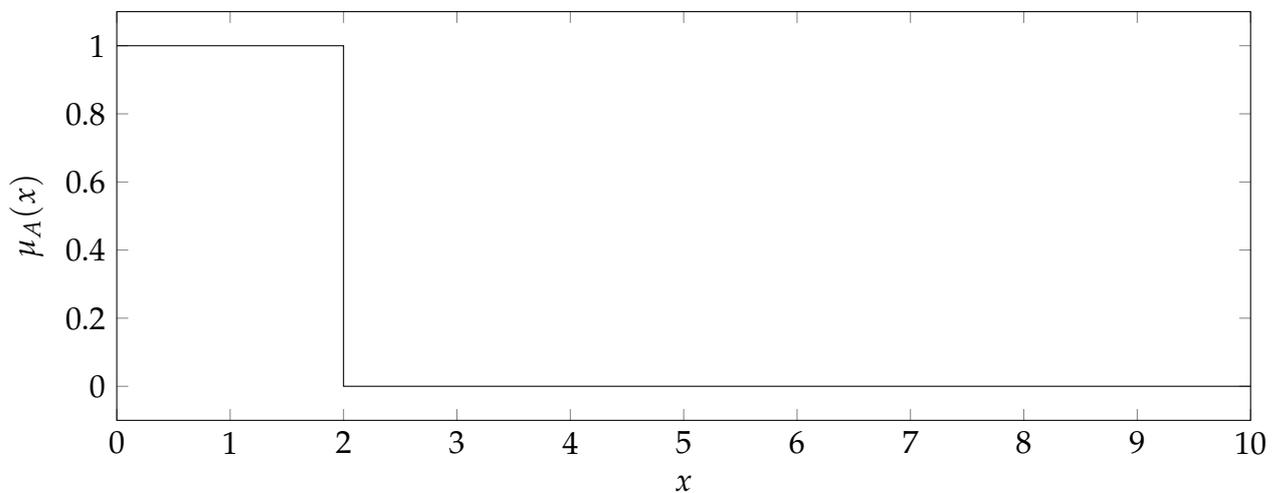


Figura 2.2: Representação gráfica da função de pertinência do conjunto $A = \{x \in \mathbb{R}^+ \mid x < 2\}$.

Esquemáticamente, é possível representar conjuntos por meio de Diagramas de Venn, nos quais elementos são indicados por pontos em um espaço bidimensional e os conjuntos como regiões delimitadas por curvas. Um elemento $x \in A$ caso o ponto representando este elemento esteja no interior da região representando o conjunto, conforme ilustrado na Figura 2.1. Seguindo esta forma, nota-se que $x \notin B$. Tal forma de representação é útil para conjuntos limitados ou para ilustrar relações e operações com conjuntos, as quais serão discutidas mais adiante.

Alternativamente, é possível representar graficamente o método da função indicadora/pertinência. No eixo das abcissas são informados os elementos enquanto no eixo das ordenadas são assinalados os respectivos valores de pertinências. A Figura 2.2 ilustra um exemplo de representação para o conjunto dos números reais não negativos menores que 2, no universo de discurso $U \leftrightarrow [0, 10]$, enquanto a Figura 2.3 ilustra o conjunto dos números naturais não negativos menores que 2 no mesmo universo de discurso.

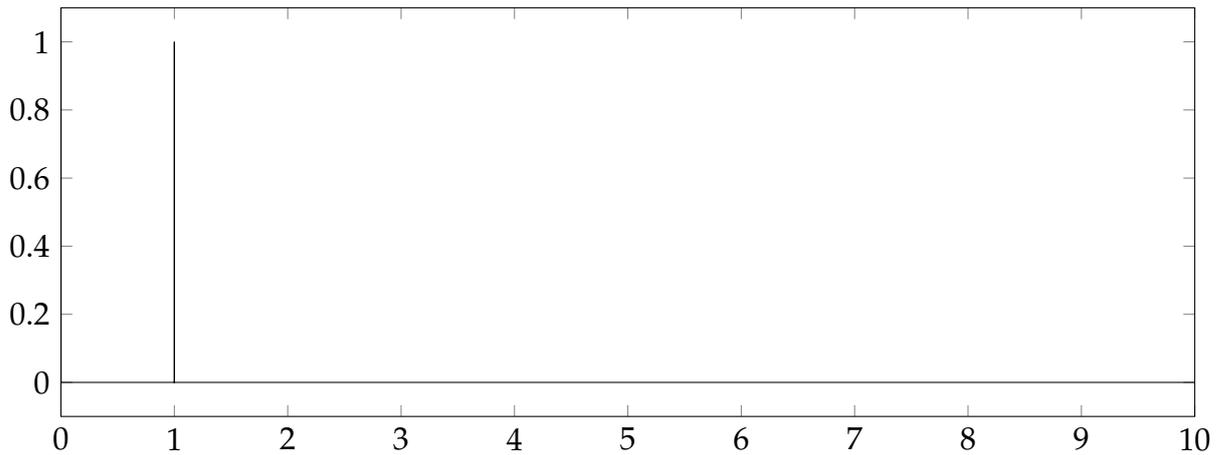


Figura 2.3: Representação gráfica da função de pertinência do conjunto $A = \{x \in \mathbb{N}^+ \mid x < 2\}$.

Apesar de ser muito utilizada, a teoria de conjunto clássica pode ser muito restritiva em alguns casos. Considere, por exemplo, que queremos descrever os dias quentes e os dias frios por dois conjuntos, QUENTE e FRIO. Podemos descrever tais conjuntos como

$$\text{QUENTE} = \{\text{dias em que a temperatura é maior do que } 27 \text{ graus}\},$$

$$\text{FRIO} = \{\text{dias em que a temperatura é menor do que } 18 \text{ graus}\}.$$

Note que essa abordagem nos permite descrever os dias quentes e os dias frios, de uma forma bastante precisa e que possa ser traduzida como um algoritmo. Entretanto, ela possui a desvantagem de não ser muito intuitiva (com o que consideramos na prática). O problema com esta representação é que enquanto um dia em que a temperatura é de 27,1 graus é considerado quente, um dia em que a temperatura é de 26,9 graus não. Tal mudança é irrelevante para nossa percepção da condição climática, porém, de acordo com a descrição clássica, há uma mudança muito súbita na classificação, o que não condiz bem com a realidade.

Outro problema nesta abordagem ocorre com valores intermediários na escala de temperatura. Considere que neste momento, do lado de fora, o termômetro esteja marcando 22,5 graus. De acordo com sua percepção, este é um dia quente? Bem, de certa maneira sim, mas nem tanto. Definitivamente, não tão quente quanto um dia que o termômetro marca 30 graus. Este é outro contra exemplo em detrimento da teoria clássica de conjuntos, denotando um caráter vago, ambíguo ou impreciso de alguns predicados, como por exemplo QUENTE.

Finalmente, a descrição apresentada anteriormente reflete um certo viés. A separação proposta para os conjuntos QUENTE e FRIO pode ser adequada para habitantes da região Sudeste do Brasil. Porém, em um país de grandes dimensões e diferenças

climáticas como o nosso, habitantes das regiões Sul podem discordar com 18 graus represente um dia FRIO.

Diferentemente dos conjuntos clássicos, os conjuntos nebulosos são descritos pelos elementos que os compõem e o grau de pertinência de cada elemento ao conjunto. Ou seja, os elementos não mais simplesmente pertencem ou não a um conjunto, podendo ter um valor intermediário de pertinência.

Definição 2.2: Conjunto Nebuloso

Um **conjunto nebuloso** A em um universo de discurso U é definido por uma função de pertinência $\mu_A(x)$ que toma valores no intervalo $[0, 1]$.

Voltando ao nosso exemplo, podemos definir os conjuntos QUENTE e FRIO pelas funções de pertinência

$$\mu_q(T) = \begin{cases} 0, & \text{se } T < 18 \\ \frac{T - 18}{27 - 18}, & \text{se } 18 \leq T \leq 27 \\ 1, & \text{se } T > 27 \end{cases}$$

$$\mu_f(T) = \begin{cases} 1, & \text{se } T < 18 \\ \frac{27 - T}{27 - 18}, & \text{se } 18 \leq T \leq 27 \\ 0, & \text{se } T > 27 \end{cases}$$

Tais funções de pertinência são apresentadas na Figura 2.4. Neste novo conceito, dias com temperaturas acima de 27 graus são considerados com pertinência 1 ao conjunto QUENTE. Retomando nosso exemplo, o dia cuja temperatura marca 26,9 graus é considerado com pertinência 0,9889, indicando também uma grande pertinência ao conjunto QUENTE, mas um pouco menor do que o dia que marca 27,1 graus. Já o dia cujo termômetro marca 22,5 graus é considerado com pertinência 0,5 tanto ao conjunto QUENTE quanto ao conjunto FRIO. Ou seja, a imprecisão associada a descrição deste dia foi melhor capturada como “meio” QUENTE e “meio” FRIO. Ajustando a função de pertinência é possível adequar esta descrição de forma mais intuitiva à percepção de um dado conjunto de pessoas, habitantes de uma determinada região, ajustando um certo viés ou se adequando a um conjunto de dados.

Para conjuntos fuzzy limitados, uma notação similar ao método da lista consiste em enumerar todos elementos acompanhados dos respectivos valores de pertinência ao conjunto. Por exemplo:

$$A = \left\{ \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \dots + \frac{\mu_A(x_n)}{x_n} \right\} = \sum_{i=1}^n \frac{\mu_A(x_i)}{x_i}$$

ou

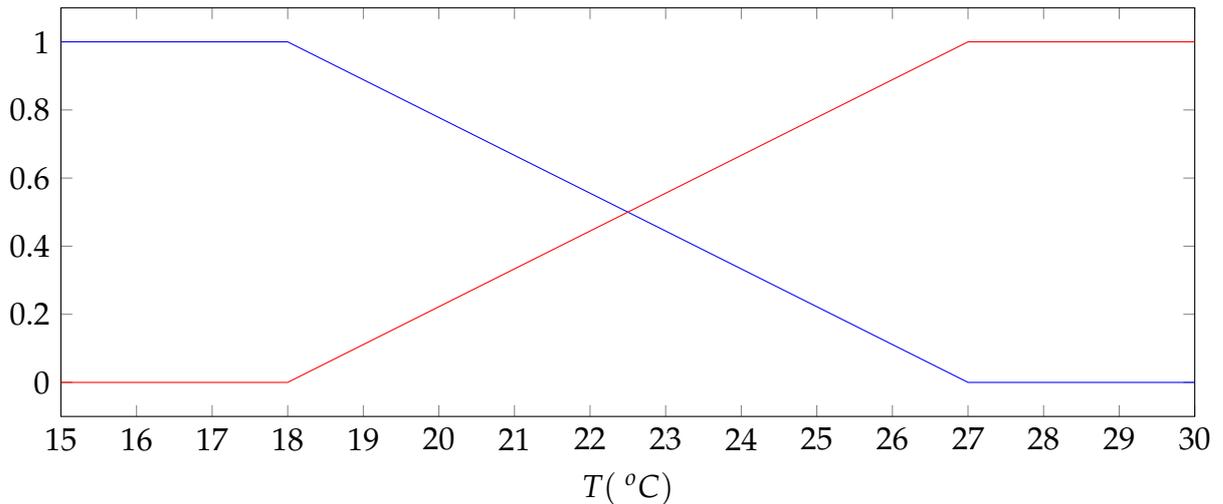


Figura 2.4: Funções de pertinência para os conjuntos de dias quentes e frios. A função de pertinência do conjunto QUENTE é apresentada em **vermelho**, enquanto que a função de pertinência do conjunto FRIO é apresentada em **azul**.

$$A = \{(\mu_A(x_1), x_1), (\mu_A(x_2), x_2), \dots, (\mu_A(x_n), x_n)\} = \{(\mu_A(x), x) | x \in U\}$$

A primeira forma de representação é atribuída a Zadeh 1965¹. Nela a barra não indica uma divisão, apenas um delimitador. Além disso os sinais de + e \sum não indicam soma, mas a união dos elementos. Normalmente, apenas elementos com pertinência não-nula são listados.

No caso de domínio contínuos, adota-se uma forma similar da primeira notação:

$$A = \left\{ \int \frac{\mu_A(X)}{x} \right\}$$

com as mesmas ressalvas sobre a barra indicar um delimitador e sinal de \int indicar uma união.

¹É um consenso acadêmico que esta referência trata-se da certidão de nascimento dos Sistemas Nebulosos. Seu autor, Lotfi Aliasker Zadeh, engenheiro eletricista e cientista da computação, nasceu em Baku, Azerbaijão, então integrante da União Soviética, em 4 de fevereiro de 1921 e faleceu, recentemente, em 6 de setembro de 2017 (Bonissone e Ruspini 2017). Ao longo de sua carreira, Zadeh recebeu diversas honrarias e distinções. Life Fellow do IEEE, recebeu três de suas mais distintas medalhas: IEEE Education Medal (1973), IEEE Richard W. Hamming Medal (1992), e IEEE Medal of Honor (1995). Também foi recipiente do ACM's Allen Newell Award (2001), Honda Prize (1989), e BBVA Foundation Frontiers of Knowledge Award (2012). Em Bělohávek, Dauben e Klir 2017 é dada uma perspectiva histórica do desenvolvimento de Sistemas Nebulosos. Destaca-se a participação de Zadeh, transcendendo o papel de inventor, sendo ativo divulgador e desenvolvedor desta teoria.

Exercício 2.1

Gás ácido ou gás azedo refere-se, em geral, ao gás natural bruto, o qual possui baixos teores de contaminantes em sua composição. Os principais componentes do gás natural são hidrocarbonetos como metano (CH_4), etano (C_2H_6) e propano (C_3H_8). Entre os contaminantes, o sulfeto de hidrogênio (H_2S), ou ácido sulfídrico, é altamente corrosivo, tóxico e compromete o desempenho do gás natural, necessitando ser removido. Por isso, recaem as especificações mais rígidas para sua presença no gás natural comercial. No Brasil, limita-se a presença de ácido sulfídrico entre 10 a 15 ppm; nos Estados Unidos esse valor é limitado a aproximadamente 4ppm; em outros países o valor máximo é 1ppm (Lopes 2003). Entre os métodos utilizados pelas indústrias para a remoção de H_2S presente no gás natural, a absorção usando aminas como solvente é bastante comum. Tal procedimento é referido como “adoçamento” do gás. Com base nestas informações, responda:

- Defina um conjunto fuzzy “BAIXA CONCENTRAÇÃO”, visando relacionar uma função de pertinência com a concentração de H_2S , indicada pela variável C , medida em ppm.
- Defina um conjunto fuzzy “DENTRO DA ESPECIFICAÇÃO”, visando relacionar uma função de pertinência com a concentração de H_2S , usando a mesma variável C .
- Com base apenas nas informações até aqui, é possível definir conjuntos “ALTA CONCENTRAÇÃO” ou “CONCENTRAÇÃO PERIGOSA”? Justifique.

Conceitos Básicos

Definição 2.3: Suporte

O suporte de um conjunto nebuloso são os elementos cuja pertinência é maior do que zero.

$$\text{supp}(A) = \{x \in U \mid \mu_A(x) > 0\}$$

Definição 2.4: Conjunto Vazio

Um conjunto nebuloso é chamado de vazio quando seu suporte é vazio.

$$A = \emptyset \Leftrightarrow \text{supp}(A) = \emptyset$$

Definição 2.5: Singleton

Um conjunto nebuloso é chamado de **singleton** quando possui um único elemento.

Definição 2.6: Ponto de Cruzamento

O **ponto de cruzamento** de um conjunto nebuloso é o ponto em que a função de pertinência é 0.5.

Definição 2.7: Altura

A altura de um conjunto nebuloso é o valor máximo de sua função de pertinência.

Definição 2.8: Conjunto nebuloso normalizado

Um conjunto nebuloso é dito **normalizado** se a sua altura é igual a 1.

Definição 2.9: Corte α

O **corte α** de um conjunto nebuloso é o conjunto dado por

$$A_\alpha = \{x \in U \mid \mu_A(x) > \alpha\}.$$

Definição 2.10: Centro

O **centro** de um conjunto nebuloso é dado pela média dos valores em que a função de pertinência atinge seu valor máximo. Caso essa média seja infinita, o centro é dado pelo primeiro valor em que o conjunto nebuloso atinge o seu valor máximo de pertinência.

Relações e Operações com Conjuntos

Na teoria de conjuntos clássica/*Crisp*, é possível estabelecer operações e relações entre os conjuntos. Recordando, para conjuntos A e B em U tem-se:

- $A \subset B$, A está completamente contido em B (se $x \in A$, então $x \in B$)
- $A \subseteq B$, A está contido ou equivale a B

Teoria de Conjuntos Clássica	Álgebra Booleana
$A \cup B$	$p \text{ OU } q$
$A \cap B$	$p \text{ E } q$
$A \leftrightarrow B$	$p = q$
$A \subseteq B$	$p \rightarrow q$
\bar{A}	NÃO p

Tabela 2.1: Equivalência entre operadores teoria de conjuntos e operadores booleanos

- $A \leftrightarrow B$, $A \subset B$ e $B \subset A$ (A é equivalente a B)
- União: $A \cup B$, elementos que pertencem a A , que pertencem a B ou pertencem a ambos.
- Interseção: $A \cap B$, elementos que pertencem, simultaneamente, aos conjuntos A e B
- Complemento: \bar{A} , todos os elementos do universo de discurso que não pertencem a A

A teoria clássica/*Crisp* de conjuntos pode servir como uma interpretação da Álgebra Booleana. Neste caso, um valor verdadeiro pode ser interpretado como pertencente e um valor falso indica não pertencente. Assim é possível estabelecer a correspondência da Tabela 2.1 para operadores e relações.

Outra forma de sistematizar as operações na teoria clássica de conjuntos é por meio de mapeamentos com os seus elementos e respectivas funções indicadoras/pertinências. Assim, temos que a operação de União entre os conjuntos A e B pode ser calculada como :

$$A \cup B \rightarrow \mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x) = \max(\mu_A(x), \mu_B(x))$$

Para a Interseção segue que:

$$A \cap B \rightarrow \mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) = \min(\mu_A(x), \mu_B(x))$$

O Complemento é tido como:

$$\bar{A} \rightarrow \mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

Um subconjunto é definido como:

$$A \subseteq B \rightarrow \mu_A(x) \leq \mu_B(x)$$

Operadores de Zadeh

Assim como para os conjuntos clássicos, também podemos definir as operações de complemento, união e interseção para os conjuntos nebulosos.

Existem várias formas possíveis de se definir estas operações, mas na aula de hoje vamos focar nos operadores de Zadeh ou operadores padrão.

Definição 2.11: Complemento de Zadeh

O complemento do conjunto nebuloso A , representado por \bar{A} , é definido pela função de pertinência

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x).$$

Definição 2.12: União de Zadeh

A união de dois conjuntos nebulosos A e B , representada por $A \cup B$, é definida pela função de pertinência

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)).$$

Exercício Resolvido 2.1

Considere dois conjuntos fuzzy A e B , tais que:

$$A = \{(0,5, x_1), (0,1, x_2), (0,4, x_3)\} \text{ e } B = \{(0,2, x_1), (0,3, x_2), (0,4, x_3)\}$$

Quais são os elementos do conjunto formado por $A \cup B$?

Solução:

Para cada elemento x_i , $i = 1, 2, 3$, os valores das pertinências no conjuntos A e B , respectivamente, $\mu_A(\cdot)$, $\mu_B(\cdot)$, devem ser comparados, sendo preservado aquele de maior valor. Logo

$$\begin{aligned} A \cup B &= \{(\max(0,5, 0,2), x_1), (\max(0,1, 0,2), x_2), (\max(0,4, 0,4), x_3)\} \\ &= \{(0,5, x_1), (0,2, x_2), (0,4, x_3)\} \end{aligned}$$

Definição 2.13: Interseção de Zadeh

A interseção de dois conjuntos nebulosos A e B , representada por $A \cap B$, é definida pela função de pertinência

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)).$$

Exercício Resolvido 2.2

Retome o Exercício Resolvido 2.1. No processo de “adoçamento”, o gás azedo entra em contato com uma solução de amina “magra” em um absorvedor. Isso permite que a amina remova o componente ácido, produzindo uma solução rica em amina e um “gás comercial” ou gás natural “doce”, com concentração aceitável de contaminantes segundo as normas locais. Considere a escolha de um analisador para monitorar a concentração de H_2S . Esta seleção se tornou complicada, pois duas figuras de mérito devem ser analisadas. O analisador X consegue efetuar a leitura em uma ampla faixa de pressão (P), mas tem um valor limítrofe para o tempo morto (TM); por outro lado, o analisador Y , de outro fabricante, realiza a análise rapidamente, porém ao custo de uma faixa de pressão curta. O conjunto P mensura a adequação da faixa de pressão, sendo 0 inadequado e 1 excelente. O conjunto TM indica o tempo morto, sendo, novamente, 0 inadequado e 1 excelente. Com base nas informações dos disponíveis chegou-se a conclusão que:

$$P = \left\{ \frac{0,8}{X} + \frac{0,3}{Y} \right\}, \quad TM = \left\{ \frac{0,5}{X} + \frac{0,9}{Y} \right\}$$

Insatisfeito com estas opções, você questiona aos fabricantes se não existem outros modelos de analisadores em seus catálogos, que possam representar uma solução de compromisso, i.e., que tanto a faixa de pressão quanto o tempo morto sejam adequados para sua aplicação. Cada fabricante apresentou um outro modelo, resultando nos novos conjuntos:

$$P = \left\{ \frac{0,8}{X} + \frac{0,3}{Y} + \frac{0,7}{X_2} + \frac{0,6}{Y_2} \right\}, \quad TM = \left\{ \frac{0,5}{X} + \frac{0,9}{Y} + \frac{0,55}{X_2} + \frac{0,6}{Y_2} \right\}$$

- Calcule $P \cup TM$. Qual a interpretação deste resultado?
- Calcule $P \cap TM$. Qual a interpretação deste resultado?
- Calcule $\bar{P} \cap \overline{TM}$. Qual a interpretação deste resultado? **Solução:**

a) Para cada elemento, o maior valor de pertinência nos conjuntos P e TM prevalecerá. Este resultado indica o valor mais adequado para um dado analisador em qualquer categoria:

$$P \cup TM = \left\{ \frac{0,8}{X} + \frac{0,9}{Y} + \frac{0,7}{X_2} + \frac{0,6}{Y_2} \right\}.$$

Nota-se que o analisador Y é o maior destaque em uma das categorias e o analisador Y_2 não prevalece em uma das categorias.

Exercício Resolvido

b) Alternativamente, o menor valor prevalecerá nesta operação. Assim, mostra-se qual analisador é adequado em ambas categorias, simultaneamente.

$$P \cap TM = \left\{ \frac{0,5}{X} + \frac{0,3}{Y} + \frac{0,55}{X_2} + \frac{0,6}{Y_2} \right\}$$

c) Neste caso será necessário aplicar o operador de complemento, seguido pelo de interseção. Este resultado mostra os analisadores que não são adequados para faixa de pressão e tempo morto, simultaneamente:

$$\bar{P} = \left\{ \frac{0,2}{X} + \frac{0,7}{Y} + \frac{0,3}{X_2} + \frac{0,4}{Y_2} \right\}, \quad \overline{TM} = \left\{ \frac{0,5}{X} + \frac{0,1}{Y} + \frac{0,45}{X_2} + \frac{0,4}{Y_2} \right\}$$

$$\bar{P} \cap \overline{TM} = \left\{ \frac{0,2}{X} + \frac{0,1}{Y} + \frac{0,3}{X_2} + \frac{0,4}{Y_2} \right\}$$

Aproveitando o exemplo apresentado com os conjuntos QUENTE e FRIO nesta aula, é fácil de se ver que os dois conjuntos são complementares (FRIO é o complemento de QUENTE, e vice-versa). Além disso, as Figuras 2.5 e 2.6 apresentam, respectivamente, a união dos dois conjuntos, e a interseção dos dois conjuntos.

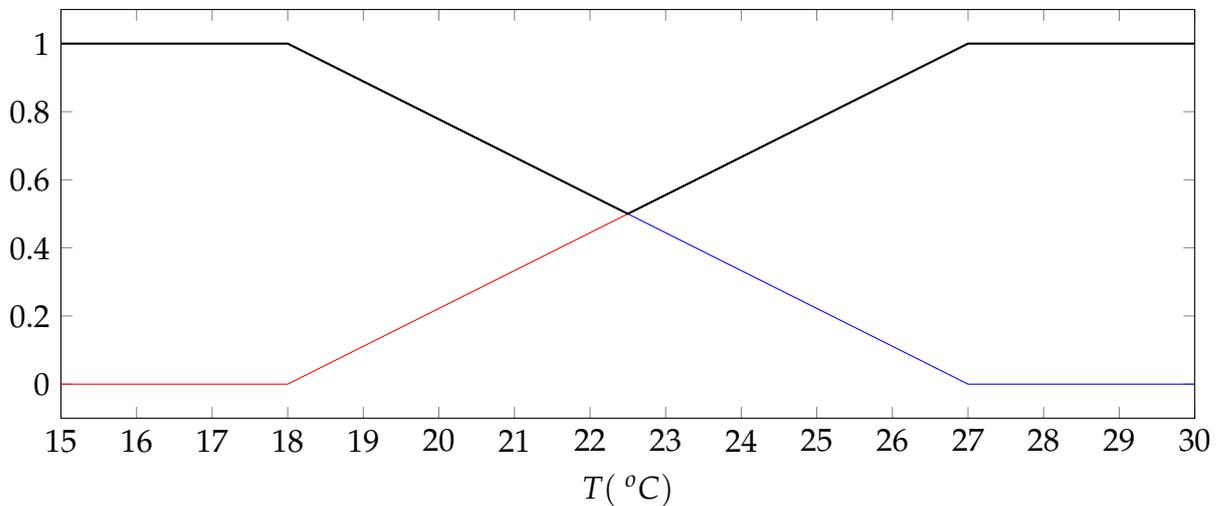


Figura 2.5: União dos conjuntos de dias frios e quentes (representada em preto).

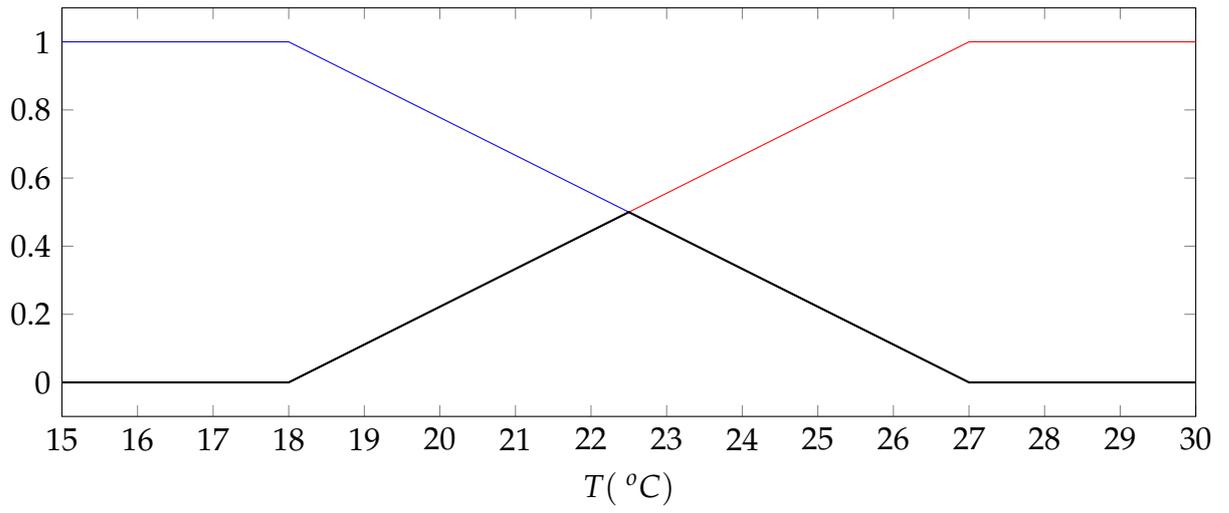


Figura 2.6: Interseção dos conjuntos de dias frios e quentes (representada em preto).

Exercício 2.2

Alguns formatos de funções de pertinência são muito comuns (devido à sua simplicidade e propriedades interessantes). Nesse exercício, vamos focar em três desses formatos: as funções triangulares, as funções trapezoidais e as funções gaussianas.

As funções triangulares são definidas por três parâmetros e, c, d de modo que

$$\mu(x) = \begin{cases} 0 & \text{se } x < e \\ \frac{x - e}{c - e} & \text{se } e \leq x < c \\ \frac{d - x}{d - c} & \text{se } c \leq x < d \\ 0 & \text{se } x \geq d \end{cases}$$

As funções trapezoidais são definidas por quatro parâmetros e, c_1, c_2, d de modo que

$$\mu(x) = \begin{cases} 0 & \text{se } x < e \\ \frac{x - e}{c_1 - e} & \text{se } e \leq x < c_1 \\ 1 & \text{se } c_1 \leq x < c_2 \\ \frac{d - x}{d - c_2} & \text{se } c_2 \leq x < d \\ 0 & \text{se } x \geq d \end{cases}$$

As funções gaussianas são definidas por dois parâmetros c, σ de modo que

$$\mu(x) = e^{\left(\frac{-(x - c)^2}{2\sigma^2}\right)}$$

Implemente as três funções de pertinência acima, bem como os operadores de Zadeh de complemento, união e interseção. Teste em um problema exemplo e gere os gráficos das funções de pertinência para verificar se a implementação está correta.

Exercício 2.3: Adaptado de Bezdek 1993

Considere que todos os líquidos existentes sejam U , o universo de discurso, e seja dado o conjunto $L = \{ \text{ todos líquidos potáveis} \}$. Suponha que você esteja perdido em um deserto há 3 dias, sem acesso a quaisquer líquidos. De repente, você se depara com um acampamento abandonado. Nele há duas garrafas, sem que seja possível identificar o líquido em seu interior.

a) Na garrafa "A" você pode ler no rótulo $\mu_L(A) = 0,99$. Já na garrafa "B" o rótulo indica $p_L(B) = 0,99$, sendo que $p_Y(x)$ indica a probabilidade do elemento x pertencer ao conjunto Y . Aqui adota-se a noção de pertinência clássica. Em outras palavras, uma a cada 100 garrafas marcadas com este rótulo pode conter qualquer líquido não potável, inclusive um conteúdo fatal como, por exemplo, ácido clorídrico concentrado. Diante destas únicas opções, de qual garrafa você vai beber?

b) Suponha agora o mesmo cenário, porém os rótulos são diferentes. Lê-se $\mu_L(A) = 0,50$ e $p_L(B) = 0,50$. Você mudaria sua escolha?

c) Após examinar o conteúdo, descobre-se que a garrafa "A" possui cerveja, porém a data de validade foi expirada em 1 dia. Qual valor você iria atribuir para $\mu_L(A)$?

d) Após examinar o conteúdo, descobre-se que a garrafa "B" possui ácido clorídrico, porém diluído em 3 partes para 7 de água. Qual valor você iria atribuir para $p_L(B)$?

Exercício 2.4

Mostre que os operadores de Zadeh quando aplicados em conjuntos *Crisp* resultam no mesmo resultado apresentado na Tabela 2.1.

Aula 3

Operações em Conjuntos Nebulosos

Nesta aula, vemos algumas outras formas de se definir as operações de complemento, união e interseção para conjuntos nebulosos.

Apesar de termos apresentado os operadores de Zadeh na última aula, outros operadores também podem ser definidos para estender as operações de complemento, união e interseção dos conjuntos clássicos para os conjuntos nebulosos.

Complemento Nebuloso

Definição 3.1: Operadores de Complemento

Para que uma função $c(\mu_A(x)) = \mu_{\bar{A}}(x)$ seja considerada um operador de complemento nebuloso, ela deve atender as seguintes propriedades

1. $c(0) = 1$ e $c(1) = 0$ (condições limite)
2. $\forall a, b \in [0, 1]$, se $a < b$ então $c(a) \geq c(b)$ (condição de não crescimento)

Outros complementos comuns na literatura são o complemento de Sugeno e o complemento de Yager.

Definição 3.2: Complemento de Sugeno

$$c_\lambda(a) = \frac{1 - a}{1 + \lambda a}$$

com algum $\lambda \in (-1, \infty)$, e no caso em que $\lambda = 0$ tem-se o complemento de Zadeh.

Definição 3.3: Complemento de Yager

$$c_w(a) = (1 - a^w)^{1/w}$$

com algum $w \in (0, \infty)$, e no caso em que $\lambda = 1$ tem-se o complemento de Zadeh.

Para ilustrar os complementos de Sugeno e de Yager, utilizamos a função de pertinência apresentada na Figura 3.1 e calculamos o seu complemento nas Figuras 3.2 e 3.3.

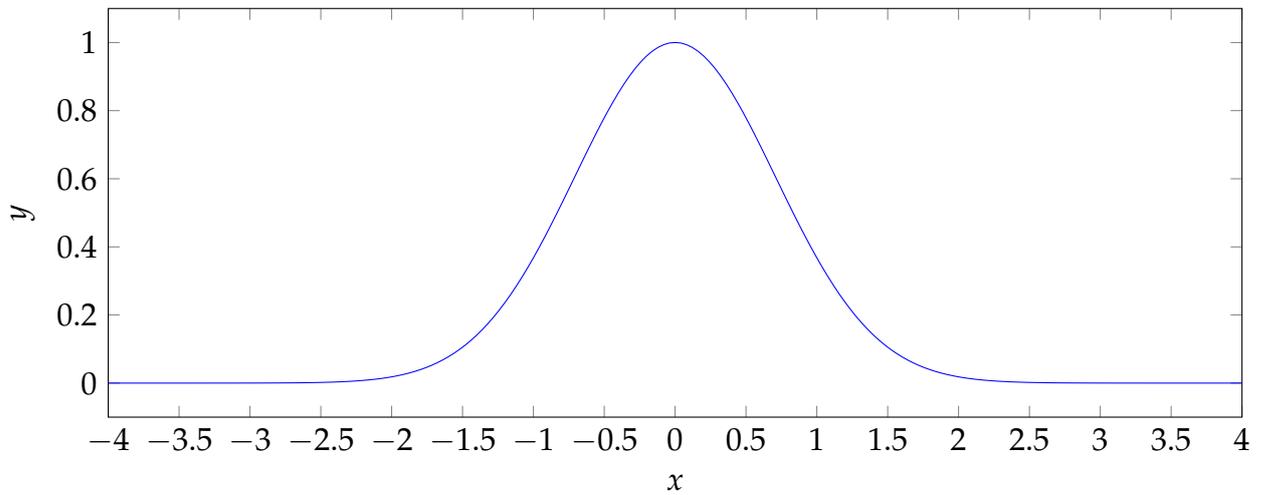


Figura 3.1: Exemplo de uma função de pertinência gaussiana.

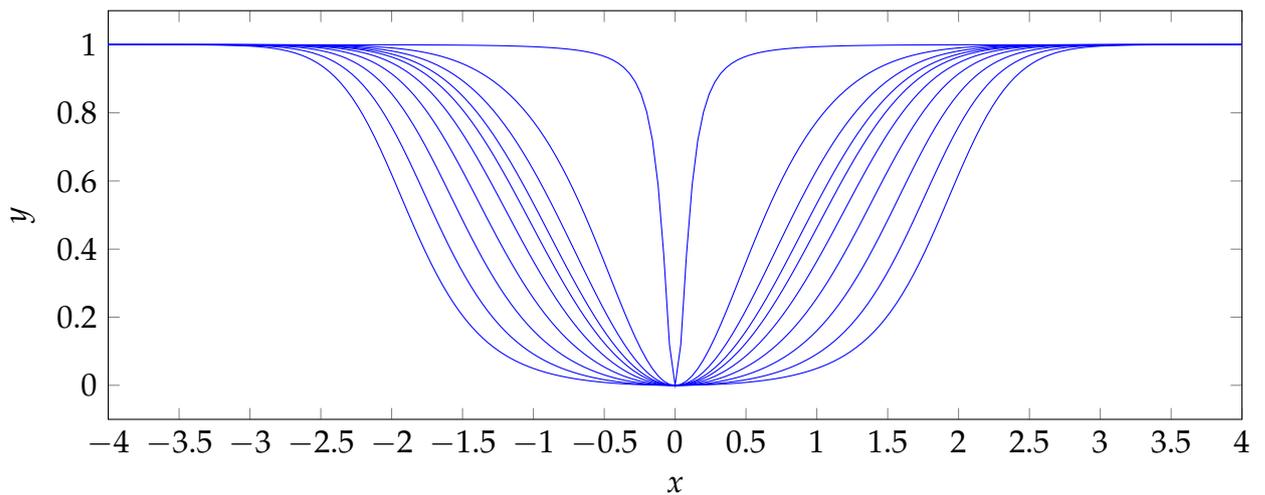


Figura 3.2: Exemplo de complementos de Sugeno aplicados à função de pertinência da Figura 3.1. Os valores de λ aumentam à medida que vamos das curvas superiores para as inferiores.

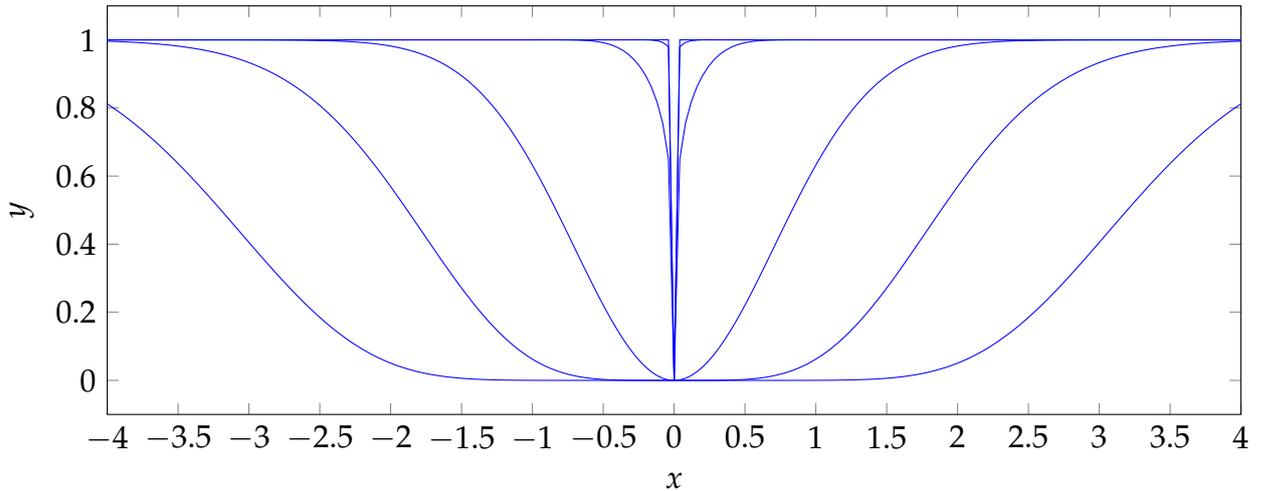


Figura 3.3: Exemplo de complementos de Yager aplicados à função de pertinência da Figura 3.1. Os valores de w aumentam à medida que vamos das curvas inferiores para as superiores.

União Nebulosa

Os operadores de união nebulosa normalmente recebem o nome de **s-normas**.

Definição 3.4: s-norma

Para que uma função $s(\mu_A(x), \mu_B(x)) = \mu_{A \cup B}(x)$ seja considerada uma s-norma, ela deve atender as seguintes propriedades

1. $s(1, 1) = 1$ e $s(0, a) = s(a, 0) = a$ (condições limite)
2. $s(a, b) = s(b, a)$ (comutatividade)
3. Se $a \leq a'$ e $b \leq b'$, então $s(a, b) \leq s(a', b')$ (condição de não-decrescimento)
4. $s(s(a, b), c) = s(a, s(b, c))$ (associatividade)

Outras s-normas comuns na literatura são apresentadas a seguir.

Definição 3.5: s-norma de Dombi

$$s_\lambda(a, b) = \frac{1}{1 + \left[\left(\frac{1}{a} - 1 \right)^{-\lambda} + \left(\frac{1}{b} - 1 \right)^{-\lambda} \right]^{-1/\lambda}}$$

com algum $\lambda \in (0, \infty)$.

Definição 3.6: s-norma de Dubois-Prade

$$s_{\alpha}(a, b) = \frac{a + b - ab - \min(a, b, 1 - \alpha)}{\max(1 - a, 1 - b, \alpha)}$$

com algum $\alpha \in [0, 1]$.

Definição 3.7: s-norma de Yager

$$s_w(a, b) = \min(1, (a^w + b^w)^{1/w})$$

com algum $w \in (0, \infty)$.

Definição 3.8: Soma drástica

$$s_{d_s}(a, b) = \begin{cases} a & \text{se } b = 0, \\ b & \text{se } a = 0, \\ 1 & \text{caso contrário.} \end{cases}$$

Definição 3.9: Soma de Einstein

$$s_{e_s}(a, b) = \frac{a + b}{1 + ab}$$

Definição 3.10: Soma Probabilística

$$s_s(a, b) = a + b - ab$$

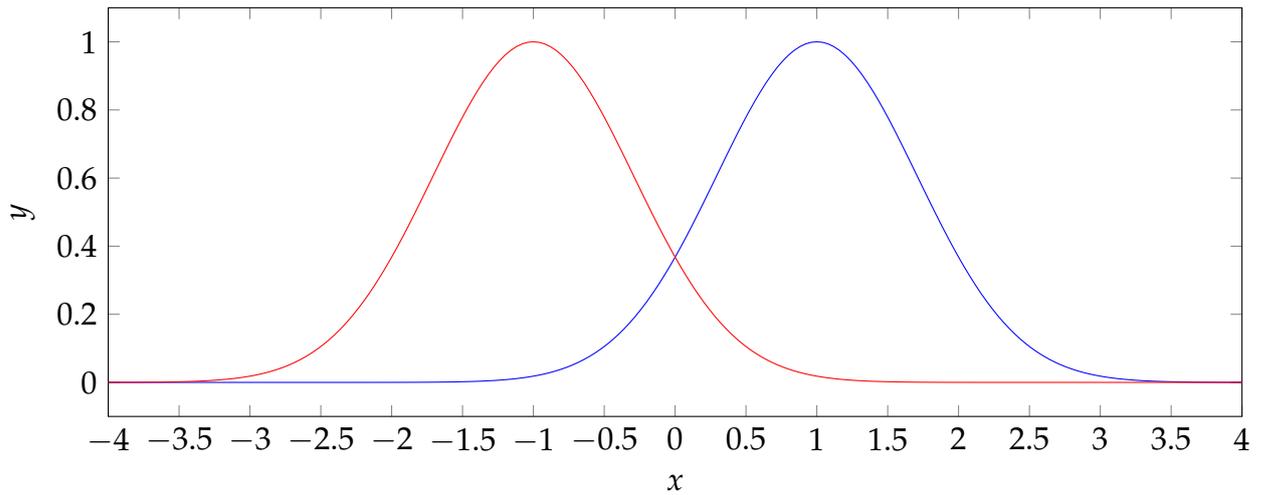


Figura 3.4: Exemplo de duas funções de pertinência gaussianas.

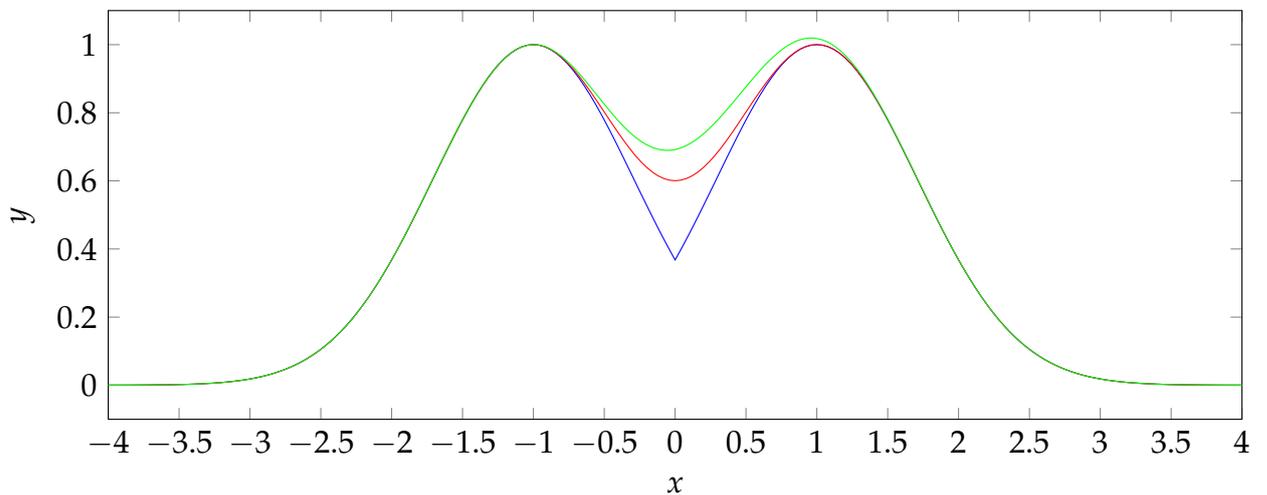


Figura 3.5: Exemplos de s-normas aplicadas às funções de pertinência da Figura 3.4. A curva em azul representa o operador de Zadeh (máximo). A curva em vermelho representa o operador de soma probabilística. A curva em verde representa o operador de soma de Einstein.

Interseção Nebulosa

Os operadores de interseção nebulosa normalmente recebem o nome de **t-normas**.

Definição 3.11: t-norma

Para que uma função $t(\mu_A(x), \mu_B(x)) = \mu_{A \cap B}(x)$ seja considerada uma t-norma, ela deve atender as seguintes propriedades

1. $t(0,0) = 0$ e $t(1,a) = t(a,1) = a$ (condições limite)
2. $t(a,b) = t(b,a)$ (comutatividade)
3. Se $a \leq a'$ e $b \leq b'$, então $t(a,b) \leq t(a',b')$ (condição de não-decrescimento)
4. $t(t(a,b),c) = t(a,t(b,c))$ (associatividade)

Outras t-normas comuns na literatura são apresentadas a seguir.

Definição 3.12: t-norma de Dombi

$$t_\lambda(a,b) = \frac{1}{1 + \left[\left(\frac{1}{a} - 1 \right)^\lambda + \left(\frac{1}{b} - 1 \right)^\lambda \right]^{1/\lambda}}$$

com algum $\lambda \in (0, \infty)$.

Definição 3.13: t-norma de Dubois-Prade

$$t_\alpha(a,b) = \frac{ab}{\max(a,b,\alpha)}$$

com algum $\alpha \in [0, 1]$.

Definição 3.14: t-norma de Yager

$$t_w(a,b) = 1 - \min(1, ((1-a)^w + (1-b)^w)^{1/w})$$

com algum $w \in (0, \infty)$.

Definição 3.15: Produto drástico

$$t_{d_p}(a,b) = \begin{cases} a, & \text{se } b = 1 \\ b, & \text{se } a = 1 \\ 0, & \text{caso contrário.} \end{cases}$$

Definição 3.16: Produto de Einstein

$$t_{ep}(a, b) = \frac{ab}{2 - (a + b - ab)}.$$

Definição 3.17: t-norma de Łukasiewicz

$$t_L(a, b) = \max(a + b - 1, 0)$$

Definição 3.18: Produto algébrico

$$t_p(a, b) = ab$$

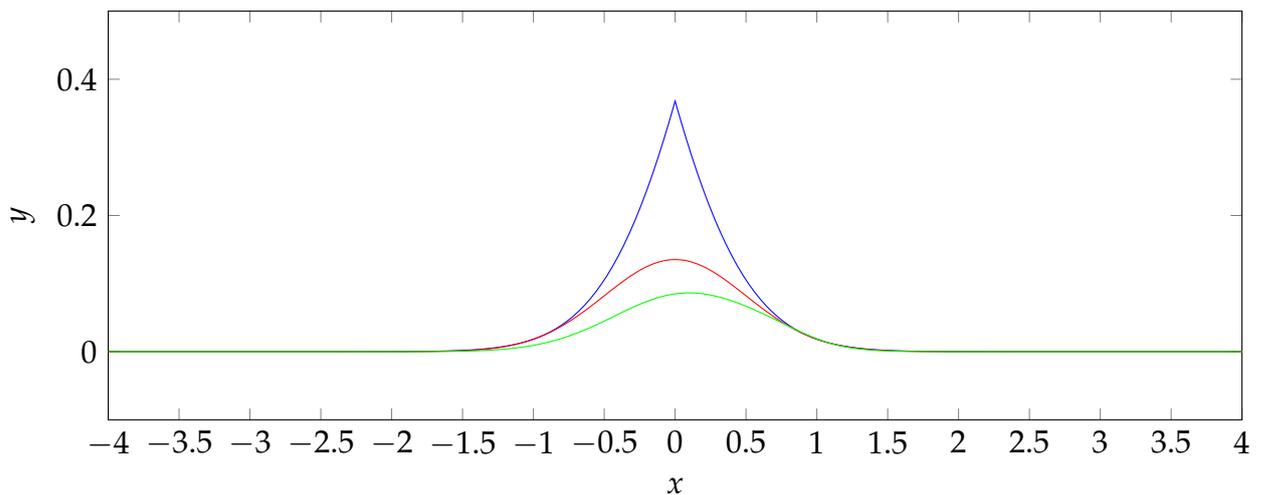


Figura 3.6: Exemplos de t-normas aplicadas às funções de pertinência da Figura 3.4. A curva em azul representa o operador de Zadeh (mínimo). A curva em vermelho representa o operador de produto. A curva em verde representa o operador de produto de Einstein.

Classe Associada

Dizemos que um complemento nebuloso, uma s-norma e uma t-norma formam uma classe associada quando os três juntos satisfazem a Lei de DeMorgan. Ou seja, quando

$$c(s(a, b)) = t(c(a), c(b)) \rightarrow (\overline{AB} = \bar{A} + \bar{B})$$

Alguns exemplos de classes associadas:

- Operadores definidos por Zadeh

- Complemento de Zadeh com as s-norma e t-norma de Yager
- * Complemento de Zadeh com a soma probabilística (s-norma) e o produto (t-norma)

Comentários sobre Operadores

É interessante notar as condições de contorno impostas às t-normas pelas propriedades da Definição 3.11. A condição $t(\alpha, 1) = \alpha$, por exemplo, implica que $t(1, 1) = 1$ e $t(0, 1) = 0$, para quaisquer t-normas. Pela comutatividade, $t(0, 1) = 0$ e $t(1, 0) = 0$. A partir da condição de não decrescimento, $t(1, 0) = 0$ garante que $t(0, 0) = 0$, fato explicitamente mencionado nas condições de limite. Logo, quando restritas aos valores binários 0 e 1, todas t-normas reproduzem a tabela verdade da operação clássica de interseção. O análogo pode ser dito com relação às s-normas e a união.

Uma outra forma de perceber as diferenças proporcionadas por diferentes operadores é através da análise gráfica do mapeamento $f : [0, 1]^2 \rightarrow [0, 1]$. i.e., traçando no plano $x - y$ todas as possíveis combinações de valores de pertinência, como entradas α e β , e no eixo z o valor resultante da aplicação do operador $f(\alpha, \beta)$. Estes gráficos para as t-normas mínimo (interseção de Zadeh), produto algébrico e de Łukasiewicz são mostrados na Figura 3.7.

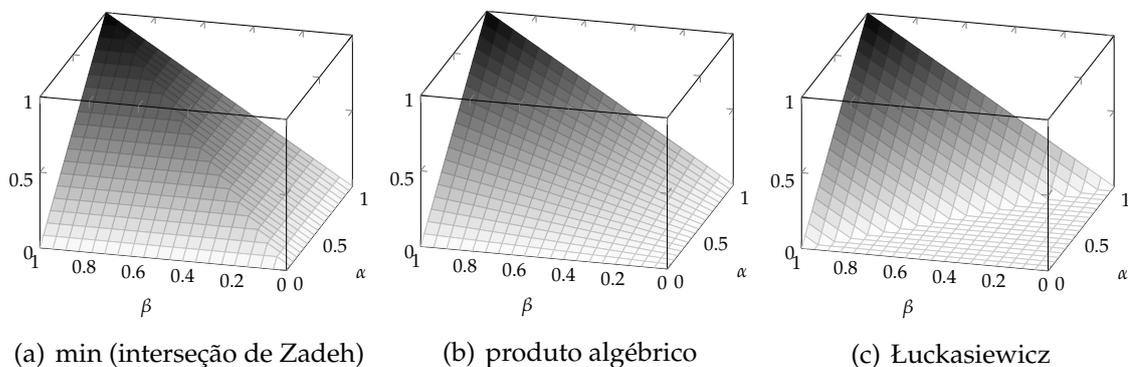


Figura 3.7: Comparação do mapeamento $t(\alpha, \beta)$ para diferentes t-normas.

Note que, devido às condições de contorno mencionadas anteriormente, nos valores limite do semiplano $[0, 1]^2$ todas t-normas produzem o mesmo resultado. É interessante notar por estes três exemplos que o espectro das t-normas é bastante amplo, internamente, entretanto.

Outro aspecto digno de nota, apenas a interseção de Zadeh é considerada idempotente, i.e., satisfaz a propriedade $t(\alpha, \alpha) = \alpha, \forall \alpha \in [0, 1]$. A princípio esta parece ser uma propriedade que deveria ser estendida a todas t-normas, mas em alguns casos pode não ser interessante, como no exemplo a seguir.

Exemplo 3.1. Um comprador está em dúvida entre dois computadores, A e B, similares em diversos aspectos. Ele então concentra sua decisão em dois aspectos: se o preço do computador é bom e se o design do computador é moderno. Após análise cuidadosa ele chega aos seguintes graus de veracidade nas afirmativas:

	afirmativa	valor verdade
ϕ_1	O preço do computador A é bom	0,7
ϕ_2	O design do computador A é moderno	0,7
ϕ_3	O preço do computador B é bom	0,7
ϕ_4	O design do computador B é moderno	0,9

Intuitivamente, pensamos que o computador B será a melhor escolha, afinal os preços são igualmente bons, porém nos parece que o design de B é mais interessante. Logo, ele irá escolher o computador $x \in \{A, B\}$ que resultar no maior valor para a frase “O preço do computador x é bom E o design do computador x é moderno”, o que matematicamente se traduz como escolher B caso $t(\phi_3, \phi_4) > t(\phi_1, \phi_2)$.

Contudo, ao escolher o operador de interseção de Zadeh (mínimo), o valor retornado para ambos os computadores é o mesmo, o que parece contra intuitivo, mas ocorreu devido a característica de idempotência:

$$\min(\phi_1, \phi_2) = 0,7 = \min(\phi_3, \phi_4)$$

Ao usar uma t-norma não idempotente, por exemplo o produto algébrico, tem-se:

$$t_p(\phi_1, \phi_2) = (0,7)(0,7) = 0,49 < t_p(\phi_3, \phi_4) = (0,7)(0,9) = 0,63$$

Logo, para as outras t-normas o resultado irá favorecer o computador B.

Exercício 3.1

Implemente os operadores das três classes associadas apresentadas acima.

Complementos

- Complemento de Zadeh

S-normas (união)

- União de Zadeh
- S-norma de Yager
- Soma probabilística

T-normas (interseção)

- Interseção de Zadeh
- T-norma de Yager
- Produto

Considere um exemplo com dois conjuntos nebulosos definidos por funções de pertinência triangulares, com os parâmetros $[0, 1, 2]$ e $[1, 2, 3]$, e gere os gráficos dos conjuntos nebuloso de união e interseção comparando os diferentes operadores implementados.

Aula 4

Relações Nebulosas

Nesta aula, veremos um pouco sobre projeções, produto cartesiano, composição de relações nebulosas e o princípio da extensão.

Produto Cartesiano

O produto cartesiano entre dois conjuntos clássicos U e V é dado pelo conjunto de pares ordenados (u, v) tal que $u \in U$ e $v \in V$, ou seja

$$U \times V = \{(u, v) | u \in U, v \in V\}$$

Note que $U \times V \neq V \times U$.

O produto cartesiano entre n conjuntos U_1, U_2, \dots, U_n é dado pelo conjunto de n -tuplas ordenadas (u_1, u_2, \dots, u_n) com $u_1 \in U_1, u_2 \in U_2, \dots, u_n \in U_n$.

Relação

Uma relação é um subconjunto do produto cartesiano. Dizemos que uma relação é binária quando ela é um subconjunto do produto cartesiano de dois conjuntos.

Como uma relação é um conjunto, ela admite uma função indicadora/pertinência.

Exemplo 4.1. Seja $U = \mathbb{R}$ e $V = \mathbb{R}$, então $U \times V = \mathbb{R}^2$. Uma relação é um subconjunto qualquer de \mathbb{R}^2 , por exemplo

$$Q(U, V) = \{(x, y) | x \in U, y \in V, x \geq y\}$$

que pode ser representada pela função indicadora

$$\mu_Q(x, y) = \begin{cases} 1, & \text{se } x \geq y \\ 0, & \text{caso contrário} \end{cases}$$

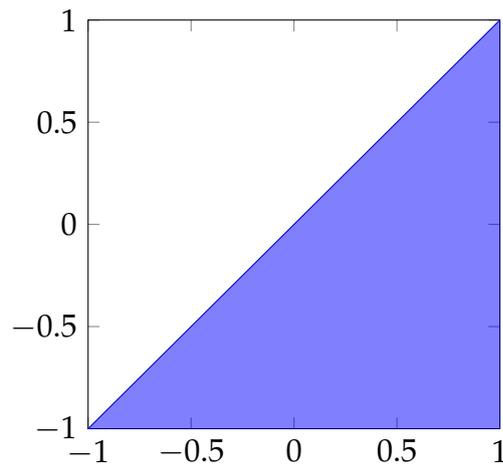


Figura 4.1: Exemplo de relação no \mathbb{R}^2 apresentado no exemplo 4.1.

e representada pela Figura 4.1.

Exemplo 4.2. Seja $U = \{1, 2\}$, $V = \{1, 4\}$, $U \times V = \{(1, 1), (1, 4), (2, 1), (2, 4)\}$. Uma relação possível é o subconjunto de $U \times V$ em que o primeiro elemento é menor do que o segundo

$$Q(U, V) = \{(1, 4), (2, 4)\}$$

No caso de conjuntos discretos, como o deste exemplo, a função indicadora/pertinência pode ser representada como uma “matriz relacional”

	1	4
1	0	1
2	0	1

Dado que relações são conjuntos também, é possível estabelecer os operadores entre relações. Logo, as propriedades que valem para os operadores clássicos de conjuntos também se estendem para relações. Uma característica interessante de relações é a chamada composição, que será detalhada na sequência.

Composição

Sejam dadas uma relação R , que mapeia elementos do universo X para o universo Y , e uma relação S , que mapeia elementos do universo Y para o universo Z . Uma questão pertinente a ser colocada: é possível obter uma relação, por exemplo T , que relaciona diretamente os mesmos elementos de X e Z ? Obter esta nova relação é possível por meio do operador Composição.

Dessa forma, como relações são conjuntos que podem ser descritos por funções indicadoras/de pertinência, podemos introduzir o conceito de relações nebulosas. Su-

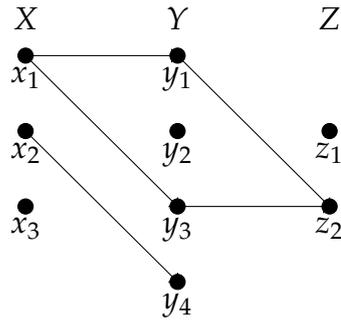


Figura 4.2: Diagrama Sagittal mostrando a relação entre elementos de três universos.

ponha o Diagrama Sagittal mostrado na Figura 4.2. Este diagrama pode indicar a existência de estradas federais conectando municípios de 3 estados. As relações para este diagrama são dadas por $R = \{(x_1, y_1), (x_1, y_3), (x_2, y_4)\}$ e $S = \{(y_1, z_2), (y_3, z_2)\}$. Percebemos que existem duas rotas relacionando as cidades x_1 do Estado X e z_2 do estado Z, a saber: $x_1 - y_1 - z_2$ ou $x_1 - y_3 - z_2$

Uma forma de estabelecer a composição da relação é dada por:

$$T = R \circ S$$

$$\mu_T(x, z) = \bigvee_{y \in Y} (\mu_R(x, y) \wedge \mu_S(y, z)).$$

Usando a forma de matrix relacional, o resultado da composição neste caso é:

$$\begin{array}{c|cccc}
 \mathbf{R} & y_1 & y_2 & y_3 & y_4 \\
 \hline
 x_1 & 1 & 0 & 1 & 0 \\
 x_2 & 0 & 0 & 0 & 1 \\
 x_3 & 0 & 0 & 0 & 0
 \end{array}
 \quad e \quad
 \begin{array}{c|cc}
 \mathbf{S} & z_1 & z_2 \\
 \hline
 y_1 & 0 & 1 \\
 y_2 & 0 & 0 \\
 y_3 & 0 & 1 \\
 y_4 & 0 & 0
 \end{array}$$

$$\mu_T(x_1, z_1) = \max[\min(1, 0), \min(0, 0), \min(1, 0), \min(0, 0)] = 0,$$

$$\mu_T(x_1, z_2) = \max[\min(1, 1), \min(0, 0), \min(1, 1), \min(0, 0)] = 1,$$

\therefore

$$\begin{array}{c|cc}
 \mathbf{T} & z_1 & z_2 \\
 \hline
 x_1 & 0 & 1 \\
 x_2 & 0 & 0 \\
 x_3 & 0 & 0
 \end{array}$$

Exercício 4.1

Mostre que para conjuntos clássicos/*Crisp*, R e S , a composição max-min:

$$\mu_T(x, z) = \bigvee_{y \in Y} (\mu_R(x, y) \wedge \mu_S(y, z)).$$

produz os mesmos resultados da composição max-produto:

$$\mu_T(x, z) = \bigvee_{y \in Y} (\mu_R(x, y) \bullet \mu_S(y, z)).$$

onde “ \bullet ” indica o produto aritmético.

Relação Nebulosa

A desvantagem da relação clássica/*Crisp* entre conjuntos reside no fato de que os elementos estão relacionados ou não estão relacionados. Existem algumas situações em que essa visão não se enquadra adequadamente, conforme o exemplo a seguir:

Exemplo 4.3. Seja dois conjuntos $A = \{\text{Belo Horizonte, Paris, Tokyo}\}$ e $B = \{\text{Natal, Tokyo}\}$. Queremos definir o conceito relacional “muito distante” entre estas cidades. Relações clássicas resultam em uma representação pouco informativa para este exemplo, haja vista que são cidades separadas por milhares de km entre si. Contudo, “muito distante” ainda pode ser qualificado, tendo em vista que é possível chegar a Natal partindo de Belo Horizonte de carro, com certo esforço, e um voo entre Natal e Paris é mais demorado que entre Natal e Tokyo. Portanto, uma alternativa seria a matrix relacional fuzzy a seguir, na qual se usa um número no intervalo $[0, 1]$ para representar o grau de pertinência ao

	Natal	Tokyo
BH	0,15	1
Paris	0,40	0,6
Tokyo	0,85	0

Este exemplo indica que é necessário estender o conceito de relações para aproximar aos conceitos mais imprecisos e vagos que lidamos habitualmente.

Definição 4.1: Relação Nebulosa

Uma relação nebulosa é um conjunto nebuloso definido no produto cartesiano de dois conjuntos clássicos/crisp

$$Q = \{((u_1, u_2, \dots, u_n), \mu_Q(u_1, u_2, \dots, u_n)) \mid (u_1, u_2, \dots, u_n) \in U_1 \times U_2 \times \dots \times U_n\}$$

com μ_Q uma função de $U_1 \times U_2 \times \dots \times U_n$ para $[0, 1]$.

Além disso, dizemos que uma relação nebulosa é uma relação binária quando ela é definida no produto cartesiano de dois conjuntos.

Exemplo 4.4. Sejam $U = \mathbb{R}, V = \mathbb{R}, U \times V = \mathbb{R}^2$. A relação nebulosa “ x é aproximadamente igual a y ” pode ser descrita pela função de pertinência $\mu_Q = e^{-(x-y)^2}$.

Exercício 4.2

Sejam $U = \mathbb{R}, V = \mathbb{R}, U \times V = \mathbb{R}^2$. Defina a relação nebulosa $\mu_M(\cdot)$, traduzindo o conceito “ x é muito maior que y ”.

Definição 4.2: Produto Cartesiano de Conjuntos Nebulosos

Sejam A_1, \dots, A_n conjuntos nebulosos em U_1, \dots, U_n . O produto cartesiano de A_1, \dots, A_n ($A_1 \times \dots \times A_n$) é uma relação nebulosa em $U_1 \times \dots \times U_n$ cuja função de pertinência é dada por

$$\mu_{A_1 \times \dots \times A_n}(u_1, \dots, u_n) = \mu_{A_1}(u_1) \star \dots \star \mu_{A_n}(u_n)$$

em que \star é qualquer t-norma.

O produto cartesiano clássico busca estabelecer composições com elementos que pertencem aos conjuntos, considerando que todos estão relacionados. Analogamente, o produto cartesiano nebuloso busca estabelecer também relação de todos os elementos, por isso adota-se uma t-norma. Afinal, se um elemento possui pouca pertinência ao conjunto de origem, este baixo grau deve ser “transportado” para o produto cartesiano.

Operações sobre Relações Nebulosas

Além de relações, podemos definir também projeções de uma relação e extensões cilíndricas.

Seja uma relação Q definida no produto cartesiano $U \times V$. A projeção de Q em U é o subconjunto de U que é utilizado na relação.

A extensão cilíndrica de um conjunto Q_1 definido em U para o produto cartesiano $U \times V$ é definido pela relação

$$Q_{1e} = \{(x, y) \in U \times V | x \in Q_1\}$$

Essas definições são ilustradas na Figura 4.3.

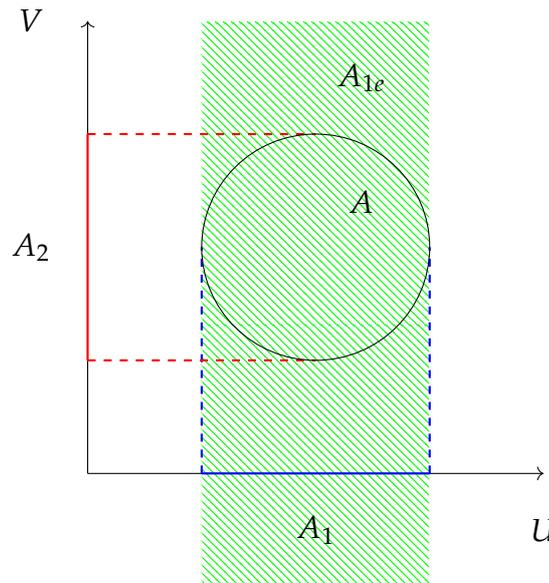


Figura 4.3: Exemplos de projeção e extensão cilíndrica. O círculo em preto representa uma relação A definida no produto cartesiano $U \times V$. O conjunto representado em azul, A_1 , é a projeção do conjunto A em U . O conjunto representado em vermelho, A_2 , é a projeção do conjunto A em V . O conjunto representado em verde, A_{1e} , é a extensão cilíndrica do conjunto A_1 no produto cartesiano $U \times V$.

Projeção de uma Relação Nebulosa

Uma generalização possível da projeção clássica para relações nebulosas é a seguinte

Definição 4.3: Projeção de uma Relação Nebulosa

Seja Q uma relação nebulosa em $U_1 \times U_2 \times \dots \times U_n$ e os índices $\{i_1, \dots, i_k\}$ uma subsequência de $\{1, \dots, n\}$. A projeção de Q em $U_{i_1} \times \dots \times U_{i_k}$ é uma relação nebulosa Q_P em $U_{i_1} \times \dots \times U_{i_k}$ definida pela pertinência

$$\mu_{Q_P}(u_{i_1}, \dots, u_{i_k}) = \max_{\text{índices que não fazem parte da projeção}} \mu_Q(u_1, \dots, u_n).$$

Numa relação binária,

$$\mu_{Q_1}(x) = \max_{y \in V} \mu_Q(x, y).$$

Extensão cilíndrica nebulosa

Uma generalização possível da extensão cilíndrica clássica para relações nebulosas é a seguinte

Definição 4.4: Extensão Cilíndrica Nebulosa

Seja Q_P uma relação fuzzy em $U_{i_1} \times \cdots \times U_{i_k}$. A extensão cilíndrica de Q_P para $U_1 \times \cdots \times U_n$ é uma relação nebulosa em $U_1 \times \cdots \times U_n$ definida pela pertinência

$$\mu_{Q_{Pe}}(u_1, \dots, u_n) = \mu_{Q_P}(u_{i_1}, \dots, u_{i_k}).$$

No caso de um conjunto nebuloso para um produto cartesiano de duas variáveis,

$$\mu_{Q_{1e}}(x, y) = \mu_{Q_1}(x).$$

O seguinte lema, apresentado em Wang 1999, Lemma 4.1, pode ser útil para entendermos a relação entre uma relação nebulosa e suas projeções.

Lema 4.1

Toda relação nebulosa está contida no produto cartesiano de suas projeções.

Composição Nebulosa

Sejam duas relações $P(U, V)$ e $Q(V, W)$, a composição $P \circ Q$ é uma relação em $U \times W$ tal que $(x, z) \in P \circ Q$ se, e somente se, existe pelo menos um $y \in V$ tal que $(x, y) \in P$ e $(y, z) \in Q$.

Usando funções de pertinência podemos descrever a composição de relações da seguinte forma

Definição 4.5: Composição de Relações Nebulosas

$P \circ Q$ é a composição de P e Q se, e somente se

$$\mu_{P \circ Q}(x, z) = \max_{y \in V} \mu_P(x, y) \star \mu_Q(y, z) \quad \forall (x, z) \in U \times W$$

As duas composições mais comuns são dadas utilizando o min (composição max-min) ou o produto (composição max-produto) como t-norma.

Exemplo 4.5. Uma forma usual de inferir sobre o grau de maturação de frutos e legumes consiste em observar sua coloração. Suponha que se deseja relacionar a cor do tomate com seu estado de maturação. Para isso foram definidos os conjuntos $C = \{\text{verde,}$

amarelo, vermelho} e $M = \{\text{n\~ao-maduro, meio maduro, maduro}\}$. Uma formulação fuzzy para a relação $R_1(C, M)$ pode ser estabelecida na seguinte matriz:

R_1	N\~ao-maduro	Meio maduro	Maduro
Verde	1	0,5	0
Amarelo	0,3	1	0,4
Vermelho	0	0,2	1

Por outro lado, sabe-se que o grau de maturação tem influência no sabor dos frutos e legumes. Neste caso foi definido outro conjunto $S = \{\text{azedo, insípido, doce}\}$ e a relação $R_2(M, S)$ pela matriz:

R_2	Azedo	Insípido	Doce
N\~ao-maduro	1	0,2	0
Meio maduro	0,7	1	0,3
Maduro	0	0,7	1

Por meio da composição $T = R_1 \circ R_2$ é possível obter uma relação entre cor e sabor do tomate, conforme:

T	Azedo	Insípido	Doce
Verde	1	0,5	0,3
Amarelo	0,7	1	0,4
Vermelho	0,2	0,7	1

Exercício 4.3

Três variáveis de interesse em MOSFETs são a quantidade de corrente que pode ser chaveada, a tensão que pode ser chaveada e o custo. As seguintes funções de pertinência foram desenvolvidas a partir de um catálogo de componentes hipotéticos:

$$\begin{aligned} \text{Corrente (ampere)} = I &= \left\{ \frac{0,4}{0,8} + \frac{0,7}{0,9} + \frac{1}{1} + \frac{0,8}{1,1} + \frac{0,6}{1,2} \right\} \\ \text{Tensão (volt)} = V &= \left\{ \frac{0,2}{30} + \frac{0,8}{45} + \frac{1}{60} + \frac{0,9}{75} + \frac{0,7}{90} \right\} \\ \text{Custo (real)} = C &= \left\{ \frac{0,4}{0,5} + \frac{1}{0,6} + \frac{0,8}{0,7} \right\} \end{aligned}$$

- Determine o produto Cartesiano fuzzy $P = V \times I$.
- Determine o produto Cartesiano fuzzy $T = I \times C$. Fisicamente, o que esta relação representa?
- Usando a composição max-min, determine $E = P \circ T$. Fisicamente, o que esta relação representa?
- Usando a composição max-produto, determine $E = P \circ T$.

Aula 5

Princípio da Extensão

Nesta aula, veremos mais um pouco sobre mapeamentos e será apresentado o princípio da extensão.

Funções representam mapeamentos entre variáveis de entrada e saída:

$$y = f(x)$$

Tipicamente, $f(\cdot)$ é uma forma analítica e a entrada é um valor determinístico x , resultando em uma entrada y também determinística. Contudo, o que ocorre quando a entrada x é uma variável ou conjunto fuzzy e a função propriamente dita também pode ser fuzzy? O princípio da extensão permite avaliar o quão fuzzy é a saída.

Novamente, vamos retomar o conceito de relação, apresentado no capítulo anterior. Seja um mapeamento (função) $f : X \rightarrow Y$. Associado a ele existe uma relação $R = (x, y) | y = f(x)$, no espaço cartesiano $X \times Y$ que possui uma função característica, a saber:

$$\mu_R(x, y) = \begin{cases} 1, & y = f(x) \\ 0, & y \neq f(x) \end{cases}$$

Logo, foi definido como elementos específicos x são mapeados para elementos específicos y . Deseja-se definir como conjuntos de elementos X são mapeados para conjuntos em Y . Para um subconjunto A no universo X , a imagem, indicada pelo conjunto B no universo Y , é obtida como $f(A) = y | \forall x \in A, y = f(x)$, definido o seguinte valor característico:

$$\mu_B(y) = \mu_{f(A)}(y) = \bigvee_{y=f(x)} \mu_A(x)$$

Exemplo 5.1. Suponha um conjunto *Crisp* $A = \{0, 1\}$, definido no universo $X = \{-2, -1, 0, 1, 2\}$. Considere o mapeamento $y = |4x| + 2$. Pelo mapeamento, $f : X \rightarrow Y$, os possíveis elementos no universo imagem são $Y = \{2, 6, 10\}$. Contudo, o subconjunto B , dado por $B = f(A)$, não irá possuir todos os elementos de Y . Para isso é necessário calcular a função característica de todos elementos em X que são mapeados para os mesmos elementos em Y :

$$\begin{aligned}\mu_B(2) &= \vee\{\mu_A(0)\} \vee \{1\} = 1 \\ \mu_B(6) &= \vee\{\mu_A(-1), \mu_A(1)\} = \vee\{0, 1\} = 1 \\ \mu_B(10) &= \vee\{\mu_A(-2), \mu_A(2)\} = \vee\{0, 0\} = 0\end{aligned}$$

Note que $2 \in Y$ só é mapeado por $0 \in X$. Já o elemento $6 \in Y$ pode ser mapeado partindo de dois elementos. Um destes elementos na origem pertencem ao conjunto A , mas o outro não. No caso de $10 \in Y$ também existem dois caminhos, mas nenhum deles faz parte de A . Portanto, o mapeamento resulta em:

$$B = \{2, 6\}.$$

Exemplo 5.2. Considere o mapeamento $y = x^2 + 4$, $X = \{-2, -1, 0, 1, 2\}$ e $Y = \{0, 1, 2, \dots, 10\}$. O conjunto A é o mesmo do exemplo anterior e deseja-se obter B , o conjunto imagem. Neste caso, a relação *Crisp* deste mapeamento pode ser expressa como:

$\mu_R(x, y)$	1	2	3	4	5	6	7	8	9	10
-2	0	0	0	0	0	0	0	1	0	0
-1	0	0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0
2	0	0	0	0	0	0	0	1	0	0

A imagem B pode ser obtida como sendo a composição $B = A \circ R$ (dado que os conjuntos são finitos e todo conjunto pode ser visto como uma relação unidimensional). Logo:

$$\mu_B(y) = \bigvee_{x \in X} (\mu_A(x) \wedge \mu_R(x, y)) = \begin{cases} 1, & \text{para } y = 2, 6 \\ 0, & \text{do contrário} \end{cases}$$

Princípio da Extensão - Transformação de variáveis

O princípio da extensão permite estender o domínio de uma função de pontos em U para conjuntos nebulosos em U .

Suponha um mapeamento $y = f(x)$ e o conjunto fuzzy:

$$A = \left\{ \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \dots + \frac{\mu_A(x_n)}{x_n} \right\}$$

Além de mapear os valores $y_i = f(x_i)$ no conjunto imagem B , é necessário mapear o grau de pertinência destes valores calculados, conforme:

$$B = \left\{ \frac{\mu_A(x_1)}{y_1} + \frac{\mu_A(x_2)}{y_2} + \dots + \frac{\mu_A(x_n)}{y_n} \right\}$$

Note que é possível existir dois elementos diferentes, digamos x_1 e x_n , que sejam mapeados para o mesmo valor, suponha y^* . Neste caso, é preciso lidar com a contradição de um mesmo elemento no conjunto imagem possuir dois graus de pertinência distintos, isto é:

$$B = \left\{ \frac{\mu_A(x_1)}{y^*} + \frac{\mu_A(x_2)}{y_2} + \dots + \frac{\mu_A(x_n)}{y^*} \right\}$$

Logo, adota-se o maior valor de pertinência, resultando na seguinte definição:

Definição 5.1: Princípio da Extensão

Seja $f : U \rightarrow V$. Dado um conjunto nebuloso A em U , o conjunto nebuloso $B = f(A)$ em V induzido por f é dado por

$$\mu_B(y) = \max_{x \in f^{-1}(y)} \mu_A(x), \quad y \in V,$$

em que $f^{-1}(y)$ é conjunto de todos os pontos $x \in U$ tal que $f(x) = y$.

No caso em que a função f admite uma inversa,

$$\mu_B(y) = \mu_A(f^{-1}(y)), \quad y \in V.$$

Exemplo 5.3. Seja o conjunto “ x é aproximadamente zero” com função de pertinência $\mu_A(x) = e^{-x^2}$, em \mathbb{R} , apresentada na Figura 5.1. Seja a função $y = f(x) = x^2 : \mathbb{R} \rightarrow \mathbb{R}_+$, a “inversa” de $f(x)$ é $\pm\sqrt{y}$. Então o conjunto “ y é aproximadamente zero” em \mathbb{R} é dado por $\mu_B(y) = e^{-y}$, $y \geq 0$, e é apresentado na Figura 5.2.

Exemplo 5.4. Considere $y = f(x) = 0,6x + 4$. O conjunto imagem é definido como “aproximadamente 5”:

$$AP_5 = \left\{ \frac{0,3}{3} + \frac{1}{5} + \frac{0,3}{7} \right\}$$

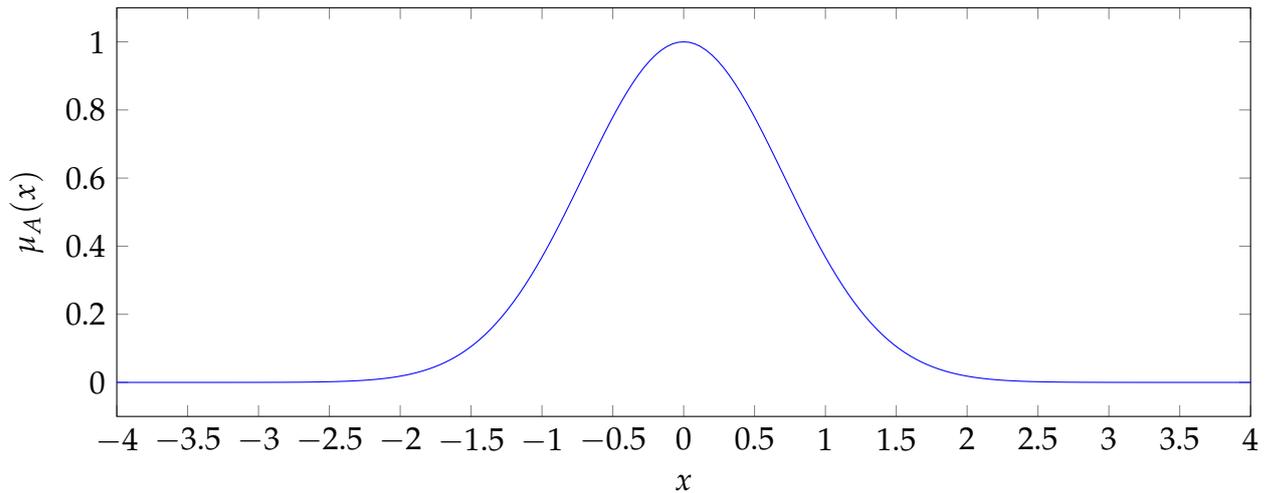


Figura 5.1: Conjunto nebuloso “x é aproximadamente zero”.

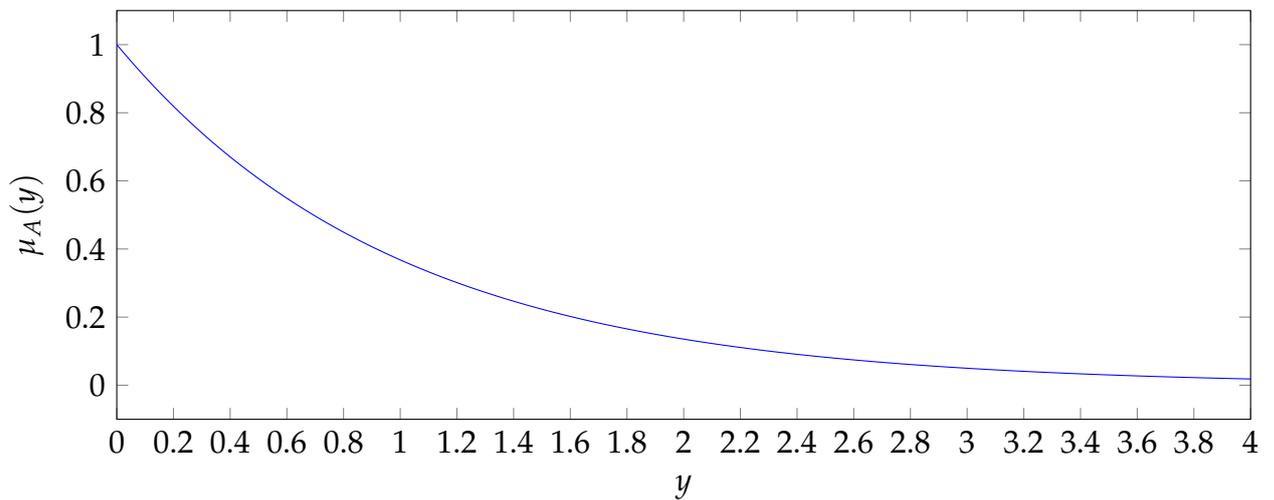


Figura 5.2: Conjunto nebuloso “y é aproximadamente zero”.

Assim:

$$\begin{aligned}
 B &= \left\{ \frac{0,3}{f(3)} + \frac{1}{f(5)} + \frac{0,3}{f(7)} \right\} \\
 &= \left\{ \frac{0,3}{0,6(3) + 4} + \frac{1}{0,6(5) + 4} + \frac{0,3}{0,6(7) + 4} \right\} \\
 &= \left\{ \frac{0,3}{5,8} + \frac{1}{7} + \frac{0,3}{8,2} \right\}.
 \end{aligned}$$

Generalizando, é possível desenvolver o Princípio da Extensão para funções com múltiplas entradas, i.e., $f : X_1 \times X_2 \times \dots \times X_n \rightarrow Y$. O mapeamento para dados alguns conjuntos fuzzy de entrada $B = f(A_1, A_2, \dots, A_n)$ resultará em:

$$\mu_B(y) = \max_{x_1, x_2, \dots, x_n \in f^{-1}(y)} \left\{ \min [\mu_{A_1}(x_1), \mu_{A_2}(x_2), \dots, \mu_{A_n}(x_n)] \right\}$$

Os operadores max e min, para conjuntos limitados, são substituídos, respectivamente, pelos operadores sup e inf, para conjuntos infinitos.

Aritmética Nebulosa

O Princípio da Extensão com duas variáveis pode ser aplicado para definir a chamada Aritmética Nebulosa.

Exemplo 5.5. Deseja-se realizar a simples adição de dois números fuzzy. Definindo o número fuzzy “um” como:

$$Um = \left\{ \frac{0,2}{0} + \frac{1}{1} + \frac{0,2}{2} \right\}$$

O resultado de “um fuzzy” mais “um fuzzy” pelo princípio da extensão é dado pelo conjunto “dois fuzzy”

$$\begin{aligned} um + um = dois &= \left\{ \frac{0,2}{0} + \frac{1}{1} + \frac{0,2}{2} \right\} + \left\{ \frac{0,2}{0} + \frac{1}{1} + \frac{0,2}{2} \right\} \\ &= \left\{ \frac{\min(0,2,0,2)}{0} + \frac{\max[\min(0,2,1), \min(0,2,1)]}{1} + \right. \\ &\quad \left. \frac{\max[\min(0,2,0,2), \min(1,1), \min(0,2,0,2)]}{2} \right. \\ &\quad \left. \frac{\max[\min(1,0,2), \min(1,0,2)]}{3} + \frac{\min(0,2,0,2)}{4} \right\} \\ &= \left\{ \frac{0,2}{0} + \frac{0,2}{1} + \frac{1}{2} + \frac{0,2}{3} + \frac{0,2}{4} \right\} \end{aligned}$$

Exercício 5.1

Considere o conjunto nebuloso “ x é aproximadamente $\pi/2$ ” dado pela função de pertinência $\mu_A(x) = e^{-\frac{(x - \pi/2)^2}{0.02}}$. Considere a função $y = \text{sen}(x)$, e encontre o conjunto nebuloso “ y é aproximadamente 1”. Apresente os gráficos das funções de pertinência.

Exercício 5.2

Considere o conjunto nebuloso “ x é aproximadamente 2” definido como:

$$\mu_A(x) = \begin{cases} x - 1, & \text{se } 1 \leq x \leq 2 \\ 3 - x, & \text{se } 2 \leq x \leq 3 \\ 0, & \text{do contrário} \end{cases}$$

Considere a função $y = x^2$ e determine a extensão para este caso. Apresente os gráficos das funções de pertinência.

Exercício 5.3

Defina o conjunto X :

$$X = \left\{ \frac{0,1}{1} + \frac{1}{2} + \frac{0,5}{3} \right\}$$

Usando o princípio da extensão, determine:

- a) $z = x^2$
- b) $z = x.x$

Exercício 5.4

Defina o conjunto X :

$$X = \left\{ \frac{0,1}{-3} + \frac{0,3}{-2} + \frac{0,7}{-1} + \frac{1}{10} + \frac{0,7}{1} + \frac{0,3}{2} + \frac{0,1}{3} \right\}$$

Repita o exercício anterior e comente diferenças e similaridades.

Aula 6

Variáveis Linguísticas e Regras Nebulosas Se-Então

Nesta aula, definiremos variáveis linguísticas e regras nebulosas se-então.

Variável Linguística

Definição 6.1: Variável Linguística

Se uma variável pode ser descrita por palavras em uma linguagem natural, ela é chamada de variável linguística e associamos um conjunto nebuloso no universo de discurso da variável para cada palavra.

Exemplo 6.1. Podemos descrever um homem como “baixo”, “médio” ou “alto”, e utilizar, por exemplo, os conjuntos nebulosos apresentados na Figura 6.1.

Por meio dos conjunto nebulosos, podemos formalizar matematicamente o valor de variáveis linguísticas.

No dia a dia, é muito comum também utilizarmos alguns modificadores quando descrevemos variáveis linguísticas.

Podemos classificar os termos mais utilizados na descrição de variáveis linguísticas em 3 grupos:

- termos primários, que utilizamos como nomes para os conjuntos nebulosos
- negação e conectores (“não”, “e”, “ou”) que implementamos utilizando complemento, união e interseção nebulosas
- modificadores, que são adjetivos utilizados na descrição de termos primários

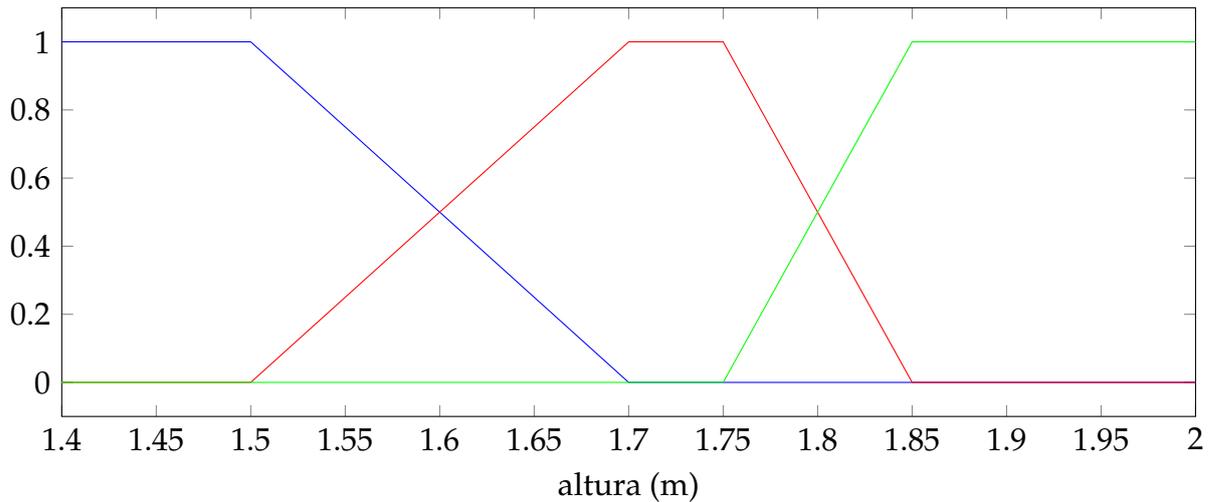


Figura 6.1: Exemplo de variável linguística representando a altura de um homem. Podemos representar um homem “baixo” pelo conjunto em azul, um homem “médio” pelo conjunto em vermelho e um homem “alto” pelo conjunto em verde.

Tais modificadores podem ser implementados como outros conjuntos nebulosos (por exemplo, podemos ter um conjunto alto e um conjunto muito alto), ou podem-se estabelecer fórmulas para utilizá-los.

Wang 1999 sugere as seguintes definições

- “muito” $\mu_{\text{muito}A}(x) = (\mu_A(x))^2$
- “mais ou menos” $\mu_{\pm A}(x) = (\mu_A(x))^{1/2}$

Regras Nebulosas Se-Então

Em sistemas nebulosos, normalmente expressamos um conhecimento na forma de condicionais se-então da forma

Se <proposição nebulosa>, então <proposição nebulosa>

Uma proposição nebulosa é uma frase do tipo

“ x é Alto” ou “ x é Alto e x é não Baixo”

E para fazermos composições, utilizamos complemento nebuloso para a negação, união nebulosa para “ou” e interseção nebulosa para “e”.

Note que a composição pode ser feita com variáveis diferentes.

Exemplo 6.2. Seja “ v ” a velocidade de um carro e “ a ” a aceleração, podemos ter um proposição do tipo

$$v \text{ é } B_v \text{ e } a \text{ é } B_a$$

com B_v um conjunto representando velocidade baixa e B_a outro conjunto representando aceleração baixa. A função de pertinência dessa proposição seria

$$\mu_p(v, a) = \mu_{B_v}(v) \star \mu_{B_a}(a)$$

em que \star representa uma t-norma.

Interpretações de Regras Se-Então Nebulosas

Uma forma de se estender as regras se-então é olhando como elas funcionam usualmente.

Se p então q , pode ser representado como $p \rightarrow q$

que tem a seguinte tabela verdade

p	q	$p \rightarrow q$
V	V	V
V	F	F
F	V	V
F	F	V

e pode ser escrita em álgebra booleana como

$$p \rightarrow q = \bar{p} + q \tag{6.1}$$

ou

$$p \rightarrow q = (pq) + \bar{p} \tag{6.2}$$

Podemos então estender essas implicações para implicações nebulosas de algumas formas diferentes

- Implicação de Dienes-Rescher: (6.1) usando os operadores de Zadeh

$$\mu_{pqD}(x, y) = \max(1 - \mu_p(x), \mu_q(y))$$

- Implicação de Lukasiewicz: (6.1) com complemento de Zadeh e s-norma de Yager com $w = 1$

$$\mu_{pqL}(x, y) = \min(1, 1 - \mu_p(x) + \mu_q(y))$$

- Implicação de Zadeh: (6.2) usando os operadores de Zadeh

$$\mu_{pqZ}(x, y) = \max(\min(\mu_p(x), \mu_q(y)), 1 - \mu_p(x))$$

Note entretanto que as implicações dessa forma geram resultados mesmo quando $\mu_p(x)$ e $\mu_q(y)$ são baixos, o que muitas vezes é estranho.

Uma outra forma de se estender as implicações é pensando que se a proposição inicial não for verdadeira, não podemos falar nada sobre a segunda. Ou seja,

Se <proposição 1>, então <proposição 2>. Senão <nada>

Isto implica que a nossa implicação pode ser escrita, em álgebra booleana, como

$$p \rightarrow q = pq.$$

Usando esta nova interpretação chegamos nas implicações de Mamdani.

Definição 6.2: Implicações de Mamdani

Seja um regra Se-Então Nebulosa da forma

Se <proposição p >, então <proposição q >.

Uma forma de interpretar tal regra é pela implicação de Mamdani do mínimo

$$\mu_{pq_{mm}}(x, y) = \min(\mu_p(x), \mu_q(y))$$

Outra forma de interpretar esta regra é pela implicação de Mamdani do produto

$$\mu_{pq_{mp}}(x, y) = \mu_p(x)\mu_q(y)$$

Exemplo 6.3. Vamos considerar uma regra se-então que nos diz que se a velocidade está baixa temos que acelerar, descrita como

Se “ v é devagar” então “ a é grande”

Considere que o conjunto v devagar é apresentado na Figura 6.2, e que o conjunto a grande é apresentado na Figura 6.3.

A implicação de Dienes-Rescher é apresentada na Figura 6.4. A implicação de Lukasiewicz é apresentada na Figura 6.5. A implicação de Zadeh é apresentada na Figura 6.6. A implicação de Mamdani do mínimo é apresentada na Figura 6.7. A implicação de Mamdani do produto é apresentada na Figura 6.8.

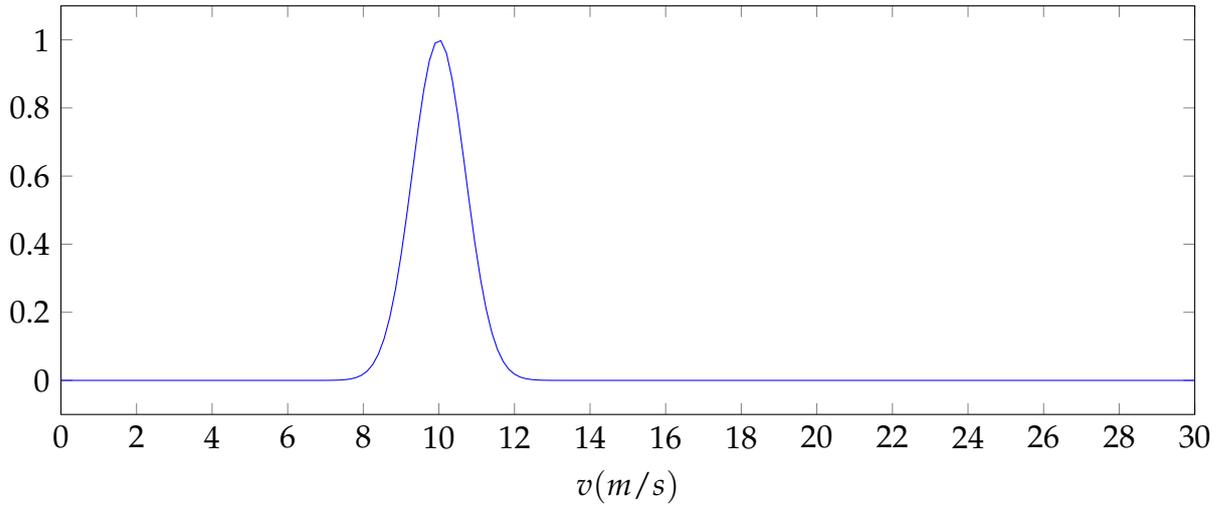


Figura 6.2: Conjunto nebuloso “ v é devagar”.

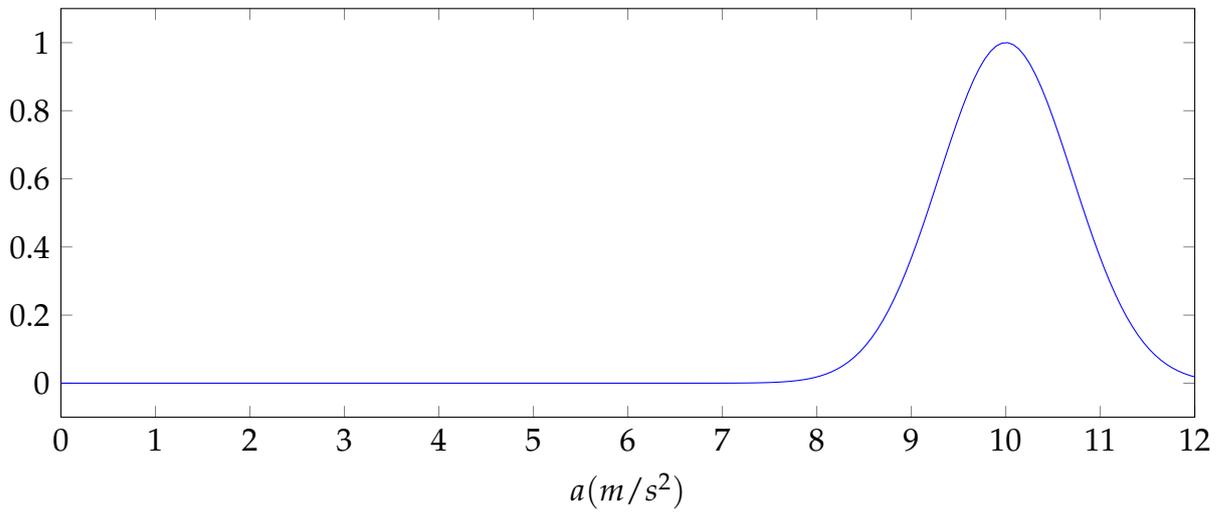


Figura 6.3: Conjunto nebuloso “ a é grande”.

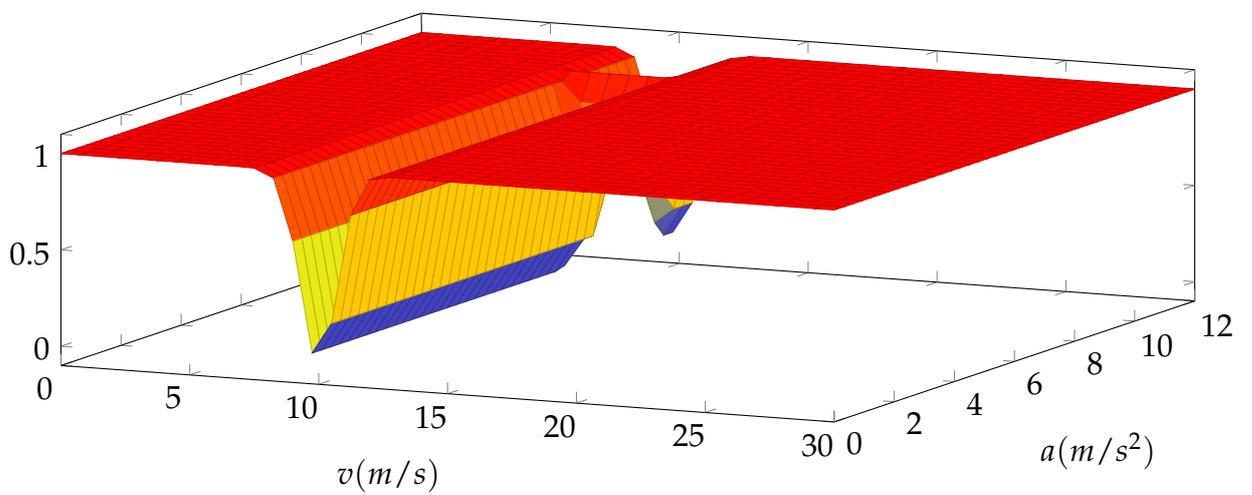


Figura 6.4: Implicação de Dienes-Rescher para Se “ v é devagar” então “ a é grande”.

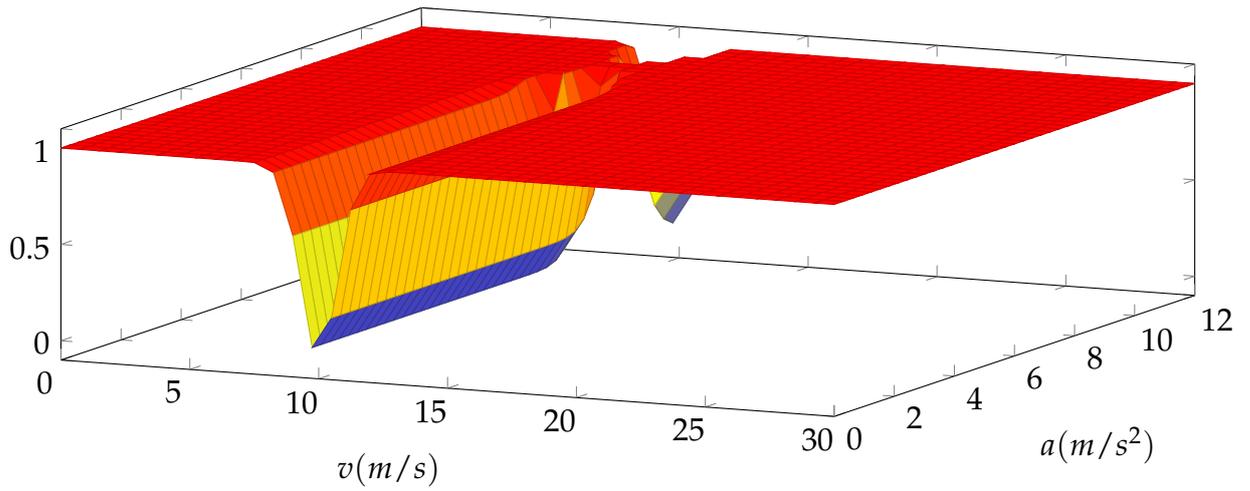


Figura 6.5: Implicação de Lukasiewicz para Se “ v é devagar” então “ a é grande”.

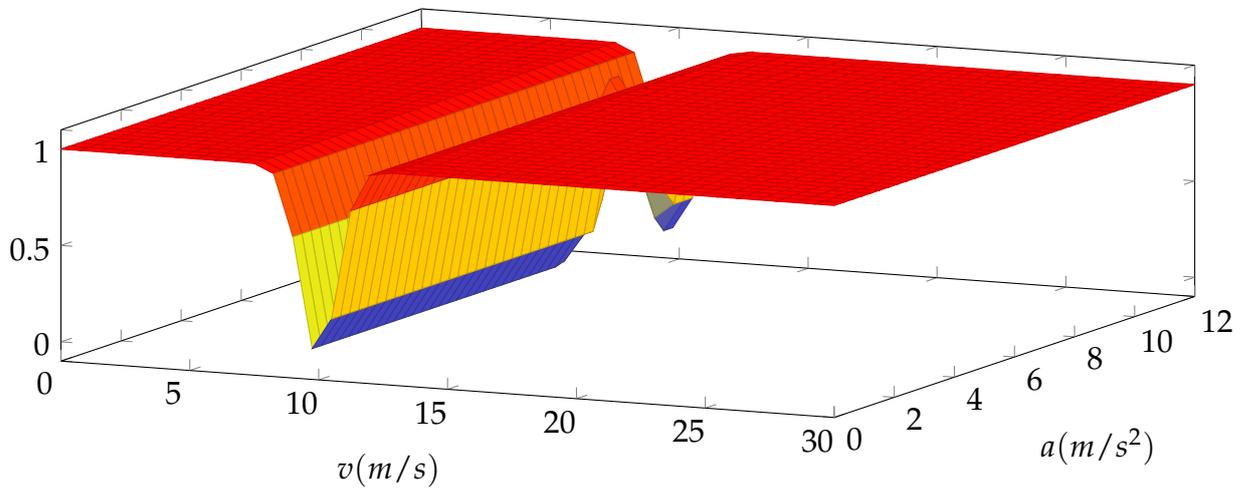


Figura 6.6: Implicação de Zadeh para Se “ v é devagar” então “ a é grande”.

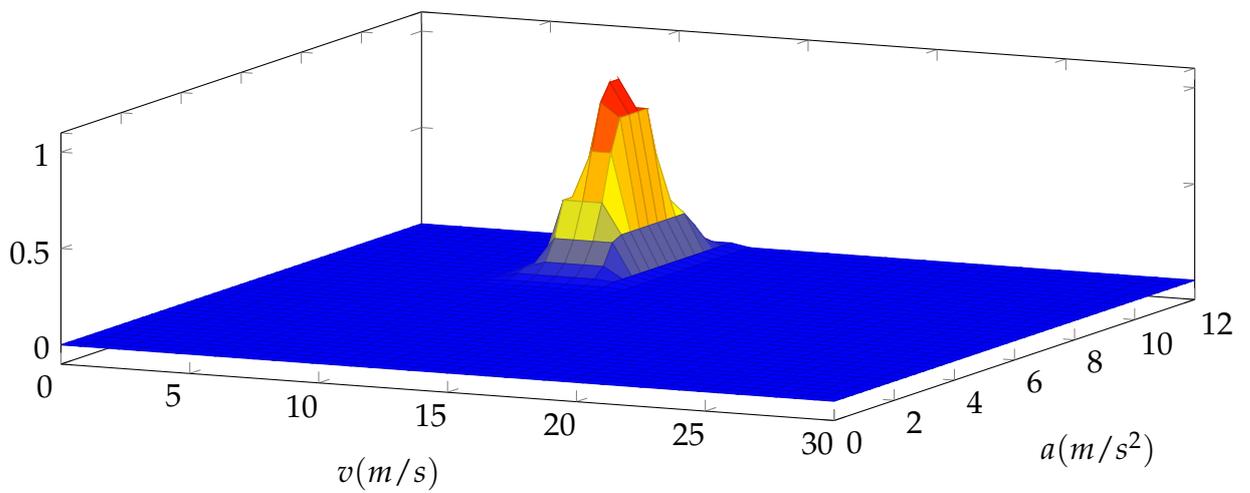


Figura 6.7: Implicação de Mamdani do mínimo para Se “ v é devagar” então “ a é grande”.

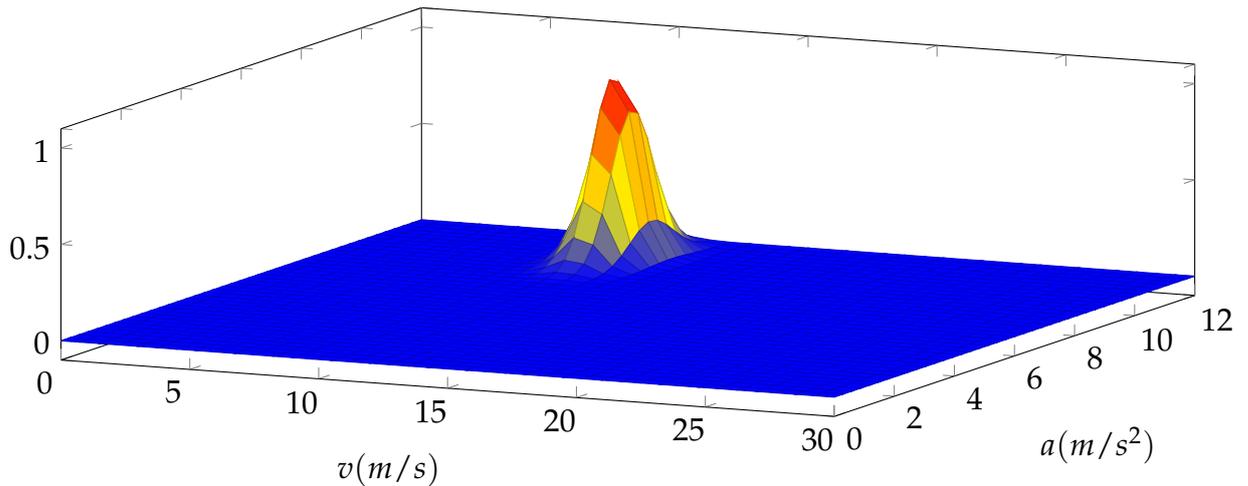


Figura 6.8: Implicação de Mamdani do produto para Se “ v é devagar” então “ a é grande”.

Exercício 6.1

Considere uma regra nebulosa que seria utilizada em um controle de nível, do tipo

Se “nível é baixo”, então “abre muito a válvula”

Em que o conjunto nível baixo é representado por uma função de pertinência trapezoidal com os parâmetros $[0, 0, 0.5, 1]$, considerando que o nível pertence ao conjunto $[0, 3]$, e o conjunto abre muito a válvula é representado por uma função de pertinência trapezoidal com os parâmetros $[0.5, 0.7, 1, 1]$, considerando que a abertura da válvula pertence ao conjunto $[0, 1]$.

Implemente as diferentes implicações apresentadas na aula de hoje e gere os gráficos que representam a função de pertinência da implicação nebulosa.

Aula 7

Lógica Nebulosa

Na lógica clássica, os valores verdade de proposições lógicas podem assumir apenas dois valores (falso ou verdadeiro; 0 ou 1) e podemos obter novas proposições de proposições existentes utilizando um raciocínio lógico. A lógica nebulosa estende a lógica clássica na forma em que ela permite que proposições assumam valores verdade quaisquer no intervalo [0,1]. Essa generalização nos permite realizar um raciocínio aproximado, ou seja, deduzir conclusões imprecisas (proposições nebulosas) a partir de premissas imprecisas (proposições nebulosas).

Lógica Clássica

Na lógica clássica, podemos utilizar um conjunto completo de primitivas e representar qualquer função lógica. O conjunto mais comum é formado pela negação (“não”), conjunção (“e”) e disjunção (“ou”), e qualquer função lógica pode ser representada a partir desse conjunto de primitivas.

Quando uma função lógica é sempre verdadeira, independente dos valores verdade das proposições na função, ela é chamada de tautologia. Quando ela é sempre falsa, ela é chamada de contradição.

Várias formas de tautologias podem ser utilizadas para fazerem inferências dedutivas. Tais tautologias são chamadas de regras de inferência. As três regras de inferência mais comuns são:

- *Modus Ponens* (Inferência pela suficiência)
 - x é A
 - Se x é A , então y é B
 - * y é B
- *Modus Tollens* (Inferência pela necessidade)

- y não é B
- Se x é A , então y é B
 - * x não é A
- *Silogismo Hipotético* (Composição de inferências)
 - Se x é A , então y é B
 - Se y é B , então z é C
 - * Se x é A , então z é C

Regra de Inferência Composicional

Dada uma função $y = f(x)$, e $x = a$, podemos inferir que $y = f(a)$, como apresentado na Figura 7.1.

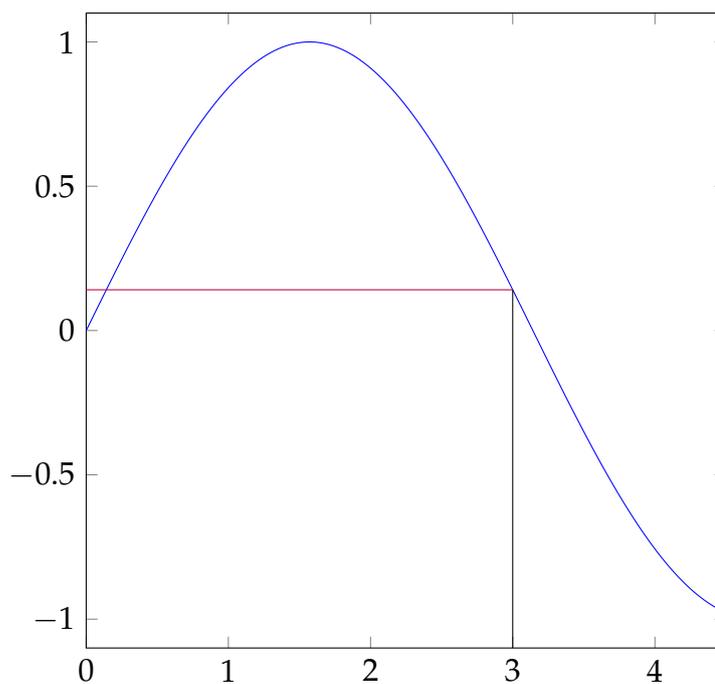


Figura 7.1: Inferência utilizando uma função $y = f(x)$.

Se ao invés de pontos, tivéssemos intervalos e uma função intervalar (uma função que retorna um intervalo), a inferência pode ser realizada da seguinte maneira: tomamos a extensão cilíndrica de a e pegamos a interseção da extensão com a função intervalar. Por fim, inferimos que y é dado pela projeção desta interseção. Tal procedimento é ilustrado na Figura 7.2.

Indo mais um passo, podemos generalizar esta relação para conjuntos e relações nebulosas. Seja $A \subset U$ um conjunto nebuloso, e Q uma relação nebulosa em $U \times V$. Podemos fazer a inferência em V tomando a extensão cilíndrica de A em $U \times$

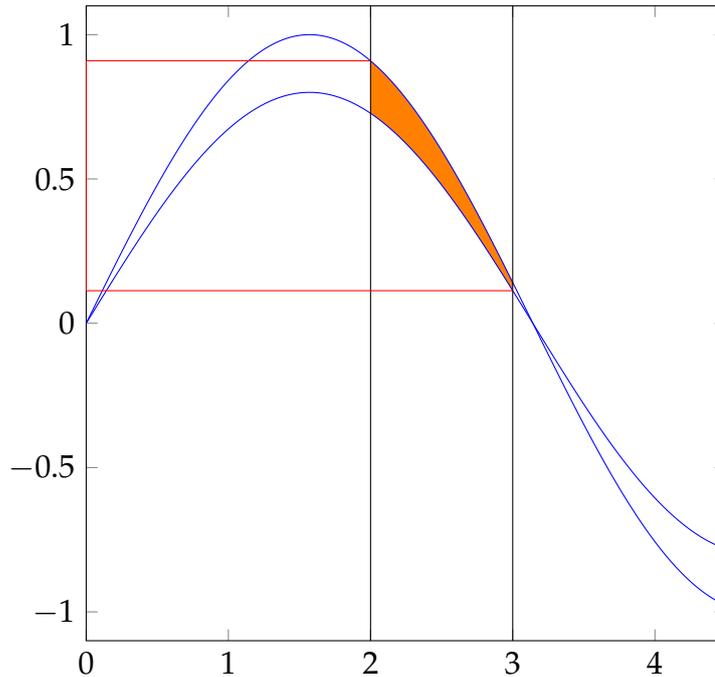


Figura 7.2: Inferência intervalar, utilizando uma função intervalar.

V , fazendo a interseção com a relação nebulosa Q e projetando-a em V , obtemos o conjunto B inferido.

De forma específica, considere $\mu_A(x)$ e $\mu_Q(x, y)$, temos que

$$\begin{aligned}\mu_{A_e}(x, y) &= \mu_A(x) \\ \mu_{A_e \cap Q}(x, y) &= \mu_{A_e}(x, y) \star \mu_Q(x, y) = \mu_A(x) \star \mu_Q(x, y) \\ \mu_B(y) &= \sup_{x \in U} (\mu_A(x) \star \mu_Q(x, y))\end{aligned}$$

em que \star representa qualquer t-norma, e a última equação acima é chamada de regra de inferência composicional, ou composição sup-estrela.

Como vimos na última aula, regras nebulosas Se-Então podem ser vistas como relações nebulosas no produto cartesiano entre as premissas e conclusões. Além disso, vimos também que existem diferentes formas de implicação possíveis e cada uma leva a diferentes relações nebulosas.

Dessa forma, podemos utilizar a regra de inferência composicional e as diferentes inferências nebulosas para generalizarmos as regras de inferência da lógica clássica:

- *Modus Ponens Generalizado*

- x é A' , representado por um conjunto nebuloso A'
- Se x é A , então y é B , representado por uma relação nebulosa $A \rightarrow B$
- * y é B' , representado por um conjunto nebuloso B' inferido, com uma função de pertinência $\mu_{B'}(y) = \sup_{x \in U} (\mu_{A'}(x) \star \mu_{A \rightarrow B}(x, y))$

- *Modus Tollens Generalizado*

- y não é B , representado por um conjunto nebuloso B'
- Se x é A , então y é B , representado por uma relação nebulosa $A \rightarrow B$
 - * x não é A , inferido como x é A' , com função de pertinência $\mu_{A'}(x) = \sup_{y \in V} (\mu_{B'}(y) \star \mu_{A \rightarrow B}(x, y))$

- *Silogismo Hipotético Generalizado*

- Se x é A , então y é B , representado por uma relação nebulosa $A \rightarrow B$
- Se y é B' , então z é C , representado por uma relação nebulosa $B' \rightarrow C$
 - * Se x é A , então z é C' , representado por uma relação nebulosa $A \rightarrow C'$ com função de pertinência $\mu_{A \rightarrow C'}(x, z) = \sup_{y \in V} (\mu_{A \rightarrow B}(x, y) \star \mu_{B' \rightarrow C}(y, z))$

Exercício 7.1

Continuando do exercício da aula anterior, utilizando a implicação de Mamdani do produto e a t-norma do produto, escolha 4 conjuntos nebulosos de entrada diferentes para o nível, e utilizando o *Modus Ponens Generalizado* encontre os 4 conjuntos nebulosos de saída correspondentes para a abertura da válvula e apresente os gráficos das funções de pertinência obtidos.

Aula 8

Banco de Regras e Sistemas de Inferência Nebulosa

Um banco de regras nebulosas consiste de um conjunto de regras Se-Então Nebulosas, e é o componente principal em um sistema nebuloso (no sentido de que todos os outros componentes servem para implementar essas regras).

Consideramos apenas regras da forma

Regra k : Se x_1 é A_1^k e x_2 é A_2^k e \dots e x_n é A_n^k , então y é B^k

Note que apesar de essa suposição parecer restritiva, várias outras formas de regras podem ser escritas dessa forma.

1. Regras parciais - podemos ter regras que não utilizam todas as m entradas (variáveis premissas) em nosso banco de regras. Para essas regras, basta assumir que os conjuntos nebulosos correspondentes a essas entradas não utilizadas têm função de pertinência $\mu_l(x) = 1$.
2. Regras com "ou" - nesse caso, basta dividir as regras em um número de regras que possuem a mesma conclusão.
3. Proposições nebulosas - podemos ter proposições nebulosas diretas sem nenhuma condição. Para isso, basta tomar as pertinências de todas as variáveis premissas como sendo 1 em todo o universo de discurso.
4. Regras graduais - regras que representam que as saídas são diretamente proporcionais ou inversamente proporcionais às entradas podem ser implementadas escolhendo-se corretamente as funções de pertinência de entrada e saída.
5. Regras não nebulosas - Para implementar regras não-nebulosas, basta tomar conjuntos nebulosos *singleton* para as entradas e saída.

Propriedades de conjuntos de regras

Quando vamos implementar bancos de regras nebulosas, é interessante que nosso conjunto de regras tenha algumas propriedades.

Definição 8.1: Completude

Um conjunto de regras nebulosas é dito completo se, para quaisquer variáveis premissas pertencentes ao universo de discurso, $x_i \in U_i$, existir pelo menos uma regra tal que

$$\mu_{A_i^k}(x_i) \neq 0, \forall i \in 1 \dots n$$

Quando dizemos que um conjunto de regras é completo sabemos que, para quaisquer variáveis premissas que tomarmos dentro do universo de discurso, pelo menos uma regra estará ativa (terá pertinência associada à parte se da regra diferente de zero).

Definição 8.2: Consistência

Dizemos que um conjunto de regras nebulosas é consistente se não existirem regras com o mesmo “Se”, mas com “então” diferente.

Definição 8.3: Continuidade

Um conjunto de regras nebulosas Se-Então é contínuo se não existirem regras vizinhas cujos conjuntos nebulosos “então” tem interseção vazia.

Sistemas de Inferência Nebulosa

Na última aula, aprendemos como mapear um conjunto nebuloso de entrada para um conjunto nebuloso de saída utilizando-se uma única regra nebulosa Se-Então. Entretanto, na maioria dos sistemas nebulosos de interesse, sempre teremos mais de uma regra. Dessa forma, precisamos definir como fazemos a inferência do conjunto nebuloso de entrada para um conjunto nebuloso de saída quando temos um banco de regras nebulosas.

Existem duas formas de inferência em um conjunto de regras: Inferência baseada na composição das regras e inferência baseada nas regras individuais

Inferência Baseada na Composição

Neste tipo de inferência, todas as regras são combinadas em uma única relação no produto cartesiano da entrada \times saída (que pode ser vista como uma única regra) e em seguida é utilizado o Modus Ponens Generalizado.

Existem duas formas de combinar as regras nebulosas em uma única regra.

A primeira forma, chamada de **combinação Mamdani**, entende que as regras são condições independentes e define que sua combinação é dada pela união das relações correspondentes às regras.

A segunda forma, chamada de **combinação de Gödel**, entende que todas as regras devem ser atendidas simultaneamente e define que sua combinação é dada pela interseção das relações correspondentes às regras.

A inferência baseada na composição pode ser então feita seguindo-se os seguintes passos:

1. Para cada uma das regras nebulosas determine a função de pertinência correspondente à todas as variáveis premissas, dada por

$$\mu_{A_1^k \times A_2^k \times \dots \times A_n^k}(x_1, x_2, \dots, x_n) = \mu_{A_1^k}(x_1) \star \mu_{A_2^k}(x_2) \star \dots \star \mu_{A_n^k}(x_n)$$

2. Utilize alguma implicação nebulosa para calcular as relações nebulosas representadas por cada uma das regras Se-Então

$$\mu_{R^k}(\mathbf{x}, y) = \mu_{A_1^k \times A_2^k \times \dots \times A_n^k \rightarrow B^k}(x_1, x_2, \dots, x_n, y)$$

3. Determine à relação nebulosa dada pela combinação das relações de cada regra

- Combinação Mamdani

$$\mu_{Q_M}(\mathbf{x}, y) = \mu_{R^1}(\mathbf{x}, y) \dot{+} \dots \dot{+} \mu_{R^r}(\mathbf{x}, y)$$

com $\dot{+}$ uma s-norma qualquer, e r o número total de regras no banco.

- Combinação Gödel

$$\mu_{Q_G}(\mathbf{x}, y) = \mu_{R^1}(\mathbf{x}, y) \star \dots \star \mu_{R^r}(\mathbf{x}, y)$$

4. Para uma entrada “ \mathbf{x} é \tilde{A} ”, com função de pertinência $\mu_{\tilde{A}}(\mathbf{x})$, encontre a função de pertinência do conjunto de saída utilizando o Modus Ponens Generalizado

$$\mu_{\tilde{B}}(y) = \sup_{\mathbf{x} \in U} \mu_{\tilde{A}}(\mathbf{x}) \star \mu_Q(\mathbf{x}, y)$$

Inferência Individual de cada Regra

Nesse tipo de inferência, determina-se a saída de cada regra separadamente e após isso os resultados são combinados usando-se união ou interseção.

A inferência individual de cada regra pode ser realizada pelos seguintes passos:

1. Para cada uma das regras nebulosas determine a função de pertinência correspondente à todas as variáveis premissas, dada por

$$\mu_{A_1^k \times A_2^k \times \dots \times A_n^k}(x_1, x_2, \dots, x_n) = \mu_{A_1^k}(x_1) \star \mu_{A_2^k}(x_2) \star \dots \star \mu_{A_n^k}(x_n)$$

2. Utilize alguma implicação nebulosa para calcular as relações nebulosas representadas por cada uma das regras Se-Então

$$\mu_{R^k}(\mathbf{x}, y) = \mu_{A_1^k \times A_2^k \times \dots \times A_n^k \rightarrow B^k}(x_1, x_2, \dots, x_n, y)$$

3. Para uma entrada “ \mathbf{x} é \tilde{A} ”, com função de pertinência $\mu_{\tilde{A}}(\mathbf{x})$, encontra-se a função de pertinência do conjunto de saída utilizando o Modus Ponens Generalizado para cada regra

$$\mu_{\tilde{B}_k}(y) = \sup_{\mathbf{x} \in U} \mu_{\tilde{A}}(\mathbf{x}) \star \mu_{R^k}(\mathbf{x}, y)$$

4. Realiza-se a combinação de todos os conjuntos nebulosos de saída

- Por união

$$\mu_{\tilde{B}}(y) = \mu_{\tilde{B}_1}(y) \dot{+} \dots \dot{+} \mu_{\tilde{B}_r}(y)$$

- Por interseção

$$\mu_{\tilde{B}}(y) = \mu_{\tilde{B}_1}(y) \star \dots \star \mu_{\tilde{B}_r}(y)$$

Sistemas de Inferência mais Comuns

Os dois sistemas de inferência nebulosos mais comumente utilizados são

- Inferência do produto: neste tipo de inferência, utilizamos a (i) inferência individual de cada regra com combinação pela união, (ii) Implicação do produto de Mamdani, (iii) produto algébrico para todas as t-normas e max para todas as s-normas.

$$\mu_{\tilde{B}}(y) = \max_{k=1 \dots r} \left(\sup_{\mathbf{x} \in U} \left(\mu_{\tilde{A}}(\mathbf{x}) \left(\prod_{i=1}^n \mu_{A_i^k}(x_i) \right) \mu_{B^k}(y) \right) \right)$$

- Inferência do mínimo: neste tipo de inferência, utilizamos a (i) inferência individual de cada regra com combinação pela união, (ii) Implicação do mínimo de Mamdani, (iii) min para todas as t-normas e max para todas as s-normas.

$$\mu_{\tilde{B}}(y) = \max_{k=1 \dots r} \left(\sup_{\mathbf{x} \in U} \min \left(\mu_{\tilde{A}}(\mathbf{x}), \mu_{A_1^k}(x_1), \dots, \mu_{A_n^k}(x_n), \mu_{B^k}(y) \right) \right)$$

É possível mostrar que, em ambos esses casos, não faz diferença se é utilizada a inferência individual ou a inferência baseada na composição. Além disso, ambos os casos podem ser simplificados quando utilizamos um conjunto singleton nebuloso como entrada.

Exercício 8.1

Considere um banco de regras nebulosas que seria utilizado em um controle de nível, do tipo

Regra 1: Se “nível é baixo”, então “abre muito a válvula”

Regra 2: Se “nível é bom”, então “fecha a válvula”

Em que o conjunto nível baixo é representado por uma função de pertinência trapezoidal com os parâmetros $[0, 0, 0.5, 1]$ e o conjunto nível bom é representado por uma função de pertinência trapezoidal com os parâmetros $[0.5, 1, 3, 3]$ considerando que o nível pertence ao conjunto $[0, 3]$.

O conjunto abre muito a válvula é representado por uma função de pertinência trapezoidal com os parâmetros $[0.5, 0.7, 1, 1]$, o conjunto fecha a válvula é representado por uma função de pertinência trapezoidal com os parâmetros $[0, 0, 0.3, 0.5]$, considerando que a abertura da válvula pertence ao conjunto $[0, 1]$.

Implemente o sistema de inferência do produto, e gere o conjunto nebuloso de saída para 4 conjuntos nebulosos de entrada diferentes.

Aula 9

Fuzzificação e Defuzzificação (*Fuzzification and Defuzzification*)

Na última aula aprendemos como as máquinas de inferência podem ser utilizadas para combinar um banco de regras nebulosas em uma única relação nebulosa da entrada para a saída. Isto nos permitiu encontrar um conjunto de saída nebuloso de um banco de regras nebulosas quando a entrada é um conjunto nebuloso.

O objetivo da aula de hoje é conseguirmos uma saída real (numérica) do banco de regras a partir de uma entrada real (numérica).

Fuzzificadores

Um fuzzificador é um mapeamento de um único ponto no universo de discurso para um conjunto nebuloso no universo de discurso. Ele é o elemento em um sistema nebuloso que permite que utilizemos pontos (ou valores numéricos) como entrada ao invés de conjuntos nebulosos.

Ou seja, um fuzzificador transforma um ponto/vetor $\mathbf{x}^* \in U \subset \mathbb{R}^n$ em um conjunto nebuloso \tilde{A} em U .

É desejável que o conjunto nebuloso produzido por um fuzzificador tenha as seguintes três características:

1. Valor de pertinência alto no ponto de entrada \mathbf{x}^* .
2. Seja capaz de suprimir/representar ruído relacionado ao valor de entrada \mathbf{x}^* .
3. Ajude a simplificar as contas da máquina de inferência

Alguns fuzzificadores comuns são:

- Fuzzificador singleton: gera-se um conjunto nebuloso que corresponde somente àquele ponto, ou seja,

$$\mu_{\tilde{A}}(\mathbf{x}) = \begin{cases} 1, & \text{se } \mathbf{x} = \mathbf{x}^* \\ 0, & \text{caso contrário} \end{cases}$$

- Fuzzificador gaussiano: utilizam-se conjuntos nebulosos gaussianos, com centro no ponto de entrada. Tal fuzzificador normalmente tem a forma

$$\mu_{\tilde{A}}(\mathbf{x}) = e^{-\frac{(x_1-x_1^*)^2}{2\sigma_1^2}} \star \dots \star e^{-\frac{(x_n-x_n^*)^2}{2\sigma_n^2}}$$

Com σ_i parâmetros positivos e normalmente utiliza-se o produto ou o mínimo como a t-norma acima.

- Fuzzificador triangular: utilizam-se conjuntos nebulosos triangulares, com centro no ponto de entrada. Tal fuzzificador normalmente tem a forma

$$\mu_{\tilde{A}}(\mathbf{x}) = \begin{cases} \left(1 - \frac{\|x_1-x_1^*\|}{b_1}\right) \star \dots \star \left(1 - \frac{\|x_n-x_n^*\|}{b_n}\right), & \text{se } \|x_i - x_i^*\| \leq b_i \forall i \\ 0, & \text{caso contrário} \end{cases}$$

Com b_i parâmetros positivos e normalmente utiliza-se o produto ou o mínimo como a t-norma acima.

Note que, em todos os 3 fuzzificadores apresentados, a função de pertinência do conjunto vale 1 no ponto \mathbf{x}^* .

Vimos em sala de aula que o uso de um conjunto nebuloso singleton como entrada de um banco de regras simplifica em muito as contas. É possível mostrar que, para o caso em que todas as funções de pertinência do banco de regras são gaussianas, pode-se encontrar uma fórmula fechada para o fuzzificador gaussiano. Além disso, também é possível mostrar que, para o caso em que todas as funções de pertinência do banco de regras são triangulares, pode-se encontrar uma fórmula fechada para o fuzzificador triangular.

Defuzzificadores

Um defuzzificador é um mapeamento de um conjunto nebuloso para um único ponto no universo de discurso. Ele é o elemento em um sistema nebuloso que nos permite transformar a saída de um banco de regras em um único ponto (valor numérico).

Ou seja, um defuzzificador transforma um conjunto nebuloso \tilde{B} em V em um único ponto/vetor $\mathbf{y}^* \in V \subset \mathbb{R}^m$.

É desejável que um defuzzificador tenha as seguintes três características:

1. Plausibilidade: O valor \mathbf{y}^* deve ser representar o conjunto nebuloso \tilde{B} de forma intuitiva, como seu valor médio ou máximo por exemplo.
2. Simplicidade Computacional: um defuzzificador simples permite que o cálculo do sistema nebuloso seja feito mais rápido.
3. Continuidade: Uma pequena variação em \tilde{B} não deve resultar em grandes variações em \mathbf{y}^* .

Alguns defuzzificadores comuns são:

- Defuzzificador por centro de gravidade: Se pensarmos nas funções de pertinência como se fossem funções de distribuição de probabilidade, o defuzzificador por centro de gravidade encontra o valor \mathbf{y}^* como sendo o valor esperado (ou a média) da nossa distribuição

$$\mathbf{y}^* = \frac{\int_V \mathbf{y} \mu_{\tilde{B}}(\mathbf{y}) d\mathbf{y}}{\int_V \mu_{\tilde{B}}(\mathbf{y}) d\mathbf{y}}$$

Lembrando da definição de valor esperado em probabilidade, notamos uma pequena diferença na fórmula utilizada (o denominador). Utilizamos este denominador pois as funções de pertinência não possuem a propriedade de terem “área/volume” unitário (integral sobre o universo de discurso igual a 1), e dessa forma fazemos uma normalização.

A grande vantagem do defuzzificador por centro de gravidade é que ele é muito intuitivo. A grande desvantagem é que seu cálculo é complicado. Um exemplo de defuzzificador por centro de gravidade é apresentado na Figura 9.1.

- Defuzzificador de média central (inferência individual das regras): O defuzzificador de média central tenta ser intuitivo como o por centro de gravidade, porém com um cálculo mais simples.

Lembrando que o conjunto de saída B é formado pela união ou interseção de r conjuntos nebulosos, a média central aproxima o centro de gravidade como sendo a soma ponderada dos pontos médios de cada conjunto nebuloso de saída cujos pesos são dados pelos valores das funções de pertinência naqueles pontos.

Considerando \mathbf{y}_k o ponto médio do conjunto nebuloso de saída \tilde{B}_k , a defuzzificação é dada por

$$\mathbf{y}^* = \frac{\sum_{k=1}^r \mu_{\tilde{B}_k}(\mathbf{y}_k) \mathbf{y}_k}{\sum_{k=1}^r \mu_{\tilde{B}_k}(\mathbf{y}_k)}$$

Um exemplo de defuzzificador de média central é apresentado na Figura 9.2.

- Defuzzificador de máximo: O mais simples dos defuzzificadores propostos, entretanto o menos intuitivo. Simplesmente escolhemos o ponto em que a função de pertinência de saída atinge seu valor máximo. Se mais de um máximo existir, podemos escolher o menor dos máximos (som - do inglês *smallest of maxima*), o maior dos máximos (lom - do inglês *largest of maxima*), o valor médio dos máximos (mom - do inglês *mean of maxima*), ou qualquer outra regra que quisermos estabelecer.

Um exemplo dos diferentes defuzzificadores de máximo é apresentado na Figura 9.3.

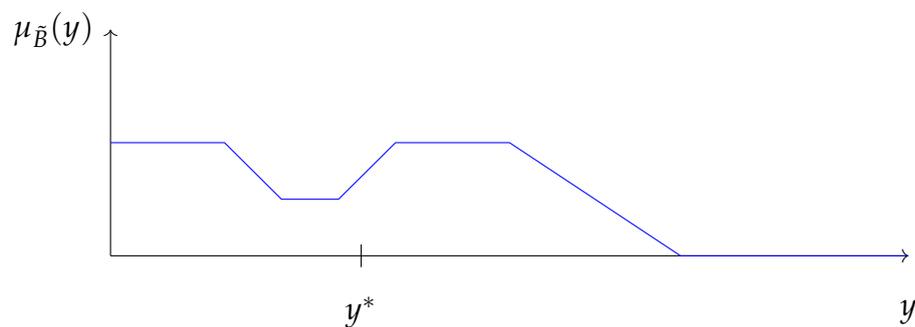


Figura 9.1: Exemplo de defuzzificador por centro de gravidade.

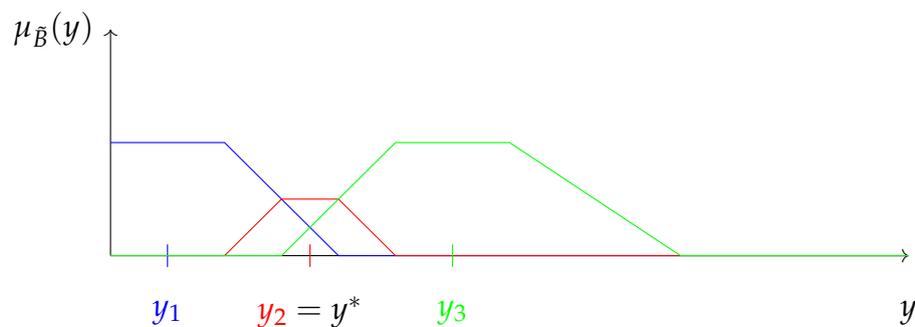


Figura 9.2: Exemplo de defuzzificador de média central. Neste exemplo, por coincidência, o valor defuzzificado coincidiu com o valor médio do segundo conjunto nebuloso.

Exercício 9.1

Continuando do exercício da última aula, implemente um fuzzificador *singleton* e algum dos defuzzificadores apresentados em conjunto com o banco de regras da aula anterior.

Varie o valor do nível (entrada do sistema nebuloso) e levante o valor de abertura da válvula (saída do sistema nebuloso). Apresente o gráfico de entrada \times saída do sistema nebuloso.

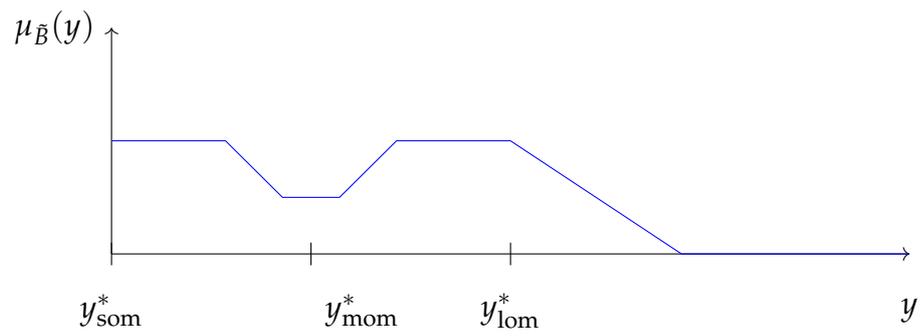


Figura 9.3: Exemplo dos defuzzificadores de máximo. y_{som}^* é o ponto encontrado pelo defuzzificador de menor máximo, y_{lom}^* é o ponto encontrado pelo defuzzificador de maior máximo, e y_{mom}^* é o ponto encontrado pelo valor médio dos máximos.

Aula 10

Exemplo de Controle Utilizando Sistemas Nebulosos

Nesta aula, apresentamos um exemplo de sistema de controle utilizando sistemas nebuloso no MATLAB/Simulink. Primeiro apresentamos o sistema a ser controlado e como podemos simulá-lo, tanto no MATLAB quanto no Simulink. Em seguida, apresentamos uma descrição do sistema de controle utilizando regras Se-Então e variáveis linguísticas e implementamos o sistema de controle nas simulações no MATLAB e no Simulink.

Descrição do Sistema

Na aula de hoje, focaremos nossa atenção no sistema de dois tanques apresentado na Figura 10.1. Para modelar o comportamento desse sistema, consideramos o balanço de massa (vazão volumétrica) nos dois tanques (a taxa de variação do volume é dada pela

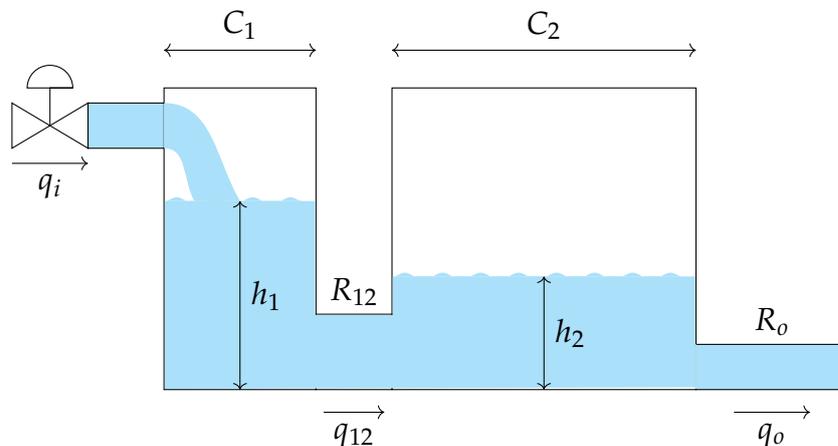


Figura 10.1: Sistema de dois tanques.

diferença entre as vazões de entrada e saída dos tanques)

$$\dot{V}_1 = q_i - q_{12}$$

$$\dot{V}_2 = q_{12} - q_o$$

Como a área da seção dos nossos tanques é constante, temos que

$$V_1 = C_1 h_1$$

$$V_2 = C_2 h_2$$

em que as capacitâncias C_1 e C_2 correspondem às area das seções dos tanques 1 e 2 respectivamente.

Vamos simplificar nossa análise e considerar que podemos modelar as vazões de saída, q_{12} e q_o por relações lineares. Quando representamos vazões dessa forma, normalmente temos um parâmetro (que depende do processo em si), que recebe o nome de resistência (R_{12} e R_o respectivamente), e as vazões de saída podem ser descritas por

$$q_{12} = \frac{h_1 - h_2}{R_{12}}$$

$$q_o = \frac{h_2}{R_o}$$

Consideramos também, para simplificar o problema, que a vazão de entrada q_i varia linearmente de acordo com a abertura da válvula apresentada na Figura 10.1. Dessa forma, podemos representar a vazão de entrada como

$$q_i = \bar{q}_i \delta$$

em que \bar{q}_i representa a vazão máxima de entrada, enquanto que $\delta \in [0, 1]$ representa a abertura da válvula (com 0 sendo a válvula totalmente fechada e 1 a válvula totalmente aberta).

Com todas essas considerações, temos que a dinâmica do sistema pode ser descrita pelas equações

$$\begin{aligned} \dot{h}_1 &= \frac{-1}{R_{12}C_1} h_1 + \frac{1}{R_{12}C_1} h_2 + \frac{\bar{q}_i}{C_1} \delta \\ \dot{h}_2 &= \frac{1}{R_{12}C_2} h_1 - \left(\frac{R_{12} + R_o}{R_{12}R_oC_2} \right) h_2 \end{aligned}$$

Simulação em Malha Aberta

Quando simulamos um sistema no MATLAB, fazemos uso das funções de integração numérica (sendo a mais comum a *ode45*, mas outras podem ser utilizadas de forma similar). Tais funções recebem o valor da derivada temporal dos estados (em função do estado atual e do instante de tempo atual) e retornam o valor dos estados ao longo do tempo.

O trecho de código abaixo apresenta uma função que pode ser utilizada para calcular a derivada temporal do sistema.

```

1 function x_dot = tanques(t,x,u,tu)
2 %retorna o valor da derivada temporal dada um tempo, o estado atual,
   e os vetores u e tu definindo a entrada aplicada sobre o sistema
3
4 %calcula o valor atual da entrada aplicada sobre o sistema por meio
   de interpolação linear
5 u_atual = interp1(tu,u,t);
6
7 %define os parâmetros do sistema
8 C1 = 2;
9 C2 = 4;
10 R12 = 1;
11 Ro = 10;
12 qi_max = 0.4;
13
14 %calcula a derivada temporal do sistema
15 x_dot = zeros(2,1);
16 x_dot(1) = -x(1)/(R12*C1) + x(2)/(R12*C1) + (qi_max/C1)*u_atual;
17 x_dot(2) = x(1)/(R12*C2) - (R12+Ro)*x(2)/(R12*Ro*C2);
18
19 end

```

O trecho de código abaixo utiliza esta função para simular o comportamento do sistema em malha aberta com a função *ode45*. Primeiramente, simula-se o comportamento do sistema no ponto de equilíbrio $h_1^* = 2.2, h_2^* = 2, u^* = 0.5$. Em seguida, simula-se o comportamento de uma variação em degrau na entrada de 0.5 para 0.8.

```

1 %vamos simular o comportamento do sistema para uma condição inicial
   em que h1 = 2.2, h2 = 2, e u = 0.5 (verificar equilíbrio do
   sistema)
2
3 %declara condição inicial e entrada aplicada sobre o sistema
4 x0 = [2.2; 2];

```

```

5 u = [0.5; 0.5]; tu = [0 1000];
6 %Simula o comportamento do sistema entre 0 e 10 segundos
7 t0 = 0; tf = 10;
8 [t,x] = ode45(@(t,x)tanques(t,x,u,tu),[t0 tf],x0);
9
10 %plota o comportamento do sistema
11 figure(1);
12 subplot(2,1,1), plot(t,x(:,1)); ylabel('h_1 (m)');
13 subplot(2,1,2), plot(t,x(:,2)); ylabel('h_2 (m)');
14 xlabel('t(s)');
15 set(gcf,'PaperUnits','centimeters','PaperSize',[16 8],'PaperPosition'
    ,[0 0 16 8]);
16 print -dpdf 'tanques_equilibrio';
17
18 %vamos agora testar o comportamento do sistema para uma entrada em
    degrau (mesmas condições iniciais)
19 u = [0.5; 0.5; 0.8; 0.8];
20 tu = [0; 2; 2.1; 1000];
21 %Simula o comportamento do sistema entre 0 e 300 segundos
22 t0 = 0; tf = 300;
23 [t,x] = ode45(@(t,x)tanques(t,x,u,tu),[t0 tf],x0);
24
25 %plota o comportamento do sistema
26 figure(2);
27 subplot(2,1,1), plot(t,x(:,1)); ylabel('h_1 (m)');
28 subplot(2,1,2), plot(t,x(:,2)); ylabel('h_2 (m)');
29 xlabel('t(s)');
30 set(gcf,'PaperUnits','centimeters','PaperSize',[16 8],'PaperPosition'
    ,[0 0 16 8]);
31 print -dpdf 'tanques_degrau';

```

Os gráficos gerados por esse código são apresentados nas Figuras 10.2 e 10.3.

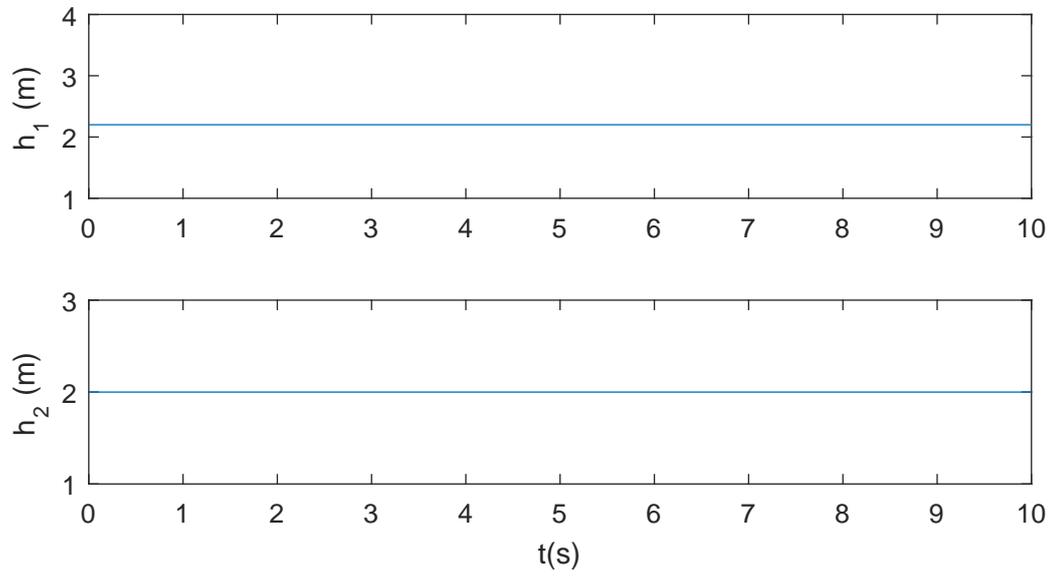


Figura 10.2: Comportamento do sistema de dois tanques em equilíbrio

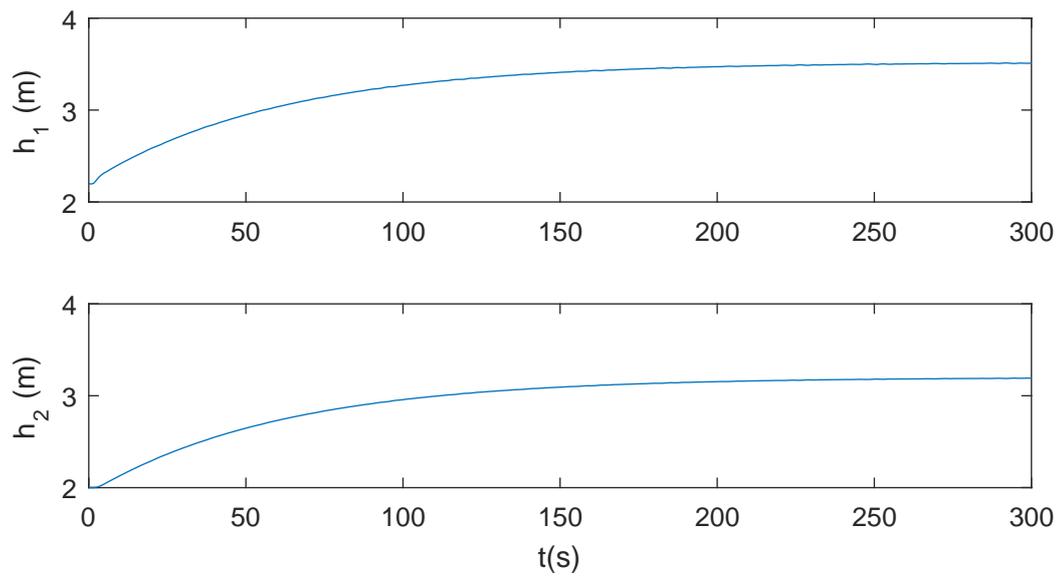


Figura 10.3: Resposta ao degrau do sistema de dois tanques

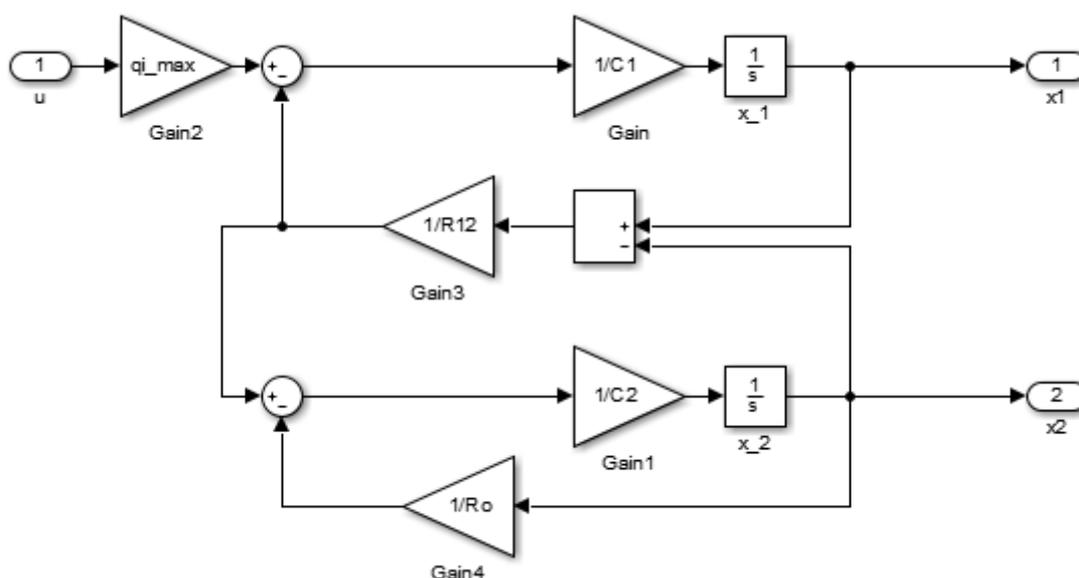


Figura 10.4: Diagrama de simulação do sistema de dois tanques. As condições iniciais são especificadas nos blocos de integradores.

Simulação em Malha Aberta - Simulink

Quando simulamos um sistema no Simulink, também fazemos uso de funções de integração numérica (de forma indireta ao desenharmos nosso diagrama de simulação). Para configurarmos a função de integração numérica utilizada, podemos ir em `Simulation > Model Configuration Parameters (ctrl+E)`.

Para simularmos sistemas contínuos, descritos por equações diferenciais, usamos blocos de integradores. Dessa forma, na entrada dos integradores colocamos a equação da derivada para cada um dos estados e na saída dos integradores teremos os valores dos estados.

Para simularmos sistemas discretos, descritos por equações de diferenças, usamos blocos de atraso da mesma forma que os blocos de integradores do caso contínuo.

Temos, basicamente, duas formas para escrevermos nosso diagrama de simulação. A primeira, apresentada na Figura 10.4, faz uso dos blocos mais básicos da biblioteca do Simulink.

A segunda forma, apresentada na Figura 10.5, faz uso de blocos que nos permitem rodar código do MATLAB em Simulink. No exemplo utilizamos o bloco “MATLAB Function” para utilizar um código do MATLAB. Tal bloco é compilado na hora da execução, e portanto pode não funcionar se o MATLAB não estiver configurado corretamente. Nesse caso, sugere-se o uso do bloco “Interpreted MATLAB Function”.

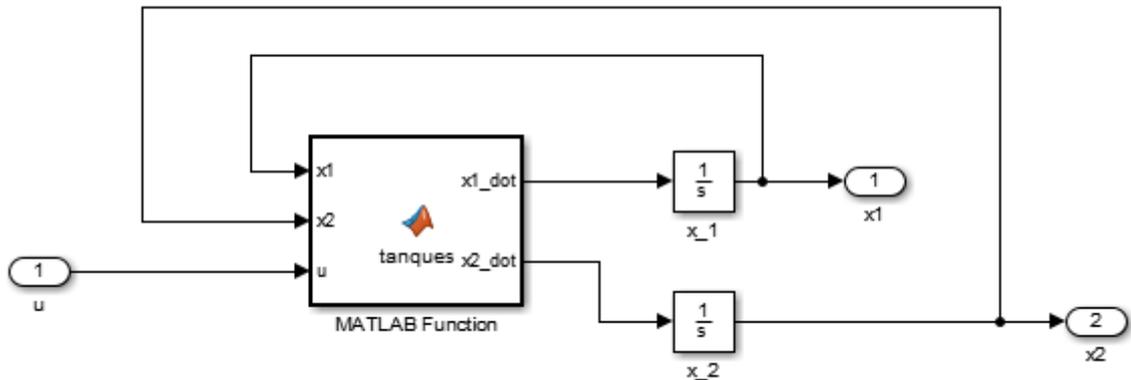


Figura 10.5: Outro diagrama de simulação do sistema de dois tanques. As condições iniciais são especificadas nos blocos de integradores.

O código utilizado no bloco é apresentado abaixo.

```

1 function [x1_dot,x2_dot] = tanques(x1,x2,u)
2 %#codegen
3
4 %define os parâmetros do sistema
5 C1 = 2;
6 C2 = 4;
7 R12 = 1;
8 Ro = 10;
9 qi_max = 0.4;
10
11 %calcula a derivada temporal do sistema
12 x1_dot = -x1/(R12*C1) + x2/(R12*C1) + (qi_max/C1)*u;
13 x2_dot = x1/(R12*C2) - (R12+Ro)*x2/(R12*Ro*C2);
14 end

```

Simulação em Malha Fechada

O objetivo nesse exemplo é o controle do nível do segundo tanque por meio da abertura da válvula de alimentação do sistema. Vamos testar duas estratégias diferentes, uma de controle “proporcional” e outra de controle “integral”.

Considere que temos duas regras

Regra 1: Se “nível é baixo”, então “abre a válvula”

Regra 2: Se “nível é alto”, então “fecha a válvula”

Considere que os conjuntos nebulosos de entrada (erro do nível) são dados por funções de pertinência trapezoidais, como apresentado na Figura 10.6, e que os conjuntos nebulosos de saída (abertura da válvula) são dados por funções de pertinência triangulares, como apresentado na Figura 10.7.

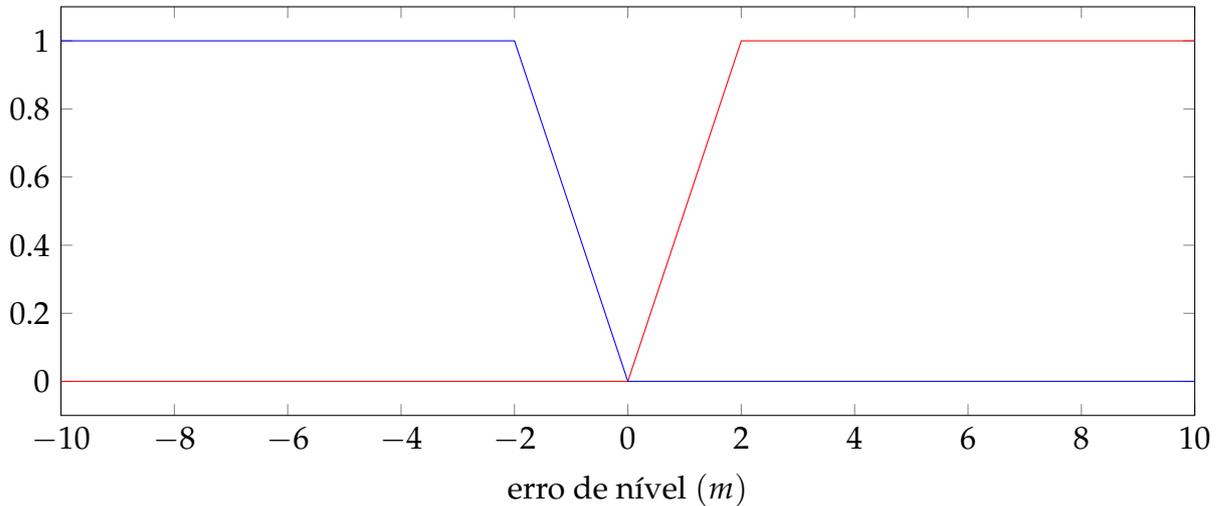


Figura 10.6: Conjuntos nebulosos para nível alto e baixo. A função de pertinência do conjunto BAIXO é apresentada em **vermelho**, enquanto que a função de pertinência do conjunto ALTO é apresentada em **azul**.

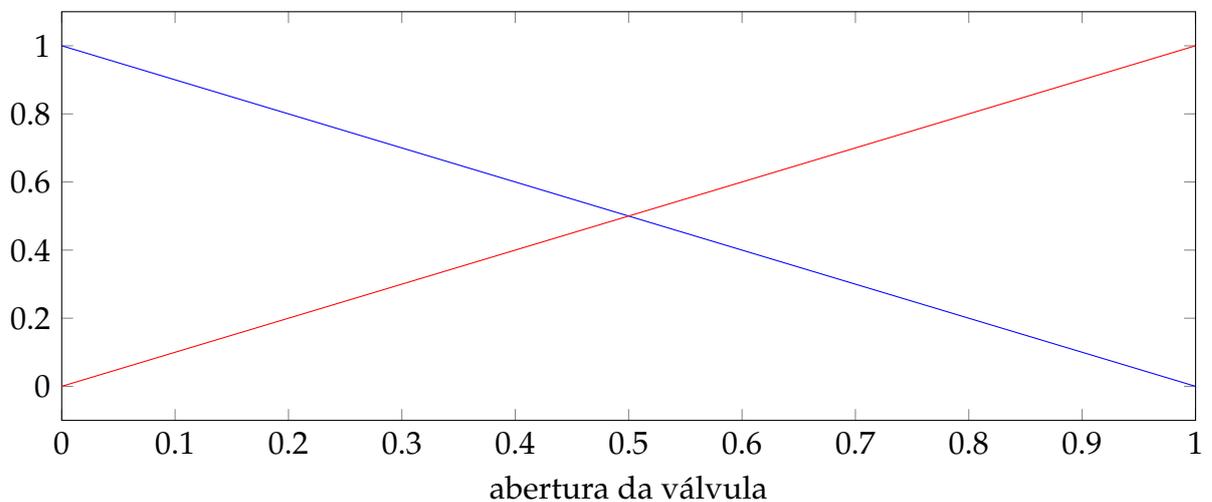


Figura 10.7: Conjuntos nebulosos para abertura da válvula. A função de pertinência do conjunto ABRE é apresentada em **vermelho**, enquanto que a função de pertinência do conjunto FECHA é apresentada em **azul**.

Se fizermos uso do sistema de inferência do produto, com um fuzzificador singleton

e um defuzzificador por centro de gravidade, temos que

$$\mu_{\tilde{B}}(y) = \max_{k=1..r} \left(\left(\prod_{i=1}^n \mu_{A_i^k}(x_i) \right) \mu_{B^k}(y) \right)$$

$$y = \frac{\int_V y \mu_{\tilde{B}}(y) dy}{\int_V \mu_{\tilde{B}}(y) dy}$$

Logo, nosso controlador nebuloso, pode ser escrito como

```

1 function [delta] = controlador_nebuloso(r, x2)
2 %#codegen
3     %calcula o erro de referência
4     erro = r-x2;
5     %encontra a saída pelo defuzzificador de centro de gravidade
6     y = linspace(0,1,100); %vamos aproximar a integral por um
       somatório
7     delta = 0; soma = 0; dy = y(2)-y(1);
8     for i = 1:length(y)
9         mu = inferencia_produto(erro, y(i));
10            soma = soma + mu*dy;
11            delta = delta + y(i)*mu*dy;
12     end
13     delta = delta/soma;
14 end
15
16 %faz a inferência do produto, considerando um fuzzificador singleton
17 function mu = inferencia_produto(erro, y)
18     %calcula as pertinências dos conjuntos de entrada
19     mu_A1 = trapmf(erro, [-inf -inf -2 0]); %nível alto
20     mu_A2 = trapmf(erro, [0 2 inf inf]); %nível baixo
21     %calcula as pertinências dos conjuntos de saída
22     mu_B1 = trimf(y, [-1 0 1]); %fecha
23     mu_B2 = trimf(y, [0 1 2]); %abre
24     %a pertinência de saída é dada pelo máximo dos produtos
25     mu = max(mu_A1*mu_B1, mu_A2*mu_B2);
26 end

```

Para deixar a simulação um pouco mais realista, vamos considerar que estamos utilizando um controlador discreto no tempo com um tempo de amostragem de 0.1s. Dessa forma, no MATLAB, podemos simular o comportamento do sistema fazendo (aproveitando o código que temos de malha aberta)

```

1 clearvars; close all; clc;
2 %vamos simular o comportamento do sistema para uma condição inicial
   em que h1 = 2.2, h2 = 2, e r = 2

```

```

3
4 %declara condição inicial , tempo inicial , final e tempo de amostragem
5 x0 = [0; 0]; ti = 0; tf = 200; ts = 0.1;
6 tr = ti:ts:tf; x = x0'; t = ti;
7 %itera sobre os tempos, calculando o valor da ação de controle
8 for i = 1:(length(tr)-1)
9     %define o intervalo de tempo atual
10    ta = [tr(i) tr(i+1)];
11    %define a referência atual – se o tempo atual for maior do
12    %que 100 segundos, mudamos a referência para 2.2
13    if (tr(i) > 100)
14        r = 2.2;
15    else
16        r = 2;
17    end
18    %calcula o valor atual da ação de controle
19    u = controlador_nebuloso(r,x(end,2))*ones(2,1);
20    tu = ta';
21    %Simula o comportamento do sistema para essa ação de controle
22    %e até o próximo instante de amostragem
23    [t_a, xa] = ode45(@(t,x) tanques(t,x,u,tu),ta,x(end,:));
24    t = [t; t_a];
25    x = [x; xa];
26 end
27 figure(1);
28 subplot(2,1,1), plot(t,x(:,1)); ylabel('h_1 (m)');
29 subplot(2,1,2), plot(t,x(:,2)); ylabel('h_2 (m)');
30 xlabel('t(s)');
31 set(gcf,'PaperUnits','centimeters','PaperSize',[16 8],'PaperPosition',
    ,[0 0 16 8]);
32 print -dpdf 'tanques_controlador_nebuloso';

```

O resultado da simulação do sistema é apresentado na Figura 10.8.

Para o controlador nebuloso integral, considere que temos duas regras

Regra 1: Se “nível é baixo”, então “abre a válvula”

Regra 2: Se “nível é alto”, então “fecha a válvula”

Considere que os conjuntos nebulosos de entrada (erro do nível) são dados por funções de pertinência trapezoidais, como apresentado na Figura 10.6, e que os con-

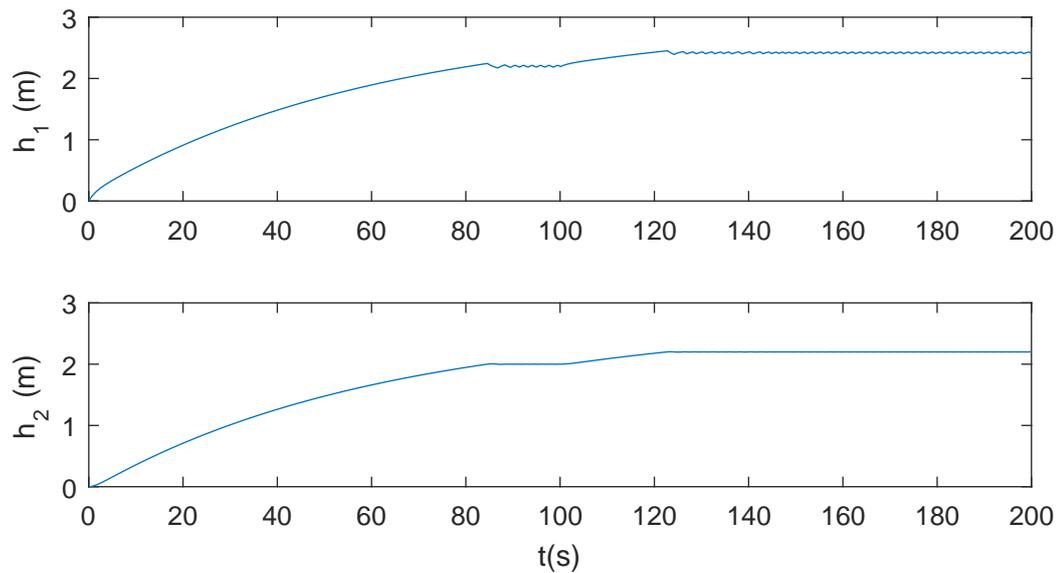


Figura 10.8: Comportamento do sistema de dois tanques com um controlador nebuloso

juntos nebulosos de saída (variação de abertura da válvula) são dados por funções de pertinência triangulares, como apresentado na Figura 10.9.

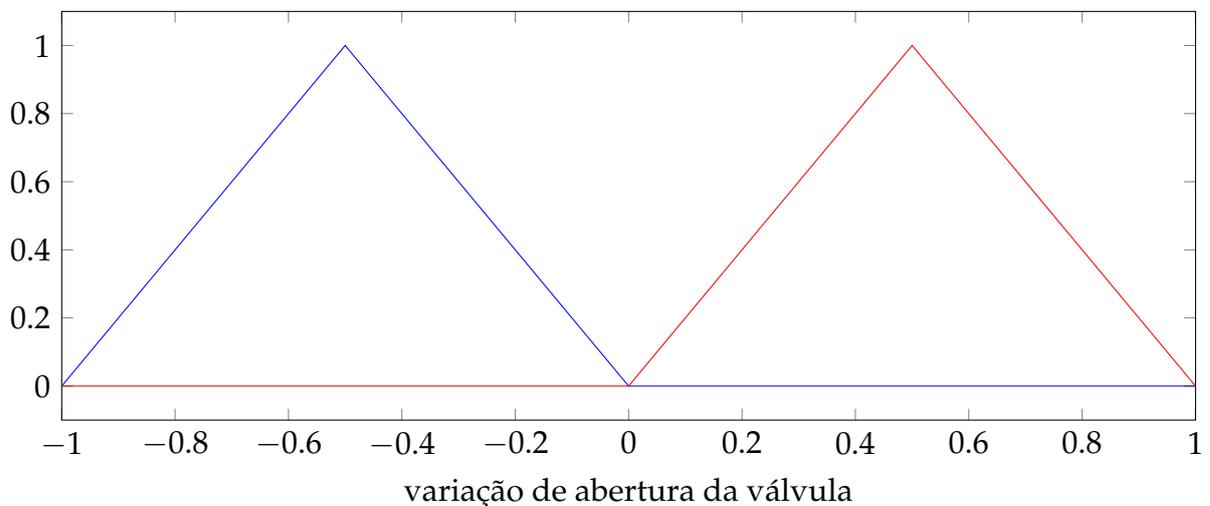


Figura 10.9: Conjuntos nebulosos para a variação de abertura da válvula. A função de pertinência do conjunto ABRE é apresentada em **vermelho**, enquanto que a função de pertinência do conjunto FECHA é apresentada em **azul**.

Se fizermos uso do sistema de inferência do produto, com um fuzzificador singleton e um defuzzificador por centro de gravidade, nosso controlador nebuloso, pode ser escrito como

```

1 function [delta] = controlador_nebuloso_delta(r,x2)
2 %#codegen
3     %calcula o erro de referência
4     erro = r-x2;

```

```

5      %encontra a saída pelo defuzzificador de centro de gravidade
6      y = linspace(-1,1,200); %vamos aproximar a integral por um
          somatório
7      delta = 0; soma = 0; dy = y(2)-y(1);
8      for i = 1:length(y)
9          mu = inferencia_produto(erro ,y(i));
10         soma = soma + mu*dy;
11         delta = delta + y(i)*mu*dy;
12     end
13     delta = delta/soma;
14 end
15
16 %faz a inferência do produto, considerando um fuzzificador singleton
17 function mu = inferencia_produto(erro ,y)
18     %calcula as pertinências dos conjuntos de entrada
19     mu_A1 = trapmf(erro ,[-inf -inf -2 0]); %nível alto
20     mu_A2 = trapmf(erro ,[0 2 inf inf]); %nível baixo
21     %calcula as pertinências dos conjuntos de saída
22     mu_B1 = trimf(y, [-1 -0.5 0]); %fecha
23     mu_B2 = trimf(y, [0 0.5 1]); %abre
24     %a pertinência de saída é dada pelo máximo dos produtos
25     mu = max(mu_A1*mu_B1,mu_A2*mu_B2);
26 end

```

Para deixar a simulação um pouco mais realista, vamos considerar que estamos utilizando um controlador discreto no tempo com um tempo de amostragem de 0.1s. Dessa forma, no MATLAB, podemos simular o comportamento do sistema fazendo (aproveitando o código que temos de malha aberta)

```

1 clearvars; close all; clc;
2 %vamos simular o comportamento do sistema para uma condição inicial
          em que h1 = 2.2, h2 = 2, e r = 2
3
4 %declara condição inicial, tempo inicial, final e tempo de amostragem
5 x0 = [0; 0]; ti = 0; tf = 200; ts = 0.1;
6 tr = ti:ts:tf; x = x0'; t = ti; u = 0;
7 %itera sobre os tempos, calculando o valor da ação de controle
8 for i = 1:(length(tr)-1)
9     %define o intervalo de tempo atual
10    ta = [tr(i) tr(i+1)];
11    %define a referência atual - se o tempo atual for maior do
          que 100 segundos, mudamos a referência para 2.2

```

```

12     if (tr(i) > 100)
13         r = 2.2;
14     else
15         r = 2;
16     end
17     %calcula o valor atual da ação de controle
18     u = u + controlador_nebuloso_delta(r,x(end),2));
19     %limita a ação de controle
20     if (u > 1)
21         u = 1;
22     elseif (u < 0)
23         u = 0;
24     end
25     ua = u*ones(2,1);
26     tu = ta';
27     %Simula o comportamento do sistema para essa ação de controle
        e até o próximo instante de amostragem
28     [t_a, xa] = ode45(@(t,x) tanques(t,x,ua,tu), ta, x(end,:));
29     t = [t; t_a];
30     x = [x; xa];
31 end
32
33 figure(2);
34 subplot(2,1,1), plot(t,x(:,1)); ylabel('h_1 (m)');
35 subplot(2,1,2), plot(t,x(:,2)); ylabel('h_2 (m)');
36 xlabel('t(s)');
37 set(gcf, 'PaperUnits', 'centimeters', 'PaperSize', [16 8], 'PaperPosition'
    , [0 0 16 8]);
38 print -dpdf 'tanques_controlador_nebuloso_integral';

```

O resultado da simulação do sistema é apresentado na Figura 10.10.

Simulação em Malha Fechada - Simulink

No Simulink, utilizamos um bloco de segurador de ordem zero para especificar a taxa de amostragem do controlador, e um bloco MATLAB function com o código do controlador (o mesmo apresentado para o MATLAB, acrescido das funções trimf e trapmf). O diagrama de simulação do sistema em malha fechada é apresentado na Figura 10.11.

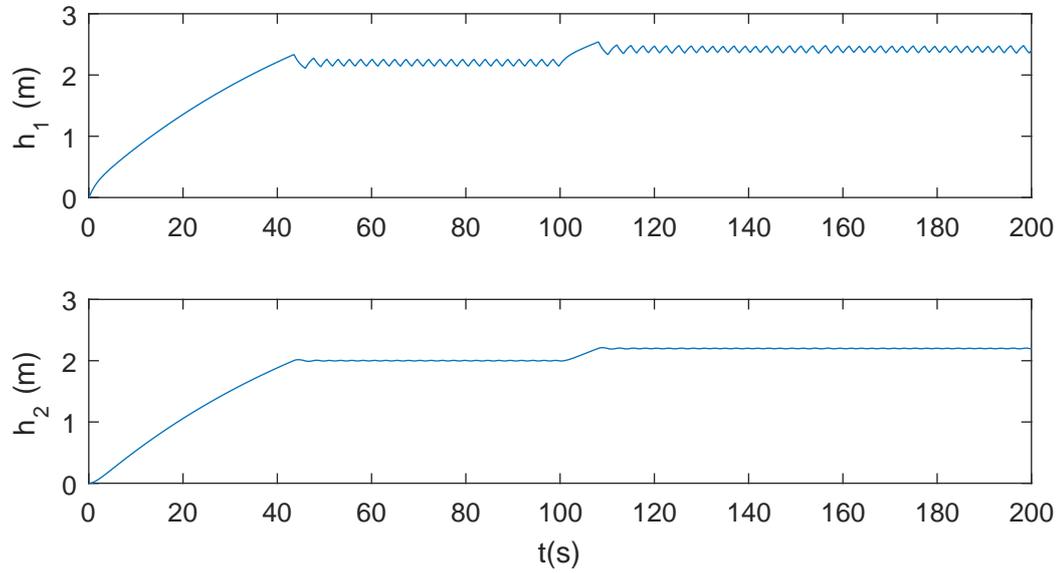


Figura 10.10: Comportamento do sistema de dois tanques com um controlador nebuloso integral

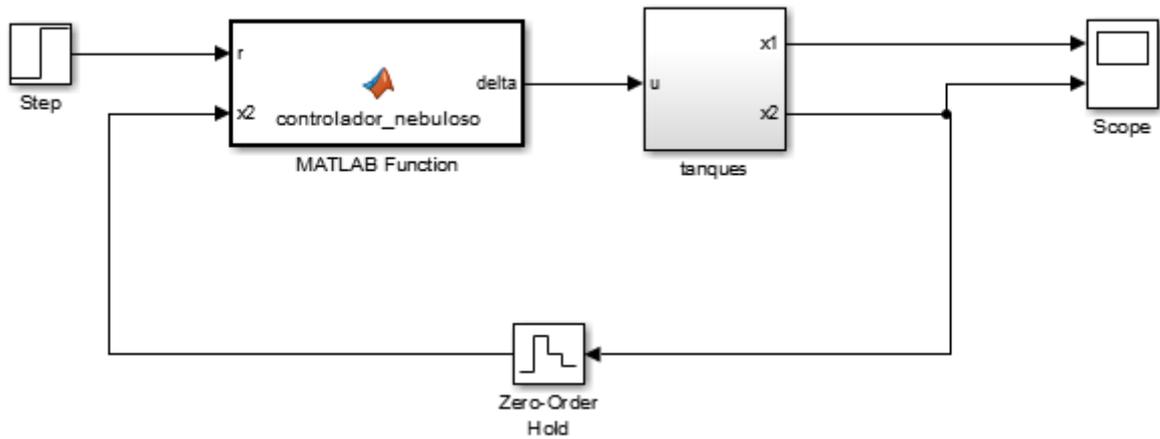


Figura 10.11: Diagrama de simulação do sistema de dois tanques em malha fechada.

Aula 11

Sistemas Takagi-Sugeno: Definições e Aproximadores Universais

Até o momento, vimos sistemas nebulosos cujos bancos de regras têm regras em que tanto o se como o então são dados por conjuntos nebulosos. É usual darmos o nome para esses sistemas de sistemas do tipo Mamdani.

Takagi-Sugeno

Sistemas Takagi-Sugeno usam regras um pouco diferentes. Ao invés de termos conjuntos nebulosos na parte então das regras, temos funções (que podem depender das entradas do sistema nebuloso e de outras variáveis externas).

Nesses sistemas, consideramos regras da forma:

Regra k : Se x_1 é A_1^k e x_2 é A_2^k e ... e x_n é A_n^k , então $y = f_k(x_1, x_2, \dots, x_n, \theta)$

Apesar de serem um pouco diferentes dos sistemas do tipo Mamdani, ainda podemos utilizar todas as ferramentas apresentadas até este ponto. Basta considerar que os conjuntos nebulosos de saída de cada regra são conjuntos singleton que tem pertinência igual a 1 no ponto que em estão, e zeros em todo o restante do universo de discurso.

Nota 11.1

Note que existe uma sutil diferença entre um sistema Takagi-Sugeno (TS) e um sistema Mamdani com a saída dada por um conjunto singleton: o conjunto singleton TS representa uma função enquanto que o conjunto singleton Mamdani representa um único ponto.

Apesar de terem sido criados no contexto de identificação de sistemas, os sistemas ou modelos TS encontram aplicações em diversas áreas (como veremos mais adiante na disciplina).

Quando utilizamos sistemas/modelos TS é usual utilizarmos um fuzzificador *singleton*, a inferência do produto e um defuzzificador de média central. Neste caso, podemos encontrar a saída da seguinte forma:

1. Encontramos a função de pertinência da parte “Se” de cada regra

$$\mu_{A_k}(\mathbf{x}) = \prod_{i=1}^n \mu_{A_i^k}(x_i)$$

2. Como estamos utilizando um fuzzificador *singleton*, e a parte “então” não é um conjunto nebuloso, podemos “calcular a pertinência de cada regra” utilizando simplesmente a função de pertinência do passo 1.
3. Note que o “centro” das funções de pertinência de cada regra é dado por $\mathbf{f}_k(x_1, x_2, \dots, x_n, \theta)$. Então utilizando o defuzzificador de média central, temos

$$\mathbf{y} = \frac{\sum_{k=1}^r \mu_{A_k}(\mathbf{x}) \mathbf{f}_k(x_1, x_2, \dots, \theta)}{\sum_{k=1}^r \mu_{A_k}(\mathbf{x})} = \sum_{k=1}^r h_k(\mathbf{x}) \mathbf{f}_k(x_1, x_2, \dots, x_n, \theta)$$

As funções $h_k(\mathbf{x})$ são dadas por $h_k(\mathbf{x}) = \frac{\mu_{A_k}(\mathbf{x})}{\sum_{i=1}^r \mu_{A_i}(\mathbf{x})}$, e podem ser entendidas como as funções de pertinência normalizadas de cada regra. Elas possuem as seguintes propriedades interessantes

$$h_k(\mathbf{x}) \geq 0, \quad \sum_{k=1}^r h_k(\mathbf{x}) = 1$$

Modelos TS como aproximadores universais

Como acabamos de ver, um modelo TS pode ser escrito como

$$\mathbf{y}(\mathbf{x}) = \sum_{k=1}^r h_k(\mathbf{x}) f_k(\mathbf{x}, \theta)$$

e, de forma intuitiva, podemos entender os modelos TS como uma combinação de vários modelos locais $\mathbf{y}_k(\mathbf{x}) = \mathbf{f}_k(\mathbf{x})$ em um único modelo.

Interpretando dessa forma, fica fácil ver que certas classes de funções podem ser representadas como modelos TS, como, por exemplo, as funções constantes por partes.

Funções constantes por partes: São funções que mantêm um valor fixo em uma determinada região do universo de discurso. Uma forma simples de representar tais funções como modelos TS é a seguinte: define-se uma regra para cada valor que a função pode assumir, determina-se que a saída de cada regra é $f_k(\mathbf{x}, \theta) = \mathbf{y}_k$ (o valor da função naquela região) e que as funções de pertinência de cada regra são dadas pela função indicadora de cada região, ou seja

$$h_k(\mathbf{x}) = \begin{cases} 1 & \text{se } \mathbf{x} \in \text{região}_k \\ 0 & \text{caso contrário} \end{cases}$$

Um exemplo de uma função constante por partes é apresentado na Figura 11.1. Note que podemos ter uma variedade maior de valores simplesmente adicionando mais regras ao nosso modelo TS.

É possível se mostrar (usando o Teorema do Valor Médio) que qualquer função continuamente diferenciável (C_1) pode ser arbitrariamente aproximada por uma função constante por partes dentro de um compacto (para nós, uma região limitada do espaço).

Interpolação de ordem zero

Uma forma simples de se aproximar uma função continuamente diferenciável qualquer por uma função constante por partes é por meio de uma interpolação de ordem zero.

O procedimento utilizado é o seguinte:

- Divida o universo de discurso em regiões
- Defina que o valor da função constante por partes em cada região é o valor da função no centro da região

Seja $\Delta \mathbf{x}_k$ a norma-2 máxima da distância entre um elemento do universo de discurso (pertencente à região k) e o centro da região k , pelo Teorema do Valor Médio, temos que a diferença entre a função aproximada, $g(\mathbf{x})$, e a interpolação de ordem zero, $\tilde{g}(\mathbf{x})$, é limitada por

$$\|g(\mathbf{x}) - \tilde{g}(\mathbf{x})\| \leq \Delta \mathbf{x}_k \max_{\mathbf{z} \in \text{região } k} \|\nabla g(\mathbf{z})\|$$

Note que o erro de aproximação pode ser arbitrariamente reduzido, reduzindo-se o tamanho das regiões no universo de discurso. Isso implica em um número maior de regiões/valores utilizados, o que corresponde a um número maior de regras no modelo TS.

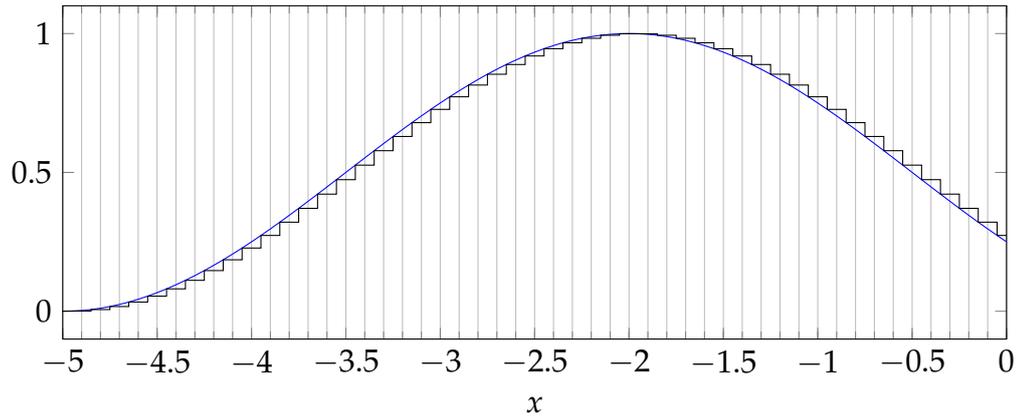


Figura 11.1: Exemplo de aproximação de função utilizando uma função constante por partes.

Quando um aproximador é tal que ele é capaz de aproximar qualquer função arbitrariamente bem, dizemos que ele é um **aproximador universal**.

Note que, apesar de podermos aproximar qualquer função C_1 arbitrariamente bem por meio de funções constantes por partes, tais aproximações também têm suas desvantagens. As principais delas sendo que o aproximador obtido é descontínuo, e que é necessário um número grande de regras para que se possa obter uma boa aproximação.

Exemplos de aproximadores obtidos com outros tipos de funções de pertinência

Funções de pertinência triangulares: O caso das funções de pertinência triangulares é bem geral e engloba uma classe de funções muito interessante: as funções contínuas lineares por partes. Para esse tipo de classe de funções, podemos considerar uma representação em que sempre existem apenas 2 funções de pertinência ativas (representando uma interpolação entre os valores das duas regras ativas).

Tais funções de pertinência podem ser descritas como:

$$\mu_{A_j^k}(x_j) = \begin{cases} \frac{x_j - x_{j(k-1)}}{x_{jk} - x_{j(k-1)}}, & x_j \in [x_{j(k-1)}, x_{jk}], \\ \frac{-x_j + x_{j(k+1)}}{x_{j(k+1)} - x_{jk}}, & x_j \in [x_{jk}, x_{j(k+1)}], \\ 0, & \text{caso contrário} \end{cases}$$

$$h_k(\mathbf{x}) = \prod_{j=1}^n \mu_{A_j^k}(x_j)$$

Um exemplo de aproximador usando funções de pertinência triangulares é apresentado na Figura 11.2.

Funções de pertinência gaussianas: Uma escolha muito popular é a de funções de pertinência gaussianas (por elas serem infinitamente diferenciáveis e intuitivas e,

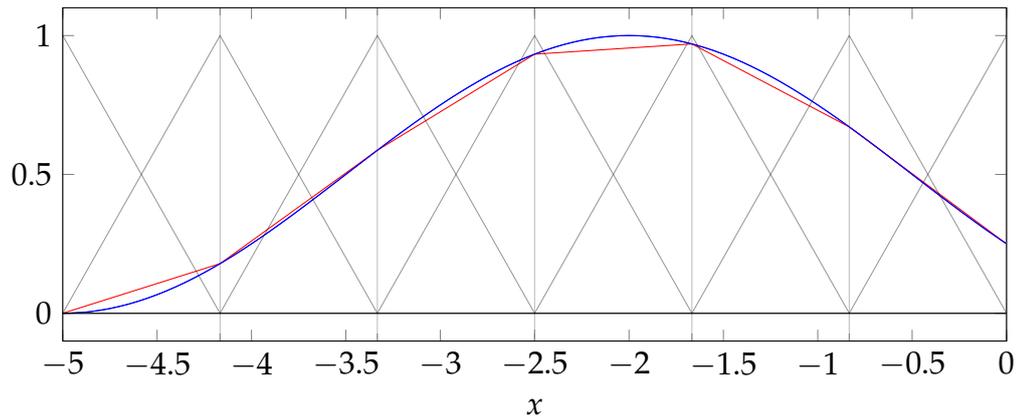


Figura 11.2: Exemplo de aproximação de função utilizando uma função linear por partes.

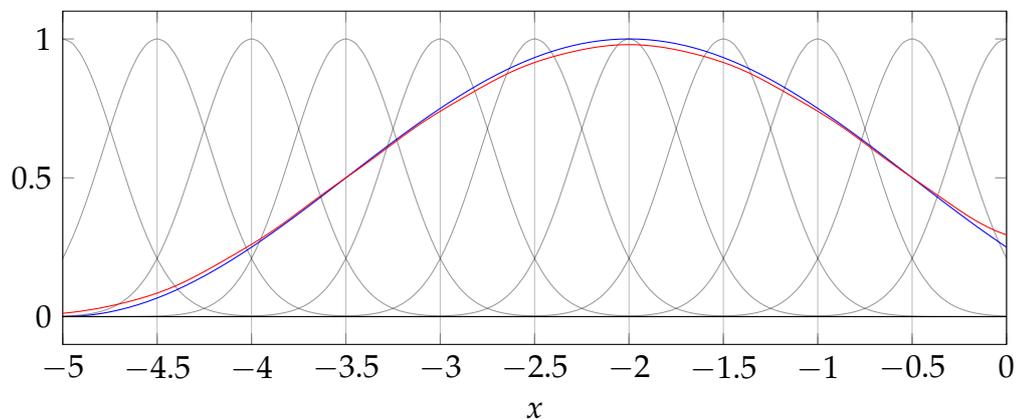


Figura 11.3: Exemplo de aproximação de função utilizando funções de pertinência gaussianas.

portanto, gerarem modelos infinitamente diferenciáveis e intuitivos). Entretanto, os modelos TS gerados com funções de pertinência gaussianas “perdem” uma característica dos modelos TS apresentados até o momento - o fato de as regras apenas terem efeito em uma região limitada do universo de discurso (devido ao suporte limitado das funções de pertinência acima). Além disso, todas as regras sempre estarão “ativas” (mesmo que com uma pertinência muito baixa) e não somos capazes de encontrar uma fórmula fechada para as funções $h_k(\mathbf{x})$ como nos casos anteriores (sempre precisamos calcular o somatório de todas as gaussianas no banco de regras).

Um exemplo de aproximador usando funções de pertinência gaussianas é apresentado na Figura 11.3.

Teorema de Stone-Weierstrass

Uma forma de verificarmos se uma classe de aproximadores possui a propriedade de aproximador universal, além do método mostrado para as funções constantes por partes, é pelo Teorema de Stone-Weierstrass.

Teorema 11.1: Teorema de Stone-Weierstrass

Seja Z um conjunto de funções contínuas em um conjunto compacto U .

Se

1. Z for uma álgebra (conjunto fechado sob adição, multiplicação e multiplicação por escalar)
2. Z separa pontos em U (para todo x e y pertencente a U tal que $x \neq y$, existe uma função $f \in Z$ com $f(x) \neq f(y)$)
3. Z desaparece em nenhum ponto de U (para todo x pertencente a U , existe uma função $f \in Z$ tal que $f(x) \neq 0$)

então para toda função real contínua $g(x)$ em U e um valor arbitrário ε , existe uma função f em Z tal que $\sup_{x \in U} \|f(x) - g(x)\| < \varepsilon$.

Note que o Teorema de Stone-Weierstrass garante que qualquer função contínua em U pode ser arbitrariamente aproximada em por uma função em Z . Isso pode parecer restrito (somente funções contínuas), mas lembre-se que qualquer função quadraticamente integrável (\mathcal{L}_2) em U pode ser arbitrariamente aproximada por uma função contínua.

É fácil ver que funções lineares por partes (ou polinomiais por partes de ordem mais elevada) atendem as restrições do Teorema. É um pouco mais difícil de ver que modelos TS com funções de pertinência gaussianas também o fazem, mas vamos tentar.

É fácil de ver que modelos com pertinências gaussianas são fechados sob a multiplicação por um escalar, separam os pontos do universo de discurso e não desaparecem em nenhum ponto.

Dadas duas funções

$$f_1(x) = \frac{\sum_{\ell=1}^{r_1} y_{1\ell} \prod_{i=1}^n a_{1i}^{\ell} e^{-\left(\frac{x_i - \bar{x}_{1i}^{\ell}}{\sigma_{1i}^{\ell}}\right)^2}}{\sum_{\ell=1}^{r_1} \prod_{i=1}^n a_{1i}^{\ell} e^{-\left(\frac{x_i - \bar{x}_{1i}^{\ell}}{\sigma_{1i}^{\ell}}\right)^2}}$$

$$f_2(x) = \frac{\sum_{\ell=1}^{r_2} y_{2\ell} \prod_{i=1}^n a_{2i}^{\ell} e^{-\left(\frac{x_i - \bar{x}_{2i}^{\ell}}{\sigma_{2i}^{\ell}}\right)^2}}{\sum_{\ell=1}^{r_2} \prod_{i=1}^n a_{2i}^{\ell} e^{-\left(\frac{x_i - \bar{x}_{2i}^{\ell}}{\sigma_{2i}^{\ell}}\right)^2}}$$

Temos que

$$f_1(x) + f_2(x) = \frac{\sum_{\ell=1}^{r_1} \sum_{\ell=1}^{r_2} (y_{1\ell} + y_{2\ell}) \prod_{i=1}^n a_{1i}^{\ell} a_{2i}^{\ell} e^{\left(-\left(\frac{x_i - \bar{x}_{1i}^{\ell}}{\sigma_{1i}^{\ell}}\right)^2 - \left(\frac{x_i - \bar{x}_{2i}^{\ell}}{\sigma_{2i}^{\ell}}\right)^2\right)}}{\sum_{\ell=1}^{r_1} \sum_{\ell=1}^{r_2} \prod_{i=1}^n a_{1i}^{\ell} a_{2i}^{\ell} e^{\left(-\left(\frac{x_i - \bar{x}_{1i}^{\ell}}{\sigma_{1i}^{\ell}}\right)^2 - \left(\frac{x_i - \bar{x}_{2i}^{\ell}}{\sigma_{2i}^{\ell}}\right)^2\right)}}$$

(as funções de pertinência se multiplicam pela propriedade de que a soma das funções de pertinência normalizadas é um)

E a soma das duas funções gera um modelo com funções de pertinência gaussianas (já que o produto de duas gaussianas é uma gaussiana) Além disso,

$$f_1(x)f_2(x) = \frac{\sum_{\ell=1}^{r_1} \sum_{\ell=1}^{r_2} (y_{1\ell}y_{2\ell}) \prod_{i=1}^n a_{1i}^{\ell} a_{2i}^{\ell} e^{\left(-\left(\frac{x_i - \bar{x}_{1i}^{\ell}}{\sigma_{1i}^{\ell}}\right)^2 - \left(\frac{x_i - \bar{x}_{2i}^{\ell}}{\sigma_{2i}^{\ell}}\right)^2\right)}}{\sum_{\ell=1}^{r_1} \sum_{\ell=1}^{r_2} \prod_{i=1}^n a_{1i}^{\ell} a_{2i}^{\ell} e^{\left(-\left(\frac{x_i - \bar{x}_{1i}^{\ell}}{\sigma_{1i}^{\ell}}\right)^2 - \left(\frac{x_i - \bar{x}_{2i}^{\ell}}{\sigma_{2i}^{\ell}}\right)^2\right)}}$$

e o produto gera um modelo com funções de pertinência gaussianas (já que o produto de duas gaussianas é uma gaussiana).

Logo temos que pelo Teorema de Stone-Weierstrass, modelos TS compostos com funções de pertinência gaussianas são aproximadores universais.

Como vimos na aula de hoje, várias escolhas de funções de pertinência diferentes levam a aproximadores universais. Dessa forma, se considerarmos modelos TS como um todo (permitindo qualquer função de pertinência) também temos uma classe de aproximadores universais.

Aula 12

Aproximação usando Modelos TS: Abordagem Linear nos Parâmetros

Na última aula, vimos que sistemas nebulosos Takagi-Sugeno possuem a propriedade de serem aproximadores universais. Nesta aula, veremos que, quando fixamos a estrutura do modelo de uma certa forma, obtemos um modelo linear nos parâmetros e podemos utilizar o método dos mínimos quadrados para aproximar seus valores.

Modelos Lineares nos Parâmetros e o Método dos Mínimos Quadrados

Dizemos que um modelo é linear nos parâmetros quando sua dependência dos parâmetros que desejamos determinar é linear.

Note que a função em si não precisa ser linear na variável que a define. Alguns exemplos de funções lineares nos parâmetros são

$$\begin{aligned}y_1 &= \theta_0 \\y_2 &= \theta_1 x + \theta_0 \\y_3 &= \theta_2 x^2 + \theta_1 x + \theta_0 \\y_4 &= \theta_1 e^{-(x-2)^2} \\y_5 &= \theta_1 \cos(x)\end{aligned}$$

Note que as funções apresentadas acima não são lineares em x mas são todas lineares nos parâmetros θ . Note que em todos os casos, sempre é possível reescrever as funções como

$$y = \mathbf{f}^T(\mathbf{x})\boldsymbol{\theta}$$

em que $\mathbf{f}^T(\mathbf{x})$ é um vetor linha, chamado de vetor de regressores, e $\boldsymbol{\theta}$ é um vetor contendo os parâmetros que desejamos determinar.

Nos exemplos apresentados, teríamos

$$\begin{aligned} y_1 &= \begin{bmatrix} 1 \end{bmatrix} \begin{bmatrix} \theta_0 \end{bmatrix} \\ y_2 &= \begin{bmatrix} x & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_0 \end{bmatrix} \\ y_3 &= \begin{bmatrix} x^2 & x & 1 \end{bmatrix} \begin{bmatrix} \theta_2 \\ \theta_1 \\ \theta_0 \end{bmatrix} \\ y_4 &= \begin{bmatrix} e^{-(x-2)^2} \end{bmatrix} \begin{bmatrix} \theta_1 \end{bmatrix} \\ y_5 &= \begin{bmatrix} \cos(x) \end{bmatrix} \begin{bmatrix} \theta_1 \end{bmatrix} \end{aligned}$$

Como queremos que nossos modelos reflitam algum dado que possuímos, normalmente, obtemos os parâmetros $\boldsymbol{\theta}$ a partir de dados de x e y correspondentes.

Uma abordagem possível, seria o uso de um número de dados (ou pares x e y) igual ao número de parâmetros que desejamos encontrar. Dessa forma, poderíamos montar um conjunto de equações lineares e encontrar os valores dos parâmetros.

Essa abordagem, entretanto, apresenta dois problemas principais: nosso modelo levará em consideração apenas os valores que utilizarmos, e tal procedimento considerará que os dados que temos são perfeitos (livres de ruídos ou diferenças entre o modelo que queremos levar e o “sistema real”).

Dessa forma, uma outra abordagem muito utilizada é a de se encontrar um modelo que seja o melhor (em relação a uma função de custo) considerando o maior número de dados que pudermos.

Considerando um modelo linear nos parâmetros, descrito por

$$y = \mathbf{f}^T(\mathbf{x})\boldsymbol{\theta}$$

e que temos k conjuntos de dados, descritos por (\mathbf{x}_i, y_i) consideramos uma função de custo da forma

$$J(\boldsymbol{\theta}) = \sum_{i=1}^k \left(y_i - \mathbf{f}^T(\mathbf{x}_i)\boldsymbol{\theta} \right)^2$$

Normalmente, é costume se representar este problema de forma matricial como

$$J(\boldsymbol{\theta}) = (\mathbf{y} - X\boldsymbol{\theta})^T(\mathbf{y} - X\boldsymbol{\theta})$$

em que \mathbf{y} é chamado de vetor de observações e contém em cada linha i a observação y_i , e X é chamada de matriz de regressores e contém em cada linha o vetor de regressores $\mathbf{f}^T(\mathbf{x}_i)$ correspondente aos valores de \mathbf{x}_i .

Como este problema de minimização não possui nenhuma restrição, podemos resolvê-lo igualando-se a derivada da função de custo a zero.

$$\begin{aligned}
 J(\boldsymbol{\theta}) &= (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) \\
 J(\boldsymbol{\theta}) &= \mathbf{y}^T \mathbf{y} - 2\boldsymbol{\theta}^T \mathbf{X}^T \mathbf{y} + \boldsymbol{\theta}^T \mathbf{X}^T \mathbf{X} \boldsymbol{\theta} \\
 \left(\frac{\partial J}{\partial \boldsymbol{\theta}} \right)^T &= -2\mathbf{X}^T \mathbf{y} + 2\mathbf{X}^T \mathbf{X} \boldsymbol{\theta} \\
 -2\mathbf{X}^T \mathbf{y} + 2\mathbf{X}^T \mathbf{X} \boldsymbol{\theta} &= 0 \\
 2\mathbf{X}^T \mathbf{X} \boldsymbol{\theta} &= 2\mathbf{X}^T \mathbf{y}
 \end{aligned}$$

Se a matriz \mathbf{X} tiver posto completo de colunas (ou seja, se o seu posto for igual ao número de parâmetros que desejamos determinar), então a matriz $\mathbf{X}^T \mathbf{X}$ tem posto completo e é portanto não-singular. Nesse caso, a solução do problema é dada por

$$\boldsymbol{\theta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (12.1)$$

A equação (12.1) representa o método dos mínimos quadrados e nos fornece uma forma de encontrarmos o modelo ótimo quando possuímos um grande conjunto de observações.

Modelos TS

Contanto que as funções de pertinências sejam previamente determinadas, e que as funções utilizadas no “então” de cada regra do modelo TS sejam lineares nos parâmetros, o modelo TS resultante será linear nos parâmetros.

Como vimos, os modelos TS podem ser escritos como

$$y(\mathbf{x}) = \sum_{k=1}^r h_k(\mathbf{x}) f_k(\mathbf{x}, \boldsymbol{\theta})$$

se considerarmos que as funções $f_k(\mathbf{x}, \boldsymbol{\theta})$ podem ser escritas como

$$f_k(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{g}_k^T(\mathbf{x}) \boldsymbol{\theta}_k$$

temos que o modelo resultante pode escrito como

$$\begin{aligned}
 y(\mathbf{x}) &= \sum_{k=1}^r h_k(\mathbf{x}) \mathbf{g}_k^T(\mathbf{x}) \boldsymbol{\theta}_k \\
 y(\mathbf{x}) &= h_1(\mathbf{x}) \mathbf{g}_1^T(\mathbf{x}) \boldsymbol{\theta}_1 + \dots + h_r(\mathbf{x}) \mathbf{g}_r^T(\mathbf{x}) \boldsymbol{\theta}_r \\
 y(\mathbf{x}) &= \begin{bmatrix} h_1(\mathbf{x}) \mathbf{g}_1^T(\mathbf{x}) & \dots & \dots & h_r(\mathbf{x}) \mathbf{g}_r^T(\mathbf{x}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_1 \\ \vdots \\ \boldsymbol{\theta}_r \end{bmatrix}
 \end{aligned}$$

No contexto de aproximação de funções, é muito comum utilizarmos funções polinomiais como o consequente (o “então”) de cada regra. Dessa forma, podemos interpretar nosso modelo como uma “mistura” de funções polinomiais. Mais especificamente, é muito comum utilizarmos polinômios de ordem zero (funções constantes) e de ordem um (funções afins).

No caso das funções constantes, temos apenas um parâmetro para determinar, e podemos escrever o modelo como

$$y(\mathbf{x}) = \begin{bmatrix} h_1(\mathbf{x}) & \dots & h_r(\mathbf{x}) \end{bmatrix} \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_r \end{bmatrix}$$

e temos r parâmetros a determinar no total.

No caso das funções afins, temos dois parâmetros para determinar, e podemos escrever o modelo como

$$y(\mathbf{x}) = \begin{bmatrix} h_1(\mathbf{x})x & h_1(\mathbf{x}) & \dots & h_r(\mathbf{x})x & h_r(\mathbf{x}) \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{2r-1} \\ \theta_{2r} \end{bmatrix}$$

e temos $2r$ parâmetros a determinar no total.

Exercício 12.1

Utilizando o método dos mínimos quadrados, encontre um modelo TS que aproxime a função

$$y = \text{sinc}(x) = \begin{cases} 1, & x = 0 \\ \frac{\text{sen}(x)}{x} & x \neq 0 \end{cases}$$

para $x \in [-2\pi, 2\pi]$. Utilize 100 pontos igualmente espaçados para encontrar os parâmetros do modelo. Considere modelos com 3, 5 e 10 funções de pertinência igualmente espaçadas triangulares e gaussianas. Além disso, considere que o consequente é dado por funções constantes e por funções lineares. Apresente os gráficos da função original versus o modelo obtido via mínimos quadrados para cada um dos casos. Comente sobre os modelos obtidos.

Mínimos Quadrados Recursivo

Quando a quantidade de dados que possuímos é muito grande, ou quando queremos estimar os parâmetros à medida que os dados vão ficando disponíveis, podemos utilizar uma versão recursiva do método dos mínimos quadrados.

Note que a solução de mínimos quadrados, dada pela equação (12.1), pode ser reescrita como

$$\boldsymbol{\theta}_k = \left(\sum_{i=1}^k \mathbf{f}(\mathbf{x}_i) \mathbf{f}^T(\mathbf{x}_i) \right)^{-1} \left(\sum_{i=1}^k \mathbf{f}(\mathbf{x}_i) \mathbf{y}_i \right)$$

Podemos definir

$$\begin{aligned} P_k &= \left(\sum_{i=1}^k \mathbf{f}(\mathbf{x}_i) \mathbf{f}^T(\mathbf{x}_i) \right)^{-1} \\ P_k^{-1} &= \left(\sum_{i=1}^{k-1} \mathbf{f}(\mathbf{x}_i) \mathbf{f}^T(\mathbf{x}_i) \right) + \mathbf{f}(\mathbf{x}_k) \mathbf{f}^T(\mathbf{x}_k) \\ P_k^{-1} &= P_{k-1}^{-1} + \mathbf{f}(\mathbf{x}_k) \mathbf{f}^T(\mathbf{x}_k) \end{aligned}$$

Aplicando o Lema da Inversão (*Igualdade de Woodbury*), temos

$$P_k = P_{k-1} - P_{k-1} \mathbf{f}(\mathbf{x}_k) \left(I + \mathbf{f}^T(\mathbf{x}_k) P_{k-1} \mathbf{f}(\mathbf{x}_k) \right)^{-1} \mathbf{f}(\mathbf{x}_k) P_{k-1}$$

e se $\mathbf{f}(\mathbf{x}_k) \in \mathbb{R}^{r \times 1}$ podemos simplificar para

$$P_k = P_{k-1} - \frac{P_{k-1} \mathbf{f}(\mathbf{x}_k) \mathbf{f}^T(\mathbf{x}_k) P_{k-1}}{1 + \mathbf{f}^T(\mathbf{x}_k) P_{k-1} \mathbf{f}(\mathbf{x}_k)}$$

Além disso, temos que

$$\begin{aligned} \boldsymbol{\theta}_k &= P_k \left(\sum_{i=1}^k \mathbf{f}(\mathbf{x}_i) \mathbf{y}_i \right) \\ P_k^{-1} \boldsymbol{\theta}_k &= \sum_{i=1}^k \mathbf{f}(\mathbf{x}_i) \mathbf{y}_i \\ P_{k-1}^{-1} \boldsymbol{\theta}_{k-1} &= \sum_{i=1}^{k-1} \mathbf{f}(\mathbf{x}_i) \mathbf{y}_i \end{aligned}$$

o que nos permite escrever

$$\begin{aligned} \boldsymbol{\theta}_k &= P_k \sum_{i=1}^k \mathbf{f}(\mathbf{x}_i) \mathbf{y}_i \\ \boldsymbol{\theta}_k &= P_k \left(\sum_{i=1}^{k-1} \mathbf{f}(\mathbf{x}_i) \mathbf{y}_i + \mathbf{f}(\mathbf{x}_k) \mathbf{y}_k \right) \\ \boldsymbol{\theta}_k &= P_k \left(P_{k-1}^{-1} \boldsymbol{\theta}_{k-1} + \mathbf{f}(\mathbf{x}_k) \mathbf{y}_k \right) \\ \boldsymbol{\theta}_k &= P_k \left(\left(P_{k-1}^{-1} - \mathbf{f}(\mathbf{x}_k) \mathbf{f}^T(\mathbf{x}_k) \right) \boldsymbol{\theta}_{k-1} + \mathbf{f}(\mathbf{x}_k) \mathbf{y}_k \right) \\ \boldsymbol{\theta}_k &= \boldsymbol{\theta}_{k-1} + P_k \mathbf{f}(\mathbf{x}_k) \left(\mathbf{y}_k - \mathbf{f}^T(\mathbf{x}_k) \boldsymbol{\theta}_{k-1} \right) \end{aligned}$$

De modo que o método dos mínimos quadrados recursivo pode ser escrito como

$$P_k = P_{k-1} - \frac{P_{k-1} \mathbf{f}(\mathbf{x}_k) \mathbf{f}^T(\mathbf{x}_k) P_{k-1}}{1 + \mathbf{f}^T(\mathbf{x}_k) P_{k-1} \mathbf{f}(\mathbf{x}_k)}$$

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + P_k \mathbf{f}(\mathbf{x}_k) \left(\mathbf{y}_k - \mathbf{f}^T(\mathbf{x}_k) \boldsymbol{\theta}_{k-1} \right)$$

Note que para utilizar este método, é necessário que sejam definidas condições iniciais para $\boldsymbol{\theta}_0$ e P_0 . Entretanto, é comum inicializar o método com $\boldsymbol{\theta}_0 = \mathbf{0}$ e $P_0 = \lambda I$, com λ um valor bem elevado entre 10^3 e 10^7 , de modo que os parâmetros encontrados se aproximem dos parâmetros encontrados pelo método tradicional (também chamado de método em *batelada*).

Nota 12.1

Note que as condições apresentadas para o mínimos quadrados recursivo acima exigem que a atualização da matriz P_k (também chamada de matriz de covariância) seja realizada antes da atualização dos parâmetros $\boldsymbol{\theta}_k$. Uma forma equivalente de representação, muito encontrada na literatura, pode ser dada por

$$K_k = \frac{P_{k-1} \mathbf{f}(\mathbf{x}_k)}{1 + \mathbf{f}^T(\mathbf{x}_k) P_{k-1} \mathbf{f}(\mathbf{x}_k)}$$

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + K_k \left(\mathbf{y}_k - \mathbf{f}^T(\mathbf{x}_k) \boldsymbol{\theta}_{k-1} \right)$$

$$P_k = P_{k-1} - K_k \mathbf{f}^T(\mathbf{x}_k) P_{k-1}$$

Exercício 12.2

Utilizando agora o método dos mínimos quadrados recursivos, repita o exercício 12.1, como se os pontos para x e y fossem obtidos um por um. Apresente os gráficos de evolução dos parâmetros estimados ao longo do tempo, e os compare com os parâmetros obtidos no exercício 12.1. Apresente também o gráfico do traço de P_k (soma dos valores da diagonal) ao longo do tempo. Comente sobre o efeito da escolha das condições iniciais.

Unidade II

Sistemas Não-Lineares

Aula 13

Sistemas Dinâmicos Não Lineares em Espaço de Estados

Nesta parte da matéria vamos focar nossos estudos principalmente na estabilidade de sistemas não-lineares.

Vamos tentar levantar formas mais gerais de se estudar a estabilidade de sistemas do que apenas pelos seus polos (pois essa relação só existe quando o sistema é linear), e que possam ser aplicadas para um grande conjunto de sistemas dinâmicos.

Consideramos sistemas cujo comportamento é descrito por um sistema de equações diferenciais de primeira ordem

$$\begin{aligned}\dot{x}_1 &= f_1(t, x_1, \dots, x_n, u_1, \dots, u_m) && \vdots \\ \dot{x}_n &= f_n(t, x_1, \dots, x_n, u_1, \dots, u_m)\end{aligned}$$

Com t sendo o tempo, x_1 a x_n as variáveis de estado (variáveis que descrevem a memória do sistema) e u_1 a u_m as variáveis de entrada do sistema.

Normalmente, utilizamos uma representação vetorial para escrever as equações do sistema de forma mais compacta

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}, \quad \mathbf{f}(t, \mathbf{x}, \mathbf{u}) = \begin{bmatrix} f_1(t, \mathbf{x}, \mathbf{u}) \\ \vdots \\ f_n(t, \mathbf{x}, \mathbf{u}) \end{bmatrix}$$

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u})$$

Em alguns casos, é interessante especificar quais são as saídas do sistema (em função dos estados e das entradas) e uma outra equação é utilizada

$$\mathbf{y} = \mathbf{h}(t, \mathbf{x}, \mathbf{u})$$

(Normalmente chamamos o primeiro conjunto de equações de *equações de estado* e o segundo conjunto de *equações de saída*).

No caso em que temos um sistema linear (um sistema para o qual vale o *princípio da superposição*), os coeficientes do sistema de equações diferenciais de primeira ordem são sempre independentes dos estados, e podemos representar as equações de estado e saída de forma vetorial como

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

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

(E o comportamento do sistema pode ser completamente descrito pelas matrizes A , B , C e D).

Uma forma muito comum de se obter modelos lineares para o comportamento do sistema é por meio da *linearização* das equações não-lineares que descrevem o comportamento do sistema.

Em uma grande parte dessa unidade, vamos lidar com sistemas em que o sinal de entrada não aparece explicitamente na equação (pode ser que ele foi representado como um sinal do tempo ou então como uma função dos estados e substituído na equação dinâmica do sistema), e podemos escrever a equação do comportamento dinâmico do sistema como

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x})$$

De forma um pouco mais específica, muitas vezes lidaremos com sistemas em que o tempo não tem nenhuma influência explícita sobre o comportamento do sistema. Nesse caso, dizemos que o sistema é *autônomo* ou *invariante no tempo* e podemos escrever seu comportamento como

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$

No caso em que o comportamento do sistema depende de forma explícita do tempo, dizemos que o sistema é *não-autônomo* ou *variante no tempo*.

Um conceito importante quando lidamos com a equação de estados é o conceito de *ponto de equilíbrio*. Um ponto \mathbf{x}^* é chamado de ponto de equilíbrio do sistema se quando o estado do sistema está naquele ponto, ele permanece naquele ponto para todos os estados futuros.

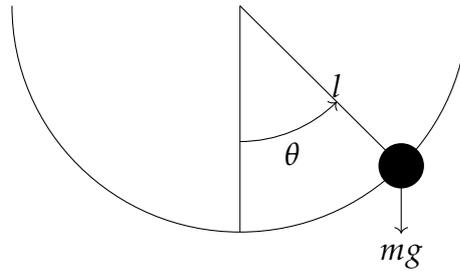


Figura 13.1: Pêndulo simples, adaptado de (Khalil 2002).

No caso de um sistema autônomo, os pontos de equilíbrio são as raízes de

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}$$

porque isso implica em $\dot{\mathbf{x}} = \mathbf{0}$.

Exemplo 13.1 (Pêndulo simples com atrito). Considere o pêndulo apresentado na Figura 13.1. Pela segunda lei de Newton, sabemos que

$$ml\ddot{\theta} = -mg \sin(\theta) - kl\dot{\theta}$$

Para estudarmos o comportamento deste sistema, devemos colocá-lo na forma apresentada acima. Escolhendo como estados $x_1 = \theta$ e $x_2 = \dot{\theta}$, as equações de estado são

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{g}{l} \sin(x_1) - \frac{k}{m} x_2 \end{aligned}$$

Para encontrarmos os pontos de equilíbrio do sistema, basta igualarmos ambas as equações a zero. Desta forma, encontramos que os pontos de equilíbrio do sistema são dados por $\begin{bmatrix} n\pi \\ 0 \end{bmatrix}$, com $n \in \mathbb{Z}$.

Estabilidade no sentido de Lyapunov (estabilidade interna)

Na grande maioria das matérias de modelagem e controle que vemos na graduação, estamos preocupados com a estabilidade entrada/saída de um sistema.

Mais especificamente, estudamos a estabilidade BIBO (*bounded input / bounded output*) que define que um sistema é estável quando, para qualquer entrada limitada aplicada ao sistema, ele gera uma saída limitada. Note que este conceito de estabilidade ignora completamente o comportamento interno (dos estados) do sistema e só lida com o comportamento entrada/saída do mesmo.

Nesta disciplina (na maioria das vezes) estamos mais preocupados com a estabilidade interna do *ponto de equilíbrio de um sistema*, ou seja, com o comportamento dos estados em torno de um ponto de equilíbrio. Além disso, sempre iremos assumir (sem perda de generalidade) que o ponto de equilíbrio sendo estudado é a origem do espaço de estados (porque caso não seja, basta fazer uma transformação de coordenadas).

Dessa forma, vamos definir o que queremos dizer com a estabilidade de um ponto de equilíbrio.

Estabilidade de um ponto de equilíbrio (considerando análise da origem)

Um ponto de equilíbrio é dito *estável* se, para todo $\varepsilon > 0$, existir um $\delta > 0$, tal que

$$\|\mathbf{x}(0)\| < \delta \Rightarrow \|\mathbf{x}(t)\| < \varepsilon, \forall t$$

Instável quando ele não é estável, e *assintoticamente estável* quando ele é estável e pode-se escolher δ tal que

$$\|\mathbf{x}(0)\| < \delta \Rightarrow \lim_{t \rightarrow \infty} \mathbf{x}(t) = \mathbf{0}$$

Na Figura 13.2, temos um exemplo de um sistema estável. Na Figura 13.3, temos um exemplo de um sistema instável. Na Figura 13.4 temos um exemplo de um sistema assintoticamente estável.

Uma forma de se verificar se um ponto de equilíbrio é estável ou não é calculando-se todas as trajetórias possíveis em torno de um ponto de equilíbrio e verificando se elas atendem as definições de estabilidade. Essa abordagem pode-se mostrar muito complicada ou até mesmo impossível em alguns casos.

Felizmente, existe uma outra possibilidade para se verificar a estabilidade de um ponto de equilíbrio sem ter de calcular as soluções da equação diferencial do sistema. Tal possibilidade é fazer uso dos *Teoremas de Estabilidade de Lyapunov*.

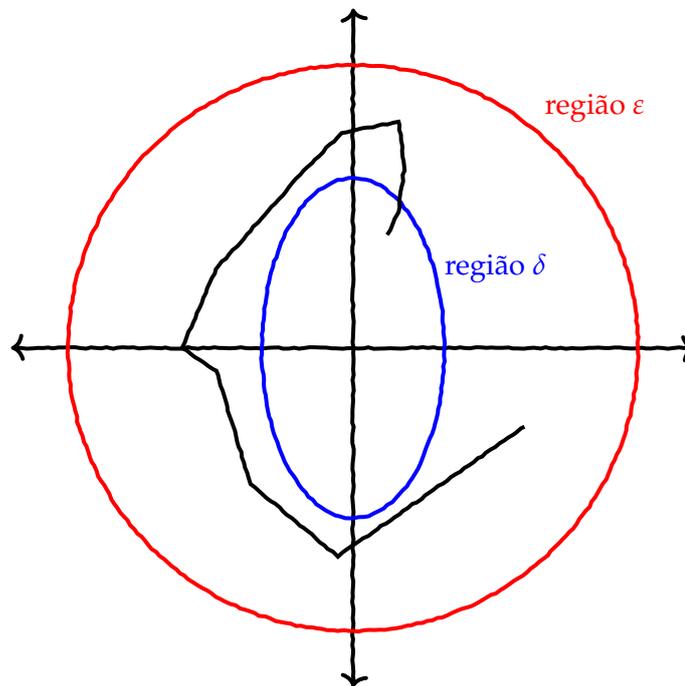


Figura 13.2: Exemplo de ponto de equilíbrio estável. Uma trajetória que se inicia na região δ se mantém na região ε . A curva em preto representa um exemplo de trajetória.

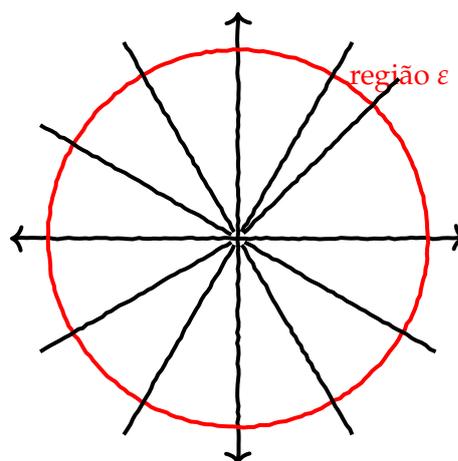


Figura 13.3: Exemplo de ponto de equilíbrio instável. As curvas em preto representam exemplos de trajetória do sistema. Não existe uma região δ tal que se o sistema se inicia nela ele permanece na região ε .

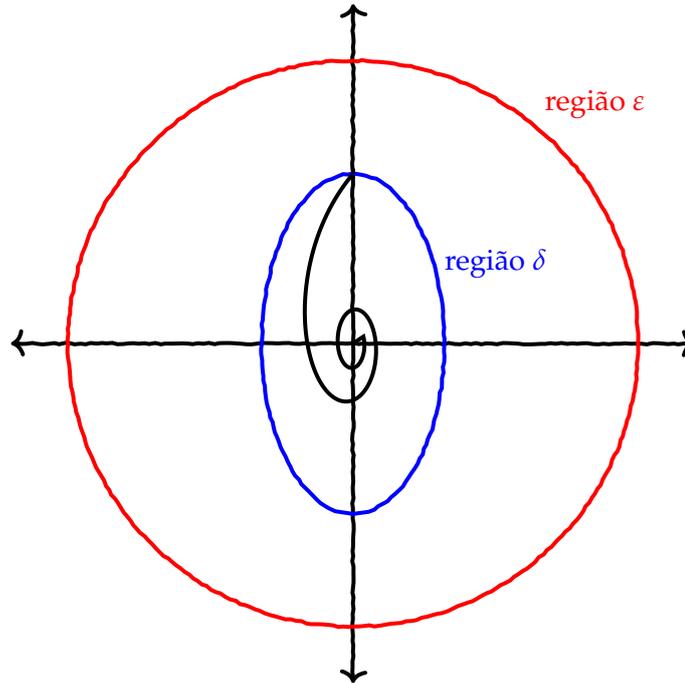


Figura 13.4: Exemplo de ponto de equilíbrio assintoticamente estável. Uma trajetória que se inicia na região δ se mantém na região ϵ . Além disso, a trajetória converge assintoticamente para o ponto de equilíbrio. A curva em preto representa um exemplo de trajetória.

Teorema 13.1: Teorema de Estabilidade de Lyapunov (Método Direto) - Sistemas Autônomos

Seja a origem ($\mathbf{x} = \mathbf{0}$) um ponto de equilíbrio do sistema, e $D \subset \mathbb{R}^n$ uma região incluindo a origem. Se existir uma função $V(\mathbf{x}) : D \rightarrow \mathbb{R}$ tal que

$$V(\mathbf{0}) = 0, V(\mathbf{x}) > 0 \quad \forall \mathbf{x} \neq \mathbf{0} \text{ (função definida positiva)}$$

$$\dot{V}(\mathbf{x}) \leq 0 \text{ (derivada semidefinida negativa)}$$

Então o ponto de equilíbrio é estável no sentido de Lyapunov. Se, ao invés disso,

$$\dot{V}(\mathbf{0}) = 0, \dot{V}(\mathbf{x}) < 0 \text{ (derivada definida negativa)}$$

Então o ponto de equilíbrio é assintoticamente estável.

Uma função que atende as condições do teorema, é chamada de *função de Lyapunov*.

O teorema acima diz que se encontrarmos uma função semelhante à energia do sistema (definida positiva) cuja derivada temporal é sempre menor ou igual a zero, então a origem do sistema é estável. Se essa derivada for sempre menor do que zero, então o sistema é assintoticamente estável.

Exemplo 13.2 (Pêndulo Simples com atrito).

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -a \operatorname{sen}(x_1) - bx_2, \\ a &= \frac{g}{l}, \quad b = \frac{k}{m}\end{aligned}$$

Escolhendo a função

$$V(\mathbf{x}) = \frac{1}{2}b^2x_1^2 + bx_1x_2 + x_2^2 + a(1 - \cos(x_1))$$

temos que $V(\mathbf{0}) = 0$, $V(\mathbf{x}) > 0 \forall \mathbf{x} \neq \mathbf{0}$ e que

$$\dot{V}(\mathbf{x}) = -\frac{1}{2}abx_1 \operatorname{sen}(x_1) - \frac{1}{2}bx_2^2 < 0,$$

para $|x_1| < \pi$, e escolhendo $D = \{\mathbf{x} \in \mathbb{R}^2 \mid |x_1| < \pi\}$ conseguimos provar que o sistema é assintoticamente estável nesta região.

Quando temos que um ponto de equilíbrio é assintoticamente estável, é interessante caracterizarmos qual a *região de atração* (ou *domínio de atração* ou *bacia de atração*) daquele ponto (a região δ mais larga para a qual os estados convergem para o ponto de equilíbrio).

Do Teorema de estabilidade de Lyapunov, temos que uma estimativa do domínio de atração de um ponto de equilíbrio assintoticamente estável é dada pela maior curva/conjunto de nível da função de Lyapunov na região D . Ou seja, o maior c tal que $V(\mathbf{x}) < c$, $\forall \mathbf{x} \in D$.

Quando o domínio de atração de um ponto de equilíbrio é todo o espaço de estados, dizemos que aquele ponto de equilíbrio é *globalmente assintoticamente estável* (quando não dizemos o globalmente, fica implícito que a estabilidade é local).

Para verificar a estabilidade global de um ponto de equilíbrio, não basta tomar $D = \mathbb{R}^n$ no teorema. Além disso, a função de Lyapunov deve ser *radialmente ilimitada*

$$\lim_{\|\mathbf{x}\| \rightarrow \infty} V(\mathbf{x}) \rightarrow \infty$$

Por fim, vale mencionar o método indireto de Lyapunov para a análise de estabilidade de pontos de equilíbrio de sistemas não-lineares

Teorema 13.2: Teorema de Estabilidade de Lyapunov - Método Indireto (sistemas autônomos)

Seja um sistema autônomo descrito por

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$

Se a sua linearização em torno de um ponto de equilíbrio \mathbf{x}_{eq}

$$\Delta \dot{\mathbf{x}} \approx \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_{eq}} \Delta \mathbf{x} = A \Delta \mathbf{x}$$

For assintoticamente estável, então o ponto de equilíbrio é assintoticamente estável (em uma região em torno do ponto de equilíbrio). Se ela for instável, então o ponto de equilíbrio é instável. Se ela for somente estável no sentido de Lyapunov (algum autovalor com parte real nula) então não podemos tirar nenhuma conclusão.

Exemplo 13.3 (Pêndulo simples com atrito). Considere novamente o pêndulo simples com atrito

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -a \operatorname{sen}(x_1) - bx_2 \end{aligned}$$

Vamos investigar a estabilidade do sistema pelo método indireto em dois pontos diferentes ($x_1 = 0, x_2 = 0$) e ($x_1 = \pi, x_2 = 0$).

A matriz jacobiana do sistema é dada por

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a \cos(x_1) & -b \end{bmatrix}$$

No ponto de equilíbrio com $x_1 = 0$

$$A = \begin{bmatrix} 0 & 1 \\ -a & -b \end{bmatrix}$$

e os autovalores são $-1/2b \pm 1/2\sqrt{b^2 - 4a}$, logo o ponto de equilíbrio é assintoticamente estável (pois os dois autovalores tem parte real negativa).

No ponto de equilíbrio $x_1 = \pi$

$$A = \begin{bmatrix} 0 & 1 \\ a & -b \end{bmatrix}$$

e os autovalores são $-1/2b \pm 1/2\sqrt{b^2 + 4a}$, logo o ponto de equilíbrio é instável (pois um dos autovalores sempre tem parte real positiva).

Aula 14

Estabilidade de Sistemas Dinâmicos Não-Lineares

Na última aula, aprendemos que o Teorema de Estabilidade de Lyapunov nos permite afirmar que um sistema é estável ou assintoticamente estável se encontramos uma função de Lyapunov para ele (uma função definida positiva e que tem derivada temporal semidefinida/definida negativa).

Em muitos casos, entretanto, procuramos por uma função de Lyapunov que prove que o ponto de equilíbrio é assintoticamente estável, mas encontramos apenas uma função de Lyapunov que prova a sua estabilidade.

Exemplo 14.1 (Pêndulo Simples - Função de Lyapunov dada pela energia mecânica). Considere a equação de um pêndulo simples com atrito

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{g}{l} \operatorname{sen}(x_1) - \frac{k}{m} x_2\end{aligned}$$

A energia cinética do pêndulo é dada por $E_c = \frac{1}{2}mv^2 = \frac{1}{2}m(\omega l)^2 = \frac{1}{2}ml^2\dot{x}_2^2$, enquanto que a energia potencial é dada por $E_p = mgh = mg(1 - \cos \theta)l = mgl(1 - \cos(x_1))$.

Uma função de Lyapunov natural que podemos pensar para o sistema é dada pela energia mecânica do sistema (note que ela é uma função definida positiva e é zero para o ponto de equilíbrio na origem) que é dada pela soma das energias potencial e

cinética.

$$V(\mathbf{x}) = mgl(1 - \cos(x_1)) + \frac{1}{2}ml^2x_2^2$$

$$\dot{V}(\mathbf{x}) = mgl(\sin(x_1))\dot{x}_1 + ml^2x_2\dot{x}_2$$

$$\dot{V}(\mathbf{x}) = mglx_2(\sin(x_1)) - mglx_2(\sin(x_1)) - kl^2x_2^2$$

$$\dot{V}(\mathbf{x}) = -kl^2x_2^2 \leq 0$$

E só podemos afirmar que o ponto de equilíbrio é estável (mas sabemos que ele é assintoticamente estável, pela função de Lyapunov que utilizamos na última aula).

Em alguns casos, mesmo que a função de Lyapunov prove apenas a estabilidade do sistema, podemos concluir a sua estabilidade assintótica. Para isso, fazemos uso do *Princípio da Invariância de LaSalle*.

Princípio da Invariância de LaSalle - Sistemas Autônomos

Para podermos enunciar e explicar o princípio da invariância, precisamos antes definir alguns conceitos.

Definição 14.1: Conjunto Invariante

Um conjunto é chamado de invariante, em relação à dinâmica do sistema, se

$$\mathbf{x}(0) \in M \Rightarrow \mathbf{x}(t) \in M, \forall t$$

Ou seja, se uma solução pertence ao conjunto M em algum momento, ele pertence àquele conjunto para todos os instantes de tempo (passados e futuros).

Definição 14.2: Conjunto Positivamente Invariante

Um conjunto é chamado de positivamente invariante, em relação à dinâmica do sistema, se

$$\mathbf{x}(0) \in M \Rightarrow \mathbf{x}(t) \in M, t \geq 0$$

Ou seja, se os estados atuais do sistema pertencem ao conjunto, todos os estados futuros pertencerão.

Definição 14.3: Solução aproximando-se de um conjunto quando o tempo tende a infinito

Dizemos que uma solução se aproxima de um conjunto à medida que o tempo tende a infinito quando, para todo $\varepsilon > 0$, existe um $T > 0$ tal que

$$\text{dist}(\mathbf{x}(t), M) < \varepsilon, \forall t > T$$

Com essas definições, estamos prontos para enunciar o Princípio da Invariância de LaSalle.

Teorema 14.1: Princípio da Invariância de LaSalle

Seja $\Omega \subset D$ um conjunto compacto positivamente invariante com respeito à dinâmica do sistema. Seja $V : D \rightarrow \mathbb{R}$ uma função continuamente diferenciável tal que $\dot{V}(\mathbf{x}) \leq 0$ em Ω . Seja E o conjunto de todos os pontos em Ω em que $\dot{V}(\mathbf{x}) = 0$. Seja M o maior conjunto invariante em E . Então toda solução (trajetória no espaço de estados) que se inicia em Ω se aproxima de M a medida que $t \rightarrow \infty$.

Intuitivamente, o princípio da invariância de LaSalle nos diz que os pontos em Ω convergem para o maior conjunto invariante em que $\dot{V}(\mathbf{x}) = 0$ (porque se $\dot{V}(\mathbf{x}) < 0$ então os pontos não ficam parados, e se um conjunto não é positivamente invariante, então mesmo com $\dot{V}(\mathbf{x}) = 0$ as soluções deixam aquele conjunto).

Note que, diferentemente do Teorema de Estabilidade de Lyapunov, o princípio da invariância não impõe que a matriz seja definida positiva, e que o conjunto Ω não precisa estar amarrado à função $V(\mathbf{x})$.

Nos casos em que a função é definida positiva, pelo Teorema de Estabilidade de Lyapunov, sabemos que o ponto de equilíbrio é estável em D , e que o maior conjunto de nível da função de Lyapunov $\Omega_c = \{\mathbf{x} \in D | V(\mathbf{x}) \leq c\}$, com c o maior possível, é um conjunto compacto (um conjunto limitado e fechado) positivamente invariante (pois como $\dot{V}(\mathbf{x}) \leq 0$, os estados nunca crescem de valor e os conjuntos de nível que não saem da região D são garantidos de serem positivamente invariantes).

Dessa forma, para aplicarmos o princípio da invariância para provar a estabilidade assintótica de um ponto de equilíbrio, mesmo com $\dot{V}(\mathbf{x}) \leq 0$, basta mostrarmos que o maior conjunto invariante dentro de $\dot{V}(\mathbf{x}) = 0$ é o ponto de equilíbrio (ou seja, o conjunto de pontos em que $\dot{V}(\mathbf{x}) = 0$ e os estados podem ficar parados é composto somente da origem).

Exemplo 14.2 (Pêndulo Simples - Função de Lyapunov dada pela energia mecânica (revisitado)). Voltando ao exemplo do pêndulo, vamos escolher $D = \{\mathbf{x} \in \mathbb{R}^2 | \|x_1\| < \pi\}$

como a nossa região de análise, e o maior conjunto de nível possível da função de Lyapunov contido em D como o nosso conjunto Ω .

O nosso conjunto E é dado pelo conjunto de pontos em que a derivada temporal da função de Lyapunov se iguala a zero. Nesse caso, $E = \{\mathbf{x} \in \Omega \mid \dot{x}_2 = 0\}$, note entretanto que para os pontos que formam um conjunto invariante dentro deste conjunto precisamos de $\dot{x}_2 = 0$, o que, no caso em que estamos analisando, só é verdade para a origem. Logo a origem é assintoticamente estável.

Nota 14.1

Vale ressaltar que o Princípio da Invariância só é válido quando o sistema que está sendo avaliado é autônomo. Quando este não é o caso, temos que recorrer a outras estratégias (como veremos mais adiante).

Sistemas não-autônomos

Quando consideramos sistemas não-autônomos, consideramos sistemas com a dinâmica descrita por

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x})$$

Assim como no caso de sistemas autônomos, consideramos que estamos analisando o comportamento de um ponto de equilíbrio na origem (caso este não seja o caso, fazemos uma transformação de coordenadas para que seja).

A diferença básica dos sistemas não autônomos para os sistemas autônomos é que, agora, o comportamento do sistema depende do tempo inicial também. De modo que, para os sistemas autônomos, sempre podemos considerar que o tempo inicial é zero (sem perda de generalidade), enquanto que não podemos fazer isso para os sistemas não autônomos.

Dessa forma, definimos os seguintes conceitos de estabilidade para sistemas não autônomos.

Estabilidade de um ponto de equilíbrio (considerando análise da origem)

Um ponto de equilíbrio é dito *estável* se, para todo $\varepsilon > 0$, existir um $\delta(\varepsilon, t_0) > 0$, tal que

$$\|\mathbf{x}(t_0)\| < \delta \Rightarrow \|\mathbf{x}(t)\| < \varepsilon, \forall t \geq t_0 > 0$$

Uniformemente estável se para todo $\varepsilon > 0$, existir um $\delta = \delta(\varepsilon) > 0$, independente de t_0 .

Instável quando ele não é estável.

Assintoticamente estável quando ele é estável e pode-se escolher $c(t_0)$ tal que

$$\|\mathbf{x}(t_0)\| < c(t_0) \Rightarrow \lim_{t \rightarrow \infty} \mathbf{x}(t) = 0$$

Uniformemente assintoticamente estável quando ele é uniformemente estável e a constante c independe do instante inicial t_0 e a convergência para o ponto de equilíbrio é uniforme, ou seja,

$$\|\mathbf{x}(t)\| < \eta, \forall t \geq t_0 + T(\eta), \forall \|\mathbf{x}(t_0)\| < c$$

Globalmente uniformemente assintoticamente estável quando ele é uniformemente assintoticamente estável e o domínio de atração do ponto de equilíbrio é todo o espaço de estados.

Teorema 14.2: Teorema de Estabilidade de Lyapunov (Método Direto) - Sistemas Não-Autônomos

Seja a origem um ponto de equilíbrio para o sistema, e $D \subset \mathbb{R}^n$ um domínio de análise contendo a origem. Seja $V : [0, \infty) \times D \rightarrow \mathbb{R}$ uma função continuamente diferenciável tal que

$$\begin{aligned} W_1(\mathbf{x}) &\leq V(t, \mathbf{x}) \leq W_2(\mathbf{x}) \\ \frac{\partial V}{\partial t} + \frac{\partial V}{\partial \mathbf{x}} \mathbf{f}(t, \mathbf{x}) &\leq 0 \end{aligned}$$

Com $W_1(\mathbf{x})$ e $W_2(\mathbf{x})$ funções contínuas e definidas positivas em D . Então a origem é uniformemente estável.

Se substituirmos a segunda desigualdade por

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial \mathbf{x}} \mathbf{f}(t, \mathbf{x}) \leq -W_3(\mathbf{x})$$

Com $W_3(\mathbf{x})$ outra função definida positiva em D , então a origem é uniformemente assintoticamente estável. Se $D = \mathbb{R}^n$ e $W_1(\mathbf{x})$ for radialmente ilimitada, então a origem é globalmente uniformemente assintoticamente estável.

Nota 14.2

Note que a diferença principal entre o teorema para sistemas autônomos e sistemas não-autônomos é que para sistemas não autônomos a função de Lyapunov pode depender do tempo. Entretanto, para provar estabilidade uniforme (que é a que nos interessa), ela tem que estar sempre contida entre outras duas funções definidas positivas (dependendo apenas dos estados), e sua derivada temporal tem de ser menor do que uma função definida negativa (dependendo apenas dos estados).

Exemplo 14.3. Considere o sistema linear variante no tempo descrito por

$$\begin{bmatrix} \dot{e} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -a & u(t) \\ -u(t) & 0 \end{bmatrix} \begin{bmatrix} e \\ \theta \end{bmatrix}$$

com $u(t)$ limitado. Escolhendo a função de Lyapunov

$$V(\mathbf{x}) = \frac{1}{2}(e^2 + \theta^2)$$

$$\dot{V}(\mathbf{x}) = e\dot{e} + \theta\dot{\theta} = -ae^2 + eu(t)\theta - eu(t)\theta = -ae^2 \leq 0$$

E, do Teorema de estabilidade para sistemas não autônomos, podemos dizer que a origem é uniformemente estável.

Como dissemos antes, o princípio da invariância não pode ser aplicado em sistemas não-autônomos. Quando este é o caso, normalmente, tentamos uma outra abordagem utilizando o *Lema de Barbalat*.

Lema 14.1: Lema de Barbalat - Primeira Forma

Seja $\phi(t) : \mathbb{R}^+ \rightarrow \mathbb{R}$ em \mathcal{L}_∞ , $\frac{d\phi}{dt} \in \mathcal{L}_\infty$, e $\frac{d\phi}{dt} \in \mathcal{L}_2$, então $\lim_{t \rightarrow \infty} \phi(t) = 0$.

(Considere um sinal no tempo. Se o sinal é limitado, a derivada do sinal é limitada e o quadrado da derivada do sinal é integrável - a integral dá um valor finito - então o sinal tende a zero quando o tempo tende a infinito).

Lema 14.2: Lema de Barbalat - Segunda Forma

Seja $\phi(t) : \mathbb{R}^+ \rightarrow \mathbb{R}$ uniformemente contínua em $[0, \infty)$. Se

$$\lim_{t \rightarrow \infty} \int_0^t \phi(s) ds = M < \infty$$

Então $\lim_{t \rightarrow \infty} \phi(t) = 0$.

(Algumas condições suficientes para que uma função seja uniformemente contínua são que ela tenha derivada temporal limitada, ou que a função seja Lipschitz).

Além disso, existe um outro lema que costuma ser útil.

Lema 14.3

Se $f(t) : \mathbb{R} \rightarrow \mathbb{R}$ for limitada abaixo e $\dot{f} \leq 0$, então $\lim_{t \rightarrow \infty} f(t) = f_\infty$ existe.

Exemplo 14.4 (Continuação do último exemplo). No último exemplo, provamos apenas a estabilidade uniforme do sistema. Considerando a função $\phi(t) = \dot{V}(\mathbf{x})$, a derivada de $\phi(t)$ é

$$\dot{\phi}(t) = -2ae(-ae + u(t)\theta) = 2a^2e^2 - 2aeu(t)\theta$$

Como $\dot{V}(\mathbf{x}) \leq 0$, temos

$$\frac{1}{2}e^2(t) \leq V(t) \leq V(0) \quad \frac{1}{2}\theta^2 \leq V(t) \leq V(0)$$

O que mostra que e e θ são limitados. Como consideramos que $u(t)$ é limitado, então $\dot{\phi}(t)$ também é limitado. Pelo lema apresentado logo antes desse exemplo, sabemos que $\lim_{t \rightarrow \infty} V(t)$ existe.

Portanto,

$$\lim_{t \rightarrow \infty} \int_0^t \phi(s) ds = \lim_{t \rightarrow \infty} V(t) - V(0)$$

existe e é limitado. Pela segunda forma do Lema de Barbalat,

$$\lim_{t \rightarrow \infty} \phi(t) = \lim_{t \rightarrow \infty} \dot{V}(t) \rightarrow 0$$

O que no caso desse exemplo implica que $e(t) \rightarrow 0$.

Aula 15

Sistemas Dinâmicos Não Lineares em Tempo Discreto

De forma semelhante ao que vimos para os sistemas em tempo contínuo, podemos utilizar funções de Lyapunov para estudar a estabilidade de sistemas em tempo discreto, fazendo algumas pequenas modificações nas definições. Informações mais detalhadas, e de forma mais formal, podem ser encontradas em (Khalil 2002; Mei e Bullo 2017).

Consideramos sistemas cujo comportamento é descrito por um sistema de equações de diferença de primeira ordem

$$\begin{aligned}x_{1_{k+1}} &= f_1(k, x_{1_k}, \dots, x_{n_k}, u_{1_k}, \dots, u_{m_k}) && \vdots \\x_{n_{k+1}} &= f_n(k, x_{1_k}, \dots, x_{n_k}, u_{1_k}, \dots, u_{m_k})\end{aligned}$$

Com k sendo o tempo, x_1 a x_n as variáveis de estado (variáveis que descrevem a memória do sistema) e u_1 a u_m as variáveis de entrada do sistema.

Normalmente, utilizamos uma representação vetorial para escrever as equações do sistema de forma mais compacta

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}, \quad \mathbf{f}(k, \mathbf{x}, \mathbf{u}) = \begin{bmatrix} f_1(k, \mathbf{x}, \mathbf{u}) \\ \vdots \\ f_n(k, \mathbf{x}, \mathbf{u}) \end{bmatrix}$$

$$\mathbf{x}_{k+1} = \mathbf{f}(k, \mathbf{x}_k, \mathbf{u}_k)$$

Em alguns casos, é interessante especificar quais são as saídas do sistema (em função dos estados e das entradas) e uma outra equação é utilizada

$$\mathbf{y}_k = \mathbf{h}(k, \mathbf{x}_k, \mathbf{u}_k)$$

(Normalmente chamamos o primeiro conjunto de equações de *equações de estado* e o segundo conjunto de *equações de saída*).

No caso em que temos um sistema linear (um sistema para o qual vale o *princípio da superposição*), os coeficientes do sistema de equações de diferença de primeira ordem são sempre independentes dos estados, e podemos representar as equações de estado e saída de forma vetorial como

$$\begin{aligned}\mathbf{x}_{k+1} &= A(k)\mathbf{x}_k + B(k)\mathbf{u}_k \\ \mathbf{y}_{k+1} &= C(k)\mathbf{x}_k + D(k)\mathbf{u}_k\end{aligned}$$

(E o comportamento do sistema pode ser completamente descrito pelas matrizes A , B , C e D).

Uma forma muito comum de se obter modelos lineares para o comportamento do sistema é por meio da *linearização* das equações não-lineares que descrevem o comportamento do sistema.

Em uma grande parte dessa unidade, vamos lidar com sistemas em que o sinal de entrada não aparece explicitamente na equação (pode ser que ele foi representado como um sinal do tempo ou então como uma função dos estados e substituído na equação dinâmica do sistema), e podemos escrever a equação do comportamento dinâmico do sistema como

$$\mathbf{x}_{k+1} = \mathbf{f}(k, \mathbf{x}_k)$$

De forma um pouco ainda mais específica, muitas vezes lidaremos com sistemas em que o tempo não tem nenhuma influência explícita sobre o comportamento do sistema. Nesse caso, dizemos que o sistema é *autônomo* ou *invariante no tempo* e podemos escrever seu comportamento como

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k)$$

(No caso em que o comportamento do sistema depende de forma explícita do tempo, dizemos que o sistema é *não-autônomo* ou *variante no tempo*).

Um conceito importante quando lidamos com a equação de estados é o conceito de *ponto de equilíbrio*. Um ponto \mathbf{x}^* é chamado de ponto de equilíbrio do sistema se quando o estado do sistema está naquele ponto, ele permanece naquele ponto para todos os estados futuros.

No caso de um sistema autônomo, os pontos de equilíbrio são as raízes de

$$\mathbf{f}(\mathbf{x}_k) = \mathbf{x}_k$$

porque isso implica em $\mathbf{x}_{k+1} = \mathbf{x}_k$.

Estabilidade no sentido de Lyapunov (estabilidade interna)

As definições de estabilidade para sistemas de tempo discreto são similares às de tempo contínuo, e por isso não vamos repetí-las aqui.

Os *Teoremas de Estabilidade de Lyapunov* são um pouco diferentes para sistemas discretos, e por isso são apresentados novamente.

Teorema 15.1: Teorema de Estabilidade de Lyapunov (Método Direto) - Sistemas Autônomos a tempo discreto

Seja a origem ($\mathbf{x} = \mathbf{0}$) um ponto de equilíbrio do sistema, e $D \subset \mathbb{R}^n$ uma região incluindo a origem. Se existir uma função $V(\mathbf{x}) : D \rightarrow \mathbb{R}$ tal que

$$V(\mathbf{0}) = 0, V(\mathbf{x}) > 0 \quad \forall \mathbf{x} \neq \mathbf{0} \text{ (função definida positiva)}$$

$$\Delta V = V(\mathbf{x}_{k+1}) - V(\mathbf{x}_k) \leq 0 \text{ (diferença semidefinida negativa)}$$

Então o ponto de equilíbrio é estável no sentido de Lyapunov. Se, ao invés disso,

$$\Delta V = V(\mathbf{x}_{k+1}) - V(\mathbf{x}_k) < 0 \text{ (diferença definida negativa)}$$

Então o ponto de equilíbrio é assintoticamente estável.

Uma função que atende as condições do teorema, é chamada de *função de Lyapunov*.

O teorema acima diz que se encontrarmos uma função semelhante à energia do sistema (definida positiva) cuja diferença temporal é sempre menor ou igual a zero, então a origem do sistema é estável. Se essa diferença for sempre menor do que zero, então o sistema é assintoticamente estável.

Quando temos que um ponto de equilíbrio é assintoticamente estável, é interessante caracterizarmos qual a *região de atração* (ou *domínio de atração* ou *bacia de atração*) daquele ponto (a região δ mais larga para a qual os estados convergem para o ponto de equilíbrio).

Do Teorema de estabilidade de Lyapunov, temos que uma estimativa do domínio de atração de um ponto de equilíbrio assintoticamente estável é dada pela maior curva/conjunto de nível da função de Lyapunov na região D . Ou seja, o maior c tal que $V(\mathbf{x}) < c, \forall \mathbf{x} \in D$.

Quando o domínio de atração de um ponto de equilíbrio é todo o espaço de estados, dizemos que aquele ponto de equilíbrio é *globalmente assintoticamente estável* (quando não dizemos o globalmente, fica implícito que a estabilidade é local).

Para verificar a estabilidade global de um ponto de equilíbrio, não basta tomar

$D = \mathbb{R}^n$ no teorema. Além disso, a função de Lyapunov deve ser *radialmente ilimitada*

$$\lim_{\|x\| \rightarrow \infty} V(x) \rightarrow \infty$$

Princípio da Invariância de LaSalle - Sistemas Autônomos

Assim como para sistemas a tempo contínuo, podemos fazer uso do Princípio da Invariância de LaSalle para sistemas que sejam autônomos.

Teorema 15.2: Princípio da Invariância de LaSalle

Seja $\Omega \subset D$ um conjunto compacto positivamente invariante com respeito à dinâmica do sistema. Seja $V : D \rightarrow \mathbb{R}$ uma função tal que $\Delta V(\mathbf{x}) \leq 0$ em Ω . Seja E o conjunto de todos os pontos em Ω em que $\Delta V(\mathbf{x}) = 0$. Seja M o maior conjunto invariante em E . Então toda solução (trajetória no espaço de estados) que se inicia em Ω se aproxima de M a medida que $k \rightarrow \infty$.

Intuitivamente, o princípio da invariância de LaSalle nos diz que os pontos em Ω convergem para o maior conjunto invariante em que $\Delta V(\mathbf{x}) = 0$ (porque se $\Delta V(\mathbf{x}) < 0$ então os pontos não ficam parados, e se um conjunto não é positivamente invariante, então mesmo com $\Delta V(\mathbf{x}) = 0$ as soluções deixam aquele conjunto).

Note que, diferentemente do Teorema de Estabilidade de Lyapunov, o princípio da invariância não impõe que a função seja definida positiva, e que o conjunto Ω não precisa estar amarrado à função $V(\mathbf{x})$.

Nos casos em que a função é definida positiva, pelo Teorema de Estabilidade de Lyapunov, sabemos que o ponto de equilíbrio é estável em D , e que o maior conjunto de nível da função de Lyapunov $\Omega_c = \{\mathbf{x} \in D \mid V(\mathbf{x}) \leq c\}$, com c o maior possível, é um conjunto compacto (um conjunto limitado e fechado) positivamente invariante (pois como $\dot{V}(\mathbf{x}) \leq 0$, os estados nunca crescem de valor e os conjuntos de nível que não saem da região D são garantidos de serem positivamente invariantes).

Dessa forma, para aplicarmos o princípio da invariância para provar a estabilidade assintótica de um ponto de equilíbrio, mesmo com $\Delta V(\mathbf{x}) \leq 0$, basta mostrarmos que o maior conjunto invariante dentro de $\Delta V(\mathbf{x}) = 0$ é o ponto de equilíbrio (ou seja, o conjunto de pontos em que $\Delta V(\mathbf{x}) = 0$ e os estados podem ficar parados é composto somente da origem).

Nota 15.1

Vale ressaltar que o Princípio da Invariância só é válido quando o sistema que está sendo avaliado é autônomo. Quando este não é o caso, temos que recorrer a outras estratégias (como veremos mais adiante).

Sistemas não-autônomos

Quando consideramos sistemas não-autônomos, consideramos sistemas com a dinâmica descrita por

$$\mathbf{x}_{k+1} = \mathbf{f}(k, \mathbf{x}_k)$$

Assim como no caso de sistemas autônomos, consideramos que estamos analisando o comportamento de um ponto de equilíbrio na origem (caso este não seja o caso, fazemos uma transformação de coordenadas para que seja).

As definições de estabilidade para sistemas não autônomos de tempo discreto são semelhantes as de tempo contínuo (*estabilidade uniforme*) e por isso não são apresentadas novamente.

Teorema 15.3: Teorema de Estabilidade de Lyapunov (Método Direto) - Sistemas Não-Autônomos

Seja a origem um ponto de equilíbrio para o sistema, e $D \subset \mathbb{R}^n$ um domínio de análise contendo a origem. Seja $V : [0, \infty) \times D \rightarrow \mathbb{R}$ uma função tal que

$$\begin{aligned} W_1(\mathbf{x}) &\leq V(k, \mathbf{x}) \leq W_2(\mathbf{x}) \\ V(k+1, \mathbf{x}_{k+1}) - V(k, \mathbf{x}_k) &\leq 0 \end{aligned}$$

Com $W_1(\mathbf{x})$ e $W_2(\mathbf{x})$ funções definidas positivas em D . Então a origem é uniformemente estável.

Se substituirmos a segunda desigualdade por

$$V(k+1, \mathbf{x}_{k+1}) - V(k, \mathbf{x}_k) \leq -W_3(\mathbf{x}_k)$$

Com $W_3(\mathbf{x}_k)$ outra função definida positiva em D , então a origem é uniformemente assintoticamente estável. Se $D = \mathbb{R}^n$ e $W_1(\mathbf{x})$ for radialmente ilimitada, então a origem é globalmente uniformemente assintoticamente estável.

Nota 15.2

Note que a diferença principal entre o teorema para sistemas autônomos e sistemas não-autônomos é que para sistemas não autônomos a função de Lyapunov pode depender do tempo. Entretanto, para provar estabilidade uniforme (que é a que nos interessa), ela tem que estar sempre contida entre outras duas funções definidas positivas (dependendo apenas dos estados), e sua diferença temporal tem de ser menor do que uma função definida negativa (dependendo apenas dos estados).

Assim como para sistemas a tempo contínuo, podemos utilizar algo semelhante ao *Lema de Barbalat*.

Lema 15.1: Lema de Barbalat - Tempo discreto

Seja $\phi_k : \mathbb{R}^+ \rightarrow \mathbb{R}$. Se

$$\lim_{k \rightarrow \infty} \sum_{i=0}^k \phi_i = M < \infty$$

Então $\lim_{k \rightarrow \infty} \phi_k = 0$.

Além disso, existe um outro lema que costuma ser útil.

Lema 15.2

Se $f_k : \mathbb{R} \rightarrow \mathbb{R}$ for limitada abaixo e $\Delta f_k \leq 0$, então $\lim_{k \rightarrow \infty} f_k = f_\infty$ existe.

Outras formas de se provar convergência para sistemas a tempo discreto

No caso de sistemas a tempo discreto, podemos fazer uso de outras ferramentas para provar a convergência dos estados para um ponto de equilíbrio. De forma a utilizarmos tais ferramentas, precisamos antes de algumas definições.

Definição 15.1: Ponto Fixo

Um ponto fixo de um mapeamento $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ é um ponto tal que

$$\mathbf{x} = \mathbf{f}(\mathbf{x})$$

(no nosso contexto de sistemas dinâmicos a tempo discreto, considerando a função f da dinâmica do sistema, os pontos fixos de f são os pontos de equilíbrio do sistema)

Definição 15.2: Contração (Contraction Mapping) - Espaço Euclidiano

Um mapeamento $T : X \rightarrow X$ é chamado de contração em X se existe $q \in [0, 1)$ tal que

$$\|T(\mathbf{x}) - T(\mathbf{y})\| \leq q\|\mathbf{x} - \mathbf{y}\|$$

para todo \mathbf{x}, \mathbf{y} em X .

Com essas duas definições podemos apresentar o *Teorema do Ponto Fixo de Banach*.

Teorema 15.4: Teorema do Ponto Fixo de Banach (Banach Fixed Point Theorem)

Seja T uma contração em X . Então T admite um único ponto fixo \mathbf{x}^* em X . Além disso, \mathbf{x}^* pode ser encontrado pela aplicação contínua de T a partir de um ponto inicial qualquer \mathbf{x}_0 em X .

Aula 16

Boundedness e Ultimate Boundedness

A análise via funções de Lyapunov de um sistema pode ser estendida para provar que a solução de um sistema é limitada, mesmo quando não existe um ponto de equilíbrio na origem.

Considere por exemplo o sistema:

$$\dot{x} = -x + \delta \operatorname{sen} t, \quad x(t_0) = a, \quad a > \delta > 0$$

que não possui nenhum ponto de equilíbrio, e cuja solução é dada por

$$x(t) = e^{-(t-t_0)}a + \delta \int_{t_0}^t e^{-(t-\tau)} \operatorname{sen} \tau d\tau$$

Note que a solução do sistema obedece ao limitante

$$|x(t)| \leq e^{-(t-t_0)}a + \delta \int_{t_0}^t e^{-(t-\tau)} d\tau = e^{-(t-t_0)}a + \delta (1 - e^{-(t-t_0)})$$
$$|x(t)| \leq a, \quad t \geq t_0$$

Que nos mostra que a solução do sistema é limitada para todo $t \geq t_0$, uniformemente em t_0 (independente de t_0). Quando conseguimos encontrar um limitante dessa forma, dizemos que as soluções do sistema são uniformemente limitadas (*uniformly bounded*).

Além disso, dá para ver que escolhendo um b tal que $\delta < b < a$, temos que

$$|x(t)| \leq b, \quad \forall t \geq t_0 + \ln \left(\frac{a - \delta}{b - \delta} \right)$$

E conseguimos um limitante melhor para a solução depois que algum tempo já se passou.

Quando conseguimos encontrar um limitante dessa forma, dizemos que as soluções do sistema são uniformemente finalmente limitadas (*uniformly ultimately bounded*) e chamamos b de limite final (*ultimate bound*).

Assim como no caso da análise de estabilidade, a análise com funções de Lyapunov nos permite verificar se as soluções de um sistema são limitadas sem precisar calcular explicitamente as soluções do mesmo.

Definição 16.1: Uniformly Bounded

Dizemos que as soluções de um sistema são uniformemente limitadas (*uniformly bounded*) se, existe um constante positiva c , independente de $t_0 \geq 0$, tal que para todo $a \in (0, c)$, existe um β , independente de $t_0 \geq 0$, tal que

$$\|\mathbf{x}(t_0)\| \leq a \Rightarrow \|\mathbf{x}(t)\| \leq \beta, \quad \forall t \geq t_0$$

Definição 16.2: Uniformly Ultimately Bounded - UUB

Dizemos que as soluções de um sistema são uniformemente finalmente limitadas (*uniformly ultimately bounded - UUB*) com limite final b , se existem constantes positivas b e c , independentes de $t_0 \geq 0$, tal que para todo $a \in (0, c)$, existe um tempo T , independente de $t_0 \geq 0$, tal que

$$\|\mathbf{x}(t_0)\| \leq a \Rightarrow \|\mathbf{x}(t)\| \leq b, \quad \forall t \geq t_0 + T$$

Dizemos que as soluções são globalmente limitadas ou globalmente finalmente limitadas quando os limites valem para a arbitrariamente grandes.

Além disso, no caso de sistemas autônomos, não utilizamos a palavra uniforme, pois o comportamento do sistema já é naturalmente independente de t_0 .

A forma de utilizar a análise via funções de Lyapunov para verificar se as soluções são limitadas é a seguinte.

Considere uma função continuamente diferenciável e definida positiva $V(\mathbf{x})$, e que o conjunto $\{\mathbf{x} \in \mathbb{R}^n | V(\mathbf{x}) < c\}$ seja fechado e limitado (compacto). Se

$$\dot{V}(\mathbf{x}) \leq -k, \quad \forall \mathbf{x} | \varepsilon < V(\mathbf{x}) < c$$

Sabemos que qualquer solução que inicie no conjunto $\{\mathbf{x} \in \mathbb{R}^n | V(\mathbf{x}) < c\}$ ficará presa no conjunto de nível $V(x) \leq \varepsilon$.

Além disso, integrando a equação acima no tempo, temos

$$V(\mathbf{x}(t)) \leq V(\mathbf{x}(t_0)) - k(t - t_0) \leq c - k(t - t_0)$$

que nos mostra que $V(\mathbf{x}(t))$ reduz para ε no intervalo $\left[t_0, t_0 + \frac{(c-\varepsilon)}{k} \right]$.

Em muitos dos casos, entretanto, chegamos a estas conclusões utilizando conjuntos definidos como limites na norma de \mathbf{x} ao invés de limites no valor da função de Lyapunov.

Nesse caso, normalmente precisamos de algumas considerações a mais em relação à função $V(\mathbf{x}(t))$.

Consideramos que ela seja limitada à esquerda e à direita por duas funções $\alpha_1(\|\mathbf{x}\|)$ e $\alpha_2(\|\mathbf{x}\|)$ estritamente crescentes de forma que

$$\alpha_1(\|\mathbf{x}\|) \leq V(\mathbf{x}) \leq \alpha_2(\|\mathbf{x}\|)$$

(note que sempre é possível encontrar tais funções para uma função definida positiva. No caso de uma função quadrática $V(\mathbf{x}) = \mathbf{x}^T P \mathbf{x}$, sempre podemos escrever

$$\lambda_{\min}(P) \mathbf{x}^T \mathbf{x} \leq V(\mathbf{x}) \leq \lambda_{\max}(P) \mathbf{x}^T \mathbf{x}$$

Considere, por exemplo, que temos que

$$\dot{V}(\mathbf{x}) \leq -k, \quad \forall \mathbf{x} | \mu < \|\mathbf{x}\| < r$$

Temos que dadas algumas condições iniciais, as soluções do sistema ficarão presas. Uma forma de se encontrar o limite sobre as soluções é vendo que

$$V(\mathbf{x}) \leq c \Rightarrow \alpha_1(\|\mathbf{x}\|) \leq c$$

E, por isso, que qualquer condição inicial em $V(\mathbf{x}) \leq \alpha_1(r)$ ficará presa nesse conjunto (o conjunto é positivamente invariante e podemos tomar $\alpha_1(r)$ como o c do outro caso acima).

Além disso, como

$$\begin{aligned} \|\mathbf{x}\| \leq \mu &\Rightarrow V(\mathbf{x}) \leq \alpha_2(\mu) \\ V(\mathbf{x}) \leq \varepsilon &\Rightarrow \alpha_1(\|\mathbf{x}\|) \leq \varepsilon \Leftrightarrow \|\mathbf{x}\| \leq \alpha_1^{-1}(\varepsilon) \end{aligned}$$

E sabemos que qualquer condição inicial em $V(\mathbf{x}) \leq \alpha_1(r)$ convergirá para o conjunto $\|\mathbf{x}\| \leq \alpha_1^{-1}(\alpha_2(\mu))$ e ficará preso nele dado tempo suficiente.

Reescrevendo a equação da derivada temporal

$$\dot{V}(\mathbf{x}) \leq -k, \quad \forall \mathbf{x} | \alpha_2(\mu) < V(\mathbf{x}) < \alpha_1(r)$$

E integrando no tempo

$$V(\mathbf{x}(t)) \leq V(\mathbf{x}(t_0)) - k(t - t_0) \leq \alpha_1(r) - k(t - t_0)$$

Que nos mostra que $V(\mathbf{x}(t))$ reduz para $\alpha_2(\mu)$ no intervalo $\left[t_0, t_0 + \frac{(\alpha_1(r) - \alpha_2(\mu))}{k} \right]$.

E sabemos que

$$\|\mathbf{x}\| \leq \alpha_1^{-1}(\alpha_2(\mu)), \quad \forall t \geq t_0 + \frac{(\alpha_1(r) - \alpha_2(\mu))}{k}.$$

Aula 17

Feedback Linearization

Tendo visto os conceitos de estabilidade para sistemas não-lineares, podemos estudar agora algumas técnicas de controle para estes sistemas. Uma das técnicas mais usuais e intuitivas é a de linearização por realimentação (feedback linearization).

Consideramos sistemas da forma (sistemas afins nas entradas de controle)

$$\begin{aligned}\dot{x} &= f(x) + G(x)u \\ y &= h(x)\end{aligned}$$

E tentamos encontrar uma lei de controle por realimentação de estados

$$u = \alpha(x) + \beta(x)v$$

E uma mudança de variáveis

$$z = T(x)$$

que quando combinadas transformam a dinâmica do sistema em uma dinâmica linear.

Motivação

Considere o problema de estabilização de um pêndulo na origem

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -a(\sin(x_1 + \delta) - \sin \delta) - bx_2 + cu\end{aligned}$$

Olhando para as equações de estado, vemos que se escolhermos u como

$$u = \frac{a}{c}(\sin(x_1 + \delta) - \sin \delta) + \frac{v}{c}$$

o termo não linear $a(\sin(x_1 + \delta) - \sin \delta)$ é cancelado na dinâmica. Este cancelamento resulta no sistema linear

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -bx_2 + v\end{aligned}$$

Podemos então escolher a lei de controle linear

$$v = -k_1x_1 - k_2x_2$$

para estabilizar o sistema, colocando os polos aonde quisermos no semiplano esquerdo do plano s . (É fácil ver que, nesse caso, a equação característica do sistema será dada por $s^2 + k_1s + (k_2 + b)$.)

Juntando v na lei de controle, vemos que a lei de controle completa é então dada por

$$u = \frac{a}{c}(\sin(x_1 + \delta) - \sin \delta) - \frac{1}{c}(k_1x_1 + k_2x_2)$$

Generalizando, vemos que, para um sistema não-linear da forma

$$\dot{x} = Ax + B\gamma(x)[u - \alpha(x)]$$

A lei de controle

$$u = \alpha(x) + \beta(x)v, \quad \beta(x) = \frac{1}{\gamma(x)}$$

Transforma a dinâmica do sistema em

$$\dot{x} = Ax + Bv$$

para todo caso em que $\gamma(x) \neq 0$.

Mas, e quando não possuímos o sistema na forma desejada, será que ainda é possível encontrarmos uma lei de controle que linearize a dinâmica do sistema?

Considere o sistema

$$\begin{aligned}\dot{x}_1 &= a \sin x_2 \\ \dot{x}_2 &= -x_1^2 + u\end{aligned}$$

Note que não é possível escolhermos um u diretamente que cancele a não linearidade na dinâmica de x_1 . Entretanto, se fizermos a mudança de variáveis

$$\begin{aligned}z_1 &= x_1 \\ z_2 &= \dot{x}_1 = a \sin x_2\end{aligned}$$

Então a dinâmica do sistema passa a ser descrita por

$$\begin{aligned}\dot{z}_1 &= z_2 \\ \dot{z}_2 &= a \cos x_2 \left(-x_1^2 + u \right)\end{aligned}$$

E podemos cancelar as não-linearidades de dinâmica por

$$u = x_1^2 + \frac{1}{a \cos x_2} v$$

(Note que esta lei de controle só é válida para $\pi/2 < x_2 < \pi/2$.)

É importante notar que, quando fazemos uma transformação de coordenadas, esta deve ser invertível e diferenciável (damos o nome de *difeomorfismo* para uma transformação com estas características). Pelo *teorema da função inversa*, sabe-se que se a matriz jacobiana $\frac{\partial T}{\partial x}$, de uma transformação $T(x)$, for não singular em um ponto, então existe uma transformação inversa numa região em torno do ponto.

Definição 17.1: Sistema linearizável por realimentação

Dizemos que um sistema

$$\dot{x} = f(x) + G(x)u$$

é linearizável por realimentação (*feedback linearizable*) se existe um difeomorfismo $z = T(x)$ que transforma o sistema para a forma

$$\dot{z} = Az + B\gamma(x) [u - \alpha(x)]$$

com (A, B) controlável e $\gamma(x)$ não-singular dentro da região desejada.

Linearização Entrada-Saída (sistema SISO - Uma entrada/ Uma saída)

Até este momento, vimos o caso em que o controlador lineariza a dinâmica de todo o sistema, entretanto, em alguns casos podemos estar interessados a obter um comportamento linear considerando apenas o comportamento entrada/saída do sistema.

Considere o sistema SISO

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= h(x)\end{aligned}$$

A derivada temporal de y pode ser escrita como

$$\dot{y} = \frac{\partial h}{\partial x} f(x) + \frac{\partial h}{\partial x} g(x)u = L_f h(x) + L_g h(x)u$$

Em que

$$L_f h(x) = \frac{\partial h}{\partial x} f(x), \quad L_g h(x) = \frac{\partial h}{\partial x} g(x)$$

são derivadas direcionais que recebem o nome de *derivadas de Lie*.

Utilizando a notação de derivadas de Lie, podemos definir

$$\begin{aligned} L_g L_f h(x) &= \frac{\partial(L_f h)}{\partial x} g(x) \\ L_f^2 h(x) &= L_f L_f h(x) = \frac{\partial(L_f h)}{\partial x} f(x) \\ L_f^k h(x) &= L_f L_f^{k-1} h(x) = \frac{\partial(L_f^{k-1} h)}{\partial x} f(x) \end{aligned}$$

A ideia da linearização entrada-saída, é que começamos com a equação de saída e derivamos ela no tempo até que a entrada apareça na equação.

Utilizando a notação de derivadas de Lie apresentada acima, para a derivada primeira temos

$$\dot{y} = L_f h(x) + L_g h(x)u$$

se $L_g h(x) = 0$, então a equação não depende de u , e continuamos derivando.

Para a derivada segunda (considerando $L_g h(x) = 0$)

$$\ddot{y} = L_f^2 h(x) + L_g L_f h(x)u$$

Se $L_g L_f h(x) = 0$, então a equação não depende de u , e continuamos derivando até que a equação dependa de u .

Tomando ρ derivadas temporais de y , temos

$$y^{(\rho)} = L_f^\rho h(x) + L_g L_f^{\rho-1} h(x)u$$

Com

$$\begin{aligned} L_g L_f^{k-1} h(x) &= 0, \quad i = 1, 2, \dots, \rho - 1 \\ L_g L_f^{\rho-1} h(x) &\neq 0 \end{aligned}$$

E a dinâmica de entrada/saída pode ser linearizada tomando-se o controlador

$$u = \frac{1}{L_g L_f^{\rho-1} h(x)} \left(-L_f^\rho h(x) + v \right)$$

Levando a dinâmica do comportamento entrada/saída para

$$y^{(\rho)} = v$$

(que é claramente uma dinâmica linear - cadeia de integradores)

O número de vezes que foi necessário se derivar a equação de saída, ρ , é chamado de grau relativo do sistema.

Note, entretanto, que ao fazermos este procedimento estamos apenas linearizando (e controlando) o comportamento entrada/saída do sistema. Se o grau relativo do

sistema for igual à ordem (número de estados), então encontramos um difeomorfismo que nos permite fazer a linearização por realimentação (entrada/estado).

É possível mostrar que se um sistema é linearizável por realimentação, então sempre é possível se escolher uma saída do sistema tal que o grau relativo seja igual ao número de estados (e se encontra uma realimentação linearizante pelo procedimento de linearização entrada/saída).

Quando este não é o caso, podemos definir uma transformação de variáveis da seguinte forma

$$z = T(x) = \begin{bmatrix} \phi_1(x) \\ \vdots \\ \phi_{n-\rho}(x) \\ \text{---} \\ h(x) \\ \vdots \\ L_f^{\rho-1}h(x) \end{bmatrix} = \begin{bmatrix} \phi(x) \\ \text{---} \\ y \\ \vdots \\ y^{\rho-1} \end{bmatrix} = \begin{bmatrix} \eta \\ \text{---} \\ \xi \end{bmatrix}$$

Com as funções ϕ_1 até $\phi_{n-\rho}$ escolhidas de modo que $T(x)$ seja um difeomorfismo e

$$\frac{\partial \phi_i}{\partial x} g(x) = 0$$

(esta condição garante que a dinâmica da parte escondida não depende das entradas)

Com esta transformação, a dinâmica em ξ está associada ao comportamento entrada/saída do sistema, e a dinâmica η está escondida do comportamento entrada/saída do sistema.

A dinâmica do sistema transformado pode ser escrita na forma

$$\begin{aligned} \dot{\eta} &= f_0(\eta, \xi) \\ \dot{\xi} &= A_c \xi + B_c \gamma(x) [u - \alpha(x)] \\ y &= C_c \xi \end{aligned}$$

Um sistema escrito nessa forma é dito estar na *forma normal*. Ela decompõe a dinâmica do sistema em uma parte externa ξ e uma parte interna η e nos permite estudar o comportamento do sistema quando fazemos uso da linearização entrada/saída por realimentação.

A dinâmica

$$\dot{\eta} = f_0(\eta, 0)$$

é chamada de dinâmica zero, e se estável, garante que uma lei de controle por linearização entrada/saída é estável em malha fechada (um sistema com dinâmica zero estável é dito de fase mínima).

Exemplo 17.1 (exemplo 5.6 Farrel e Polycarpou 2006). Considere o sistema

$$\begin{aligned}\dot{x}_1 &= x_2 - ax_1^3 \\ \dot{x}_2 &= 2x_3^2 + u \\ \dot{x}_3 &= x_1 + x_2^3 - 3x_3^3 \\ y &= x_1\end{aligned}$$

Em que a é uma constante. Tomando as duas primeiras derivadas temporais de y , temos

$$\begin{aligned}\dot{y} &= x_2 - ax_1^3 \\ \ddot{y} &= 2x_3^2 - ax_1^2 + 3a^2x_1^5 + u\end{aligned}$$

e portanto o grau relativo do sistema é 2. Usando a transformação

$$\begin{aligned}\eta &= x_3 \\ \zeta_1 &= x_1 \\ \zeta_2 &= x_2 - ax_1^3\end{aligned}$$

Podemos converter o sistema para a forma normal

$$\begin{aligned}\dot{\eta} &= \zeta_1 + (\zeta_2 + a\zeta_1^3)^3 - 3\eta^3 \\ \dot{\zeta}_1 &= \zeta_2 \\ \dot{\zeta}_2 &= -3a\zeta_1^2\zeta_2 + 2\eta^2 + u\end{aligned}$$

Usando a lei de controle

$$u = 3a\zeta_1^2\zeta_2 - 2\eta^2 + v$$

Chegamos à dinâmica

$$\begin{aligned}\dot{\eta} &= \zeta_1 + (\zeta_2 + a\zeta_1^3)^3 - 3\eta^3 \\ \dot{\zeta}_1 &= \zeta_2 \\ \dot{\zeta}_2 &= v\end{aligned}$$

E a dinâmica zero é dada por

$$\dot{\eta} = -3\eta^3$$

(Que é claramente assintoticamente estável. Logo o sistema é de fase mínima, e a lei de controle linearizante pode ser utilizada para estabilizar o sistema)

Aula 18

Backstepping

Outra técnica de controle de sistemas não-lineares muito utilizada com controladores nebulosos adaptativos é a técnica chamada Backstepping. Como veremos na aula de hoje, ela é uma técnica sistemática que permite o desenvolvimento de uma cascata de controladores para sistemas com uma estrutura específica.

O desenvolvimento aqui apresentado é baseado em (Farrel e Polycarpou 2006, seção 5.3).

Apesar de ser uma técnica sistemática, como veremos na aula de hoje, a principal desvantagem é que a lei de controle final aplicada ao sistema pode ser bastante complicada no caso de sistemas de ordem elevada.

Sistemas de Segunda Ordem

Considere um sistema de segunda ordem da seguinte forma

$$\begin{aligned}\dot{x}_1 &= f_1(x_1) + g_1(x_1)x_2 \\ \dot{x}_2 &= f_2(\mathbf{x}) + g_2(\mathbf{x})u\end{aligned}$$

com uma entrada de controle, u , e uma saída de controle, $y = x_1$, e o vetor de estados, $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$.

Nosso objetivo de controle é projetar uma lei de controle para u que leve o estado x_1 a seguir uma referência desejada y_d . De modo a simplificar a síntese da lei de controle desejada para o sistema, supomos que as funções $g_1(x_1)$ e $g_2(\mathbf{x})$ são tais que $g_1(x_1) \neq 0$ e $g_2(\mathbf{x}) \neq 0$ dentro da região de operação do controlador.

A ideia principal do *backstepping* é a seguinte, se pudessemos arbitrar o valor de x_2 como se fosse uma ação de controle, poderíamos fazer com que x_1 tivesse a dinâmica que desejássemos. Portanto, criamos uma ação de controle virtual (que chamaremos de α) que servirá de referência para o segundo estado. Dessa forma, projetamos então uma ação de controle u tal que o segundo estado siga o comportamento da ação de controle virtual.

Definindo uma mudança de variáveis para variáveis de erro em relação ao desejado, podemos fazer

$$z_1 = x_1 - y_d$$

$$z_2 = x_2 - \alpha$$

e podemos escrever a dinâmica do erro como

$$\dot{z}_1 = \dot{x}_1 - \dot{y}_d = f_1(x_1) + g_1(x_1)x_2 - \dot{y}_d$$

$$\dot{z}_2 = \dot{x}_2 - \dot{\alpha} = f_2(\mathbf{x}) + g_2(\mathbf{x})u - \dot{\alpha}$$

Mas, note que podemos escrever a dinâmica do erro como

$$\dot{z}_1 = f_1(x_1) + g_1(x_1)\alpha - g_1(x_1)\alpha + g_1(x_1)x_2 - \dot{y}_d$$

$$\dot{z}_2 = f_2(\mathbf{x}) + g_2(\mathbf{x})u - \dot{\alpha}$$

o que leva a

$$\dot{z}_1 = f_1(x_1) + g_1(x_1)\alpha + g_1(x_1)z_2 - \dot{y}_d$$

$$\dot{z}_2 = f_2(\mathbf{x}) + g_2(\mathbf{x})u - \dot{\alpha}$$

Definindo

$$\alpha(x_1, y_d, \dot{y}_d) = \frac{-f_1(x_1) - k_1(x_1 - y_d) + \dot{y}_d}{g_1(x_1)}$$

temos que

$$\dot{\alpha} = \frac{\partial \alpha}{\partial x_1} \dot{x}_1 + \frac{\partial \alpha}{\partial y_d} \dot{y}_d + \frac{\partial \alpha}{\partial \dot{y}_d} \ddot{y}_d$$

e a dinâmica do erro dada por

$$\dot{z}_1 = -k_1 z_1 + g_1(x_1)z_2$$

$$\dot{z}_2 = f_2(\mathbf{x}) + g_2(\mathbf{x})u - \dot{\alpha}$$

Escolhendo uma função de Lyapunov quadrática da forma

$$V(z_1, z_2) = \frac{1}{2}z_1^2 + \frac{1}{2}z_2^2$$

temos que sua derivada temporal será dada por

$$\begin{aligned}\dot{V} &= z_1 (-k_1 z_1 + g_1(x_1)z_2) + z_2 (f_2(\mathbf{x}) + g_2(\mathbf{x})u - \dot{\alpha}) \\ &= -k_1 z_1^2 + z_2 (g_1(x_1)z_1 + f_2(\mathbf{x}) + g_2(\mathbf{x})u - \dot{\alpha})\end{aligned}$$

Logo, se escolhermos a ação de controle como

$$u = \frac{-f_2(\mathbf{x}) - g_1(x_1)z_1 - k_2 z_2 + \dot{\alpha}}{g_2(\mathbf{x})} \quad (18.1)$$

teremos que a derivada temporal da função de Lyapunov será dada por

$$\dot{V} = -k_1 z_1^2 - k_2 z_2^2$$

que é definida negativa e portanto garante a estabilidade assintótica da dinâmica do erro (que será global, caso as considerações feitas sejam globais).

Note que a ação de controle encontrada para u em (18.1) é sintonizada pela função de Lyapunov utilizada e não somente de forma a fazer com que a dinâmica de z_2 seja assintoticamente estável (quando analisada separadamente). Entretanto, se desconsiderarmos o termo em **vermelho**, ela seria justamente isso. O termo destacado é responsável por cancelar o termo que restou da "primeira parte" da função de Lyapunov.

Sistemas de Ordem Mais Alta

Para sistemas de ordem mais alta, consideramos que eles podem ser escritos numa forma de realimentação estrita (*strict-feedback form*)

$$\begin{aligned}\dot{x}_1 &= f_1(x_1) + g_1(x_1)x_2 \\ \dot{x}_2 &= f_2(x_1, x_2) + g_2(x_1, x_2)x_3 \\ \dot{x}_3 &= f_3(x_1, x_2, x_3) + g_3(x_1, x_2, x_3)x_4 \\ &\vdots \\ \dot{x}_n &= f_n(\mathbf{x}) + g_n(\mathbf{x})u\end{aligned}$$

Seguimos a mesma ideia que utilizamos para o sistema de segunda ordem, repetindo a ideia cada vez de forma recursiva.

Projetamos uma ação de controle virtual, α_1 , de modo a controlar x_1 para seguir a referência y_d . Depois projetamos uma ação de controle virtual, α_2 , de modo a controlar x_2 para seguir ação de controle virtual α_1 . E repetimos o processo até que possamos projetar u de forma que x_n siga a ação de controle virtual α_{n-1} .

Exemplo 18.1 (Exemplo 5.7 Farrel e Polycarpou 2006). Considere um sistema de terceira ordem

$$\begin{aligned}\dot{x}_1 &= x_1^2 + (1 + x_1^2)x_2 \\ \dot{x}_2 &= x_1x_2 + (2 + \cos x_2)x_3 \\ \dot{x}_3 &= x_3^2 + (1 + x_1^2x_2^2)u\end{aligned}$$

podemos fazer a mudança de variáveis

$$\begin{aligned}z_1 &= x_1 - y_d \\ z_2 &= x_2 - \alpha_1 \\ z_3 &= x_3 - \alpha_2\end{aligned}$$

e podemos escrever a dinâmica das nossas variáveis de erro como

$$\begin{aligned}\dot{z}_1 &= x_1^2 + (1 + x_1^2)x_2 - \dot{y}_d + (1 + x_1^2)\alpha_1 - (1 + x_1^2)\alpha_1 \\ \dot{z}_2 &= x_1x_2 + (2 + \cos x_2)x_3 - \dot{\alpha}_1 + (2 + \cos x_2)\alpha_2 - (2 + \cos x_2)\alpha_2 \\ \dot{z}_3 &= x_3^2 + (1 + x_1^2x_2^2)u - \dot{\alpha}_2\end{aligned}$$

que podemos reescrever como

$$\begin{aligned}\dot{z}_1 &= x_1^2 - \dot{y}_d + (1 + x_1^2)\alpha_1 + (1 + x_1^2)z_2 \\ \dot{z}_2 &= x_1x_2 - \dot{\alpha}_1 + (2 + \cos x_2)\alpha_2 + (2 + \cos x_2)z_3 \\ \dot{z}_3 &= x_3^2 + (1 + x_1^2x_2^2)u - \dot{\alpha}_2\end{aligned}$$

Escolhendo uma função de Lyapunov para o sistema como

$$\begin{aligned}V(z_1, z_2, z_3) &= V_1(z_1) + V_2(z_2) + V_3(z_3) \\ V_1(z_1) &= \frac{1}{2}z_1^2 \\ V_2(z_2) &= \frac{1}{2}z_2^2 \\ V_3(z_3) &= \frac{1}{2}z_3^2\end{aligned}$$

temos

$$\dot{V}_1 = z_1 \left(x_1^2 - \dot{y}_d + (1 + x_1^2)\alpha_1 + (1 + x_1^2)z_2 \right)$$

Escolhendo

$$\alpha_1 = \frac{\dot{y}_d - x_1^2 - k_1z_1}{1 + x_1^2}$$

passamos a ter

$$\dot{V}_1 = -k_1z_1^2 + z_1z_2(1 + x_1^2)$$

Tomando a derivada temporal dos dois primeiros termos da função de Lyapunov, temos

$$\begin{aligned}\dot{V}_1 + \dot{V}_2 &= -k_1 z_1^2 + z_1 z_2 (1 + x_1^2) + z_2 (x_1 x_2 - \dot{\alpha}_1 + (2 + \cos x_2) \alpha_2 + (2 + \cos x_2) z_3) \\ &= -k_1 z_1^2 + z_2 \left(z_1 (1 + x_1^2) + x_1 x_2 - \dot{\alpha}_1 + (2 + \cos x_2) \alpha_2 \right) + z_2 z_3 (2 + \cos x_2)\end{aligned}$$

Escolhendo

$$\alpha_2 = \frac{\dot{\alpha}_1 - z_1 (1 + x_1^2) - x_1 x_2 - k_2 z_2}{2 + \cos x_2} \quad (18.2)$$

passamos a ter

$$\dot{V}_1 + \dot{V}_2 = -k_1 z_1^2 - k_2 z_2^2 + z_2 z_3 (2 + \cos x_2)$$

Tomando a derivada temporal da função de Lyapunov do sistema como um todo, temos

$$\begin{aligned}\dot{V} = \dot{V}_1 + \dot{V}_2 + \dot{V}_3 &= -k_1 z_1^2 - k_2 z_2^2 + z_2 z_3 (2 + \cos x_2) + z_3 \left(x_3^2 + (1 + x_1^2 x_2^2) u - \dot{\alpha}_2 \right) \\ &= -k_1 z_1^2 - k_2 z_2^2 + z_3 \left(z_2 (2 + \cos x_2) + x_3^2 + (1 + x_1^2 x_2^2) u - \dot{\alpha}_2 \right)\end{aligned}$$

Escolhendo

$$u = \frac{\dot{\alpha}_2 - z_2 (2 + \cos x_2) - x_3^2 - k_3 z_3}{1 + x_1^2 x_2^2}$$

passamos a ter

$$\dot{V} = -k_1 z_1^2 - k_2 z_2^2 - k_3 z_3^2$$

e a dinâmica do erro será globalmente assintoticamente estável, para $k_1 > 0$, $k_2 > 0$, $k_3 > 0$.

O cálculo da ação de controle envolve o cálculo de $\dot{\alpha}_1$ e $\dot{\alpha}_2$ que pode ser algebricamente tedioso.

Para demonstrar isto, temos que, no caso deste exemplo, a derivada temporal de α_1 pode ser calculada como

$$\dot{\alpha}_1 = \frac{\dot{y}_d - (2x_1 + k_1) (x_1^2 + (1 + x_1^2) x_2) + k_1 y_d}{1 + x_1^2} - 2 \left(\frac{\dot{y}_d - x_1^2 - k_1 z_1}{(1 + x_1^2)^2} \right) \left(x_1^3 + (x_1 + x_1^3) x_2 \right) \quad (18.3)$$

Note que o cálculo da derivada temporal de α_2 envolve a substituição de (18.3) em (18.2) para depois calcularmos a derivada temporal.

Formulação usando filtro de comando

Como podemos ver com o exemplo anterior, grande parte da complexidade no cálculo de uma lei de controle *backstepping* vem do cálculo das derivadas temporais das ações de controle virtuais definidas.

Dessa forma, consideramos aqui uma formulação que evita o uso destas derivadas temporais por meio de filtros de comando.

Considere um sistema de segunda ordem, descrito por

$$\begin{aligned}\dot{x}_1 &= f_1(x_1) + g_1(x_1)x_2 \\ \dot{x}_2 &= f_2(\mathbf{x}) + g_2(\mathbf{x})u\end{aligned}$$

com as mesmas considerações que anteriormente.

Definimos os erros de rastreamento como

$$\begin{aligned}z_1 &= x_1 - y_d \\ z_2 &= x_2 - x_{2d}\end{aligned}$$

em que x_{2d} será definido pelo controlador *backstepping*. Como vimos anteriormente, no *backstepping* tradicional, escolheríamos a ação de controle virtual como

$$\alpha_1(x_1, z_1, \dot{y}_d) = \frac{-f_1 - k_1 z_1 + \dot{y}_d}{g_1}$$

Entretanto, ao invés de prosseguirmos como de costume, definimos uma série de filtros de comando e de erro no controlador para evitar o cálculo das derivadas temporais das ações de controle virtuais.

Ao invés de definirmos x_{2d} como α_1 , como de costume, definimos

$$x_{2d}^o = \alpha_1(x_1, z_1, \dot{y}_d) - \xi_2$$

em que x_{2d}^o representa o sinal que gostaríamos que x_2 rastreasse originalmente e ξ_2 o erro filtrado entre a entrada de controle (real ou virtual) desejada e a utilizada para controlar o próximo estado.

Ao invés de utilizar x_{2d}^o como a referência para o nosso próximo estado, utilizamos um filtro de comando, que tenha como saídas x_{2d} e \dot{x}_{2d} . Um exemplo de filtro que poderia ser utilizado é

$$\begin{aligned}\dot{x}_{2d} &= -ax_{2d} + ax_{2d}^o \\ \begin{bmatrix} \dot{x}_{2d} \\ x_{2d} \end{bmatrix} &= \begin{bmatrix} -a \\ 1 \end{bmatrix} x_{2d} + \begin{bmatrix} a \\ 0 \end{bmatrix} x_{2d}^o\end{aligned}$$

em que utilizamos um filtro passa baixas de primeira ordem (com a sendo um parâmetro utilizado para determinar a largura de banda do filtro) modificado para que as saídas sejam a entrada filtrada e a derivada temporal da entrada filtrada.

Com o filtro de comando, e a nossa referência desejada para o segundo estado, definimos também o filtro de erro entre o estado desejado e o estado desejado filtrado

$$\dot{\zeta}_1 = -k_1\zeta_1 + g_1(x_1)(x_{2d} - x_{2d}^o)$$

A partir dos erros filtrados, podemos definir os erros de trajetória compensados como

$$s_i = z_i - \zeta_i, \quad i = 1 \text{ ou } 2$$

E por último, podemos definir

$$u^o = \frac{-k_2 z_2 + \dot{x}_{2d} - f_2 - s_1 g_1}{g_2}$$

$$\dot{\zeta}_2 = -k_2\zeta_2 + g_2(\mathbf{x})(u - u^o)$$

em que u^o é filtrado para gerar a ação de controle u aplicada ao sistema. Como está é a entrada de controle real aplicada ao sistema, tal filtro não precisa ser utilizado, e neste caso teremos que $u = u^o$ e $\zeta_2 = 0$.

Tendo definido o controlador como um todo, podemos analisar o comportamento do sistema em malha fechada. A dinâmica do erro de rastreamento pode ser escrita como

$$\begin{aligned} \dot{z}_1 &= f_1 + g_1 x_{2d}^o - \dot{y}_d + g_1(x_{2d} - x_{2d}^o) + (g_1 x_2 - g_1 x_{2d}) \\ &= f_1 + g_1 \alpha_1 - \dot{y}_d - g_1 \zeta_2 + g_1(x_{2d} - x_{2d}^o) + g_1(x_2 - x_{2d}) \\ &= -k_1 z_1 - g_1 \zeta_2 + g_1(x_{2d} - x_{2d}^o) + g_1(x_2 - x_{2d}) \\ &= -k_1 z_1 + g_1 s_2 + g_1(x_{2d} - x_{2d}^o) \\ \dot{z}_2 &= f_2 + g_2 u^o - \dot{x}_{2d} + g_2(u - u^o) \\ &= -k_2 z_2 - g_1 s_1 + g_2(u - u^o) \end{aligned}$$

A dinâmica dos erros de rastreamento compensados pode ser escrita como

$$\begin{aligned} \dot{s}_1 &= \dot{z}_1 - \dot{\zeta}_1 \\ &= -k_1 z_1 + g_1 s_2 + g_1(x_{2d} - x_{2d}^o) + k_1 \zeta_1 - g_1(x_1)(x_{2d} - x_{2d}^o) \\ &= -k_1 s_1 + g_1 s_2 \\ \dot{s}_2 &= \dot{z}_2 - \dot{\zeta}_2 \\ &= -k_2 z_2 - g_1 s_1 + g_2(u - u^o) + k_2 \zeta_2 - g_2(\mathbf{x})(u - u^o) \\ &= -k_2 s_2 - g_1 s_1 \end{aligned}$$

Considerando a função de Lyapunov

$$V = \sum_{i=1}^2 V_i(s_i), \quad \text{com } V_i(s_i) = \frac{1}{2} s_i^2$$

temos que as derivadas temporais serão dadas por

$$\begin{aligned}\dot{V}_1 &= -k_1 s_1^2 + s_1 g_1 s_2 \\ \dot{V}_2 &= -k_2 s_2^2 - s_2 g_1 s_1 \\ \dot{V} &= \dot{V}_1 + \dot{V}_2 = -k_1 s_1^2 - k_2 s_2^2 \leq -\lambda V\end{aligned}$$

em que $\lambda = 2 \min(k_1, k_2)$, e o fato de que $\dot{V} \leq -\lambda V$ nos garante que a origem é assintoticamente estável.

Note que a função de Lyapunov utilizada garante que $\mathbf{s} \rightarrow 0$ exponencialmente rápido, e não que $\mathbf{z} \rightarrow 0$. A diferença entre esses dois é dada pelos erros compensados

$$\dot{\xi}_i = -k_i \xi_i + g_i(x_{(i+1)d} - x_{(i+1)d}^o)$$

e é determinada pelo projeto do filtro de comando (contanto que o comando desejado esteja razoavelmente dentro da banda de passagem do filtro de comando, tal diferença deve se aproximar de zero).

Aula 19

Lista de Exercícios - Sistemas Não Lineares

Exercício 19.1: Khalil 2002 - Exercício 4.3

Para cada um dos sistemas abaixo, use uma função de Lyapunov quadrática para mostrar que a origem é assintoticamente estável

1.

$$\dot{x}_1 = -x_1 + x_1x_2$$

$$\dot{x}_2 = -x_2$$

2.

$$\dot{x}_1 = -x_2 - x_1(1 - x_1^2 - x_2^2)$$

$$\dot{x}_2 = x_1 - x_2(1 - x_1^2 - x_2^2)$$

3.

$$\dot{x}_1 = x_2(1 - x_1^2)$$

$$\dot{x}_2 = -(x_1 + x_2)(1 - x_1^2)$$

4.

$$\dot{x}_1 = -x_1 - x_2$$

$$\dot{x}_2 = 2x_1 - x_2^3$$

Exercício 19.2: Khalil 2002 - Exercício 4.27

Considere o sistema

$$\dot{x}_1 = -x_2x_3 + 1$$

$$\dot{x}_2 = x_1x_3 - x_2$$

$$\dot{x}_3 = x_3^2(1 - x_3)$$

1. Mostre que o sistema só tem um único ponto de equilíbrio
2. Usando linearização, mostre que o ponto de equilíbrio é assintoticamente estável.

Exercício 19.3: Modificado de Khalil 2002 - Exercício 4.19

Considere o sistema

$$m(x)\ddot{x} + c(x, \dot{x})\dot{x} + g(x) = u$$

Com a lei de controle

$$u = g(x) - k_p x - k_d \dot{x}$$

Com $k_p > 0$, e $k_d > 0$.

Use a função de Lyapunov

$$V(x, \dot{x}) = \frac{1}{2}m(x)\dot{x}^2 + \frac{1}{2}k_p x^2$$

e o fato de que

$$\dot{m}(x) - 2c(x, \dot{x}) = 0$$

para mostrar que o sistema é estável em malha fechada. Use o Princípio da Invariância de LaSalle para mostrar que o sistema é assintoticamente estável em malha fechada.

Exercício 19.4: Questão de feedback linearizarion - Pêndulo Invertido

Usando uma lei de controle de linearização por realimentação (feedback linearization), projete um controlador que estabilize um pêndulo invertido descrito por

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{g \operatorname{sen}(x_1) - \frac{amlx_2^2 \operatorname{sen}(2x_1)}{2} - a \cos(x_1)u}{\frac{4l}{3} - aml \cos^2(x_1)}\end{aligned}$$

considere que g é a aceleração da gravidade, $a = \frac{1}{m + M}$, m é a massa do pêndulo, M é a massa do carrinho, $2l$ é o comprimento do pêndulo.

1. Escolha valores para as constantes do sistema, e simule (no MATLAB ou no Simulink) o comportamento do mesmo em malha fechada com a lei de controle projetada. Apresente os gráficos de ação de controle e comportamento do sistema para uma certa condição inicial. Busque os limites de condições iniciais aceitáveis para os quais a lei de controle estabiliza o sistema.
2. Modifique, de forma aleatória alguns dos parâmetros do sistema (mudança de no máximo 20% do valor do parâmetro), e verifique o comportamento do sistema em malha fechada. Houve alguma mudança significativa?
3. Considere que o sistema agora possui uma força de atrito, ou seja, que sua dinâmica é descrita por

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{g \operatorname{sen}(x_1) - \frac{amlx_2^2 \operatorname{sen}(2x_1)}{2} - a \cos(x_1)u + v}{\frac{4l}{3} - aml \cos^2(x_1)}\end{aligned}$$

$$v = \begin{cases} 0, & \text{se } x_2 = 0, \\ -v_{\text{seco}} \frac{x_2}{|x_2|} - kx_2, & \text{caso contrário} \end{cases}$$

Simule o comportamento deste novo sistema com a mesma lei de controle projetada anteriormente. Comente sobre o comportamento do sistema em malha fechada.

Unidade III

Controle Usando Desigualdades Matriciais Lineares

Aula 20

Sistemas Lineares e Desigualdades Matriciais Lineares

Até este momento, vimos como podemos utilizar o método direto de Lyapunov para verificar a estabilidade de um ponto de equilíbrio de um sistema não linear. Quando temos um sistema linear, podemos especificar um pouco mais as condições do teorema.

Considere o sistema linear invariante no tempo

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$$

$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}$$

Sabemos das aulas de modelagem e controle que este sistema é BIBO estável se a parte real dos seus autovalores for menor do que zero. Sabemos disso, pois

$$G(s) = C(sI - A)^{-1}B + D = C \frac{\text{adj}(sI - A)}{\det(sI - A)}B + D$$

E sabemos que o denominador da função de transferência é dado por $\det(sI - A)$ (a menos que haja algum cancelamento de pólos e zeros).

Para o sistema

$$\dot{\mathbf{x}} = A\mathbf{x}$$

Temos que a resposta temporal do sistema é dada por

$$\mathbf{x}(t) = e^{At}\mathbf{x}(0)$$

em que a matrix e^{At} (chamada de matriz de transição de estados) é limitada (para todo t), implicando que o sistema é estável no sentido de Lyapunov quando todos os autovalores da matrix A tem parte real menor ou igual a zero e todos os autovalores com parte real igual a zero tem multiplicidade algébrica igual à multiplicidade geométrica.

Ademais, converge para zero (implicando que o sistema é assintoticamente estável no sentido de Lyapunov) quando todos os autovalores de A tem parte real negativa.

Uma outra forma de se analisar a estabilidade de um sistema linear é utilizando o Teorema de Estabilidade de Lyapunov. Assim como no caso não linear, basta que encontremos uma função que seja definida positiva e cuja derivada temporal seja definida negativa para provarmos a estabilidade assintótica do ponto de equilíbrio na origem.

A diferença entre os sistemas lineares e os sistemas não lineares é que, para os sistemas lineares sempre podemos encontrar uma função de Lyapunov quadrática que prova que o sistema é estável (ou seja, o Teorema de Estabilidade de Lyapunov pode ser reescrito como uma condição necessária e suficiente).

Teorema 20.1: Teorema de Estabilidade de Lyapunov - Sistemas Lineares Invariantes no Tempo

Um sistema linear com a dinâmica $\dot{\mathbf{x}} = A\mathbf{x}$ é assintoticamente estável se, e somente se, existe uma função quadrática definida positiva

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

Com derivada temporal definida negativa

$$\dot{V}(\mathbf{x}) = \dot{\mathbf{x}}^T P \mathbf{x} + \mathbf{x}^T P \dot{\mathbf{x}} = \mathbf{x}^T (A^T P + P A) \mathbf{x} < 0, \quad \mathbf{x} \neq \mathbf{0}$$

Interlúdio - Sinal de Matrizes

Assim como definimos que uma função é definida (semidefinida) positiva quando ela é maior que zero (maior ou igual a zero) para todo ponto diferente de zero, e que uma função é definida (semidefinida) negativa quando ela é menor que zero (menor ou igual a zero) para todo ponto diferente de zero, definimos que uma matriz é definida (semidefinida) positiva (negativa) quando a forma quadrática associada à ela é definida (semidefinida) positiva (negativa).

$$P > 0 \Leftrightarrow \mathbf{x}^T P \mathbf{x} > 0$$

$$Q < 0 \Leftrightarrow \mathbf{x}^T Q \mathbf{x} < 0$$

Faz sentido que toda vez que formos falar do sinal de uma matriz consideremos apenas matrizes simétricas porque toda matriz pode ser decomposta como

$$R = \left(\frac{R + R^T}{2} \right) + \left(\frac{R - R^T}{2} \right)$$

Com uma parte simétrica e uma parte antissimétrica. Entretanto matrizes antissimétricas têm a propriedade de que

$$\mathbf{x}^T \left(\frac{R - R^T}{2} \right) \mathbf{x} = 0, \forall \mathbf{x}$$

de serem sempre zero (e portanto não contribuindo para a forma quadrática). Dessa forma, quando falamos do sinal de uma matriz sempre estamos nos referindo à parte simétrica da matriz.

Existem várias formas de se verificar se uma matriz é definida positiva, para nós utilizaremos o conceito de que uma matriz simétrica é definida (semidefinida) positiva se, e somente se, todos os seus autovalores são positivos (não negativos). Uma matriz é definida negativa (semidefinida) se, e somente se, todos os seus autovalores são definidos negativos (não positivos).

Além disso, podemos utilizar essas definições para definir um ordenamento parcial das matrizes simétricas (definir quando uma matriz é maior ou menor do que outra).

$$P > Q \Leftrightarrow P - Q > 0$$

Note que o ordenamento é chamado de parcial porque existem casos em que não podemos dizer qual matriz é maior do que qual ($P - Q$ é indefinida/sem sinal).

Algumas propriedades de sinal de matrizes

1. $P > 0 \Rightarrow \det(P) > 0 \Rightarrow \exists P^{-1}$
2. $P > 0 \Leftrightarrow P^{-1} > 0$
3. $P > 0 \Leftrightarrow -P < 0$
4. $P \geq Q > 0 \Leftrightarrow Q^{-1} \geq P^{-1} > 0$
5. $P > 0 \Leftrightarrow M^T P M > 0, \forall M$ com posto completo
6. $P > 0 \Leftrightarrow M^T P M \geq 0, \forall M$
7. $P > 0 \Leftrightarrow P \geq \lambda_{\min} I$, de forma geral, temos $\lambda_{\min} I \leq P \leq \lambda_{\max} I$
8. $P > 0 \Leftrightarrow P > \alpha I, \forall \alpha < \lambda_{\min}$
9. $P > 0 \Leftrightarrow P = R^T R$ (Decomposição de Cholesky/Raiz Quadrada da Matriz)

Esboço de prova - equação de Lyapunov contínua

A suficiência do teorema (a parte se) segue do Teorema de Estabilidade de Lyapunov para sistemas não lineares. Basta provar a necessidade (a parte do somente se - basta

construir uma função de Lyapunov quadrática para um sistema assintoticamente estável).

Como sabemos que o sistema é assintoticamente estável, sabemos que a matriz e^{At} converge para zero (exponencialmente rápido) e por isso que a matriz

$$P = \int_0^{\infty} e^{A^T t} Q e^{At} dt$$

É simétrica, limitada e definida positiva, se $Q > 0$.

Definindo a função de Lyapunov *candidata* como

$$V(\mathbf{x}) = \mathbf{x}^T P \mathbf{x} > 0$$

definida positiva (já que P é definida positiva).

A derivada temporal da função é dada por

$$\dot{V}(\mathbf{x}) = \mathbf{x}^T (A^T P + PA) \mathbf{x}$$

E podemos testar se ela é definida negativa testando se

$$A^T P + PA < 0$$

Mas note que

$$\begin{aligned} A^T P + PA &= \int_0^{\infty} A^T e^{A^T t} Q e^{At} dt + \int_0^{\infty} e^{A^T t} Q e^{At} A dt \\ &= \int_0^{\infty} \frac{d}{dt} (e^{A^T t} Q e^{At}) dt = e^{A^T t} Q e^{At} \Big|_0^{\infty} = -Q < 0 \end{aligned}$$

E podemos ver que a função definida realmente é uma função de Lyapunov. Como fomos capazes de construir uma função de Lyapunov quadrática para um sistema assintoticamente estável arbitrário, fica provada a necessidade do Teorema.

A equação

$$A^T P + PA = -Q$$

É conhecida como equação de Lyapunov (contínua), e permite que testemos a estabilidade de um sistema linear invariante no tempo. Basta escolher uma matriz Q definida positiva, se a matriz P obtida for definida positiva então o sistema é assintoticamente estável. Se ela não for, então o sistema não é assintoticamente estável.

Argumento em favor de desigualdades

Sabe-se, na literatura, que não existe nenhum ganho direto em se resolver a equação de Lyapunov sobre se encontrar os autovalores da matriz A (até porque os autovalores nos

dão mais informação). No capítulo seguinte, entretanto, veremos que existem várias vantagens em se usar as desigualdades sobre a equação de Lyapunov. Dentre elas, será possível analisar casos mais complexos que o linear, a saber: sistemas incertos; sistemas variantes no tempo; e, principalmente neste curso, sistemas não lineares representados como modelos nebulosos Takagi-Sugeno (TS). Ademais, nestes casos, não é possível usar diretamente o conceito de autovalores.

Por ora, vamos apresentar um simples argumento em favor das desigualdades, sob o ponto de vista de factibilidade de soluções. Considere o seguinte conjunto de equações lineares em duas variáveis x e y :

$$\begin{aligned}y &= 2x - 3 \\y &= -1,25x + 2,5\end{aligned}$$

Resolvendo este sistema por meio do método de Jordan

$$\begin{aligned}Ax &= b \\ \begin{bmatrix} 2 & -1 \\ -1,25 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} 3 \\ -2,5 \end{bmatrix} \\ \Downarrow \\ \left(\begin{array}{cc|c} 2 & -1 & 3 \\ -1,25 & -1 & -2,5 \end{array} \right) &\Rightarrow \left(\begin{array}{cc|c} 2 & -1 & 3 \\ 0 & -\frac{13}{8} & -\frac{5}{8} \end{array} \right) \Rightarrow x_1 = \frac{22}{13}, x_2 = \frac{5}{13}\end{aligned}$$

Graficamente, esta solução pode ser visualizada como sendo a interseção das retas $-2x + y = 3$ e $1,25x + y = -2,5$, conforme mostrado na Figura 20.1.

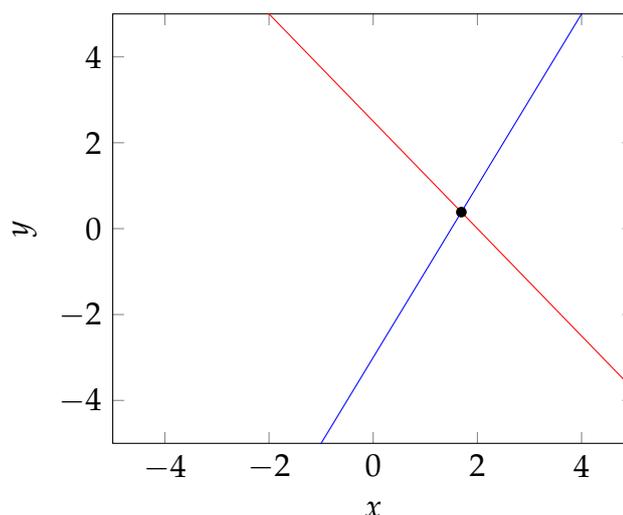


Figura 20.1: Representação gráfica de um sistema de duas equações em duas variáveis. Em azul a restrição $y = 2x - 3$ e em vermelho a restrição $y = -1,25x + 2,5$. A solução é dada pelo ponto $(22/13, 5/13)$, a interseção das duas retas.

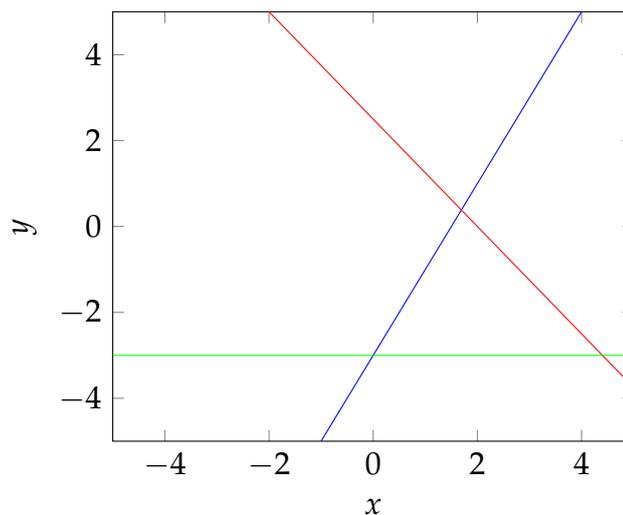


Figura 20.2: Representação gráfica de um sistema de três equações em duas variáveis. Em azul a restrição $y = 2x - 3$; em vermelho a restrição $y = -1,25x + 2,5$; em verde a restrição $y = 3$. A solução não existe neste caso. Graficamente, nota-se que as três retas não possuem uma interseção comum.

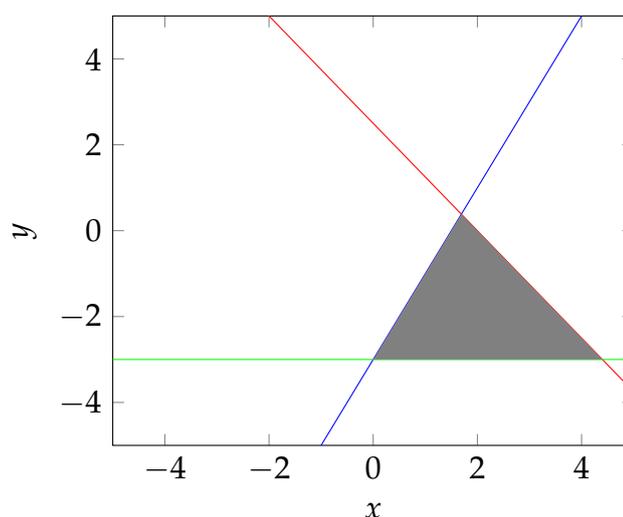


Figura 20.3: Representação gráfica de um sistema de três desigualdades em duas variáveis. Em azul a restrição $y \leq 2x - 3$; em vermelho a restrição $y \leq -1,25x + 2,5$; em verde a restrição $y \geq 3$. A solução existe neste caso, ilustrada pela região cinza.

Considere que uma outra restrição seja adicionada, por exemplo $y \geq 3$. O sistema linear contendo estas três restrições não possui solução, o que pode ser visto, graficamente, na Figura 20.2 como a ausência de um ponto de interseção comum às três retas.

Contudo, considere que agora ao invés de resolver o sistema com três restrições de igualdade, o problema possa ser relaxado. Por exemplo, as novas restrições conduzem

ao sistema de desigualdades a seguir:

$$y \leq 2x - 3$$

$$y \leq -1,25x + 2,5$$

$$y \geq -3$$

Tal problema possui várias soluções, as quais podem ser verificadas graficamente, conforme a Figura 20.3 ilustra.

Nota-se, portanto, que a região viável para um conjunto de desigualdades é, via de regra, maior do que a região viável para o mesmo conjunto de restrições, mas escritas como igualdades. Contudo, numericamente, é mais simples resolver problemas formulados por meio de igualdades do que com desigualdades.

Portanto, muitos problemas de controle até meados da década de 80 eram formulados por meio de equações, como a Equação de Lyapunov. Com o advento de ferramentas de otimização mais eficientes, principalmente a partir dos anos 90, muitos desses problemas passaram a ser formulados como desigualdades matriciais lineares, o tema a ser abordado no próximo capítulo.

Aula 21

Desigualdades Matriciais Lineares Revisitadas

Na Aula 20, definimos rapidamente sinais para matrizes a partir de formas quadráticas associadas à essas matrizes (matrizes definidas positivas/negativas e semidefinidas positivas/negativas).

Vimos também que toda matriz real pode ser decomposta em uma parte simétrica e uma parte antissimétrica, e que a forma quadrática associada a uma matriz antissimétrica é sempre zero. Por conta disso, quando nos referimos ao sinal de uma matriz, estamos nos referindo ao sinal da parte simétrica daquela matriz.

Além disso, ainda na aula 20, vimos que um sistema linear é assintoticamente estável, se, e somente se, existe uma função de Lyapunov quadrática para ele, e que esta condição poderia ser testada ou por meio da equação de Lyapunov

$$A^T P + PA = -Q$$

ou pelas desigualdades

$$P > 0$$

$$A^T P + PA < 0$$

Note que é muito mais fácil se lidar com a Equação de Lyapunov (sistema linear de equações, restrições de igualdade) do que com as desigualdades. Relembre a seção ao final do capítulo anterior. Entretanto, as desigualdades nos permitem uma maior flexibilidade e nos permitem estender os resultados (de forma suficiente) para outros tipos de sistemas (como veremos nesta unidade).

Minha Primeira LMI

Considere o sistema dinâmico linear dado por:

$$\dot{\mathbf{x}} = A\mathbf{x} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \mathbf{x}$$

Assumindo a matriz de Lyapunov simétrica:

$$P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix}$$

sendo $p_i, i = 1, 2, 3$ variáveis reais, e com base no Teorema 20.1, o sistema é assintoticamente estável se:

$$P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} > 0 \Leftrightarrow p_1 \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + p_2 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + p_3 \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} > 0$$

$$A^T P + P A = \begin{bmatrix} -4p_2 & p_1 - 3p_2 - 2p_3 \\ p_1 - 3p_2 - 2p_3 & 2p_2 - 6p_3 \end{bmatrix} < 0$$

\Updownarrow

$$p_1 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + p_2 \begin{bmatrix} -4 & -3 \\ -3 & 2 \end{bmatrix} + p_3 \begin{bmatrix} 0 & -2 \\ -2 & -6 \end{bmatrix} < 0$$

Tais restrições podem ser formuladas como os menores principais líderes de P e $A^T P + P A$:

$$p_1 > 0$$

$$p_3 > 0$$

$$p_1 p_3 - p_2^2 > 0$$

$$4p_2 > 0$$

$$6p_3 - 2p_2 > 0$$

$$(4p_2)(6p_3 - 2p_2) - (3p_2 + 2p_3 - p_1)^2 > 0$$

Note que as restrições apresentadas na forma matricial são lineares. Contudo, ao traduzir a imposição de sinal sobre as matrizes P e $A^T P + P A$ para as variáveis $p_i, i = 1, 2, 3$, as desigualdades obtidas indicam relações não lineares, e.g. termos quadráticos. Mesmo assim, restrições deste tipo resultam em regiões convexas, conforme ilustrado na Figura 21.1.

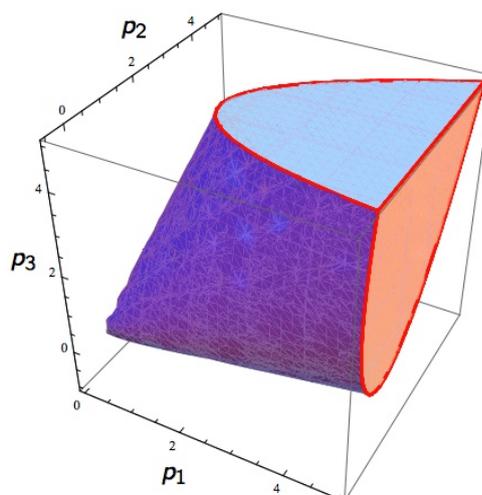


Figura 21.1: Região viável para as restrições associadas ao menores principais de P e $A^T P + PA$.

Formalização de LMIs

Uma Desigualdade Matricial Linear (*Linear Matrix Inequality* - LMI) é uma expressão na forma:

$$F(x) = F_0 + x_1 F_1 + x_2 F_2 + \cdots + x_n F_n < 0$$

sendo:

- $\mathbf{x} = [x_1, x_2, \dots, x_n]$ o vetor de variáveis de decisão (escalares desconhecidos a serem determinados)
- F_0, F_1, \dots, F_n matrizes simétricas conhecidas
- $< (\leq) 0$ indicador de definida (semidefinida) negativo. Em muitas referências adota-se $\prec (\preceq)$ para diferenciar restrições entre matrizes e escalares.

Uma característica interessante das restrições sobre o sinal de matrizes, é que são restrições convexas. Pela definição

$$\mathbf{x}^T P_1 \mathbf{x} > 0 \Leftrightarrow P_1 > 0$$

$$\mathbf{x}^T P_2 \mathbf{x} > 0 \Leftrightarrow P_2 > 0$$

logo, para qualquer $\alpha \in [0, 1]$,

$$P_1 > 0 \text{ e } P_2 > 0 \Leftrightarrow \alpha P_1 + (1 - \alpha) P_2 > 0$$

pois

$$\mathbf{x}^T P_1 \mathbf{x} > 0 \text{ e } \mathbf{x}^T P_2 \mathbf{x} > 0 \Leftrightarrow \alpha \mathbf{x}^T P_1 \mathbf{x} + (1 - \alpha) \mathbf{x}^T P_2 \mathbf{x} > 0$$

Isto é interessante porque, se em um problema de otimização todas as restrições são convexas e a função objetivo é convexa, então todo mínimo local é também um mínimo global. Logo qualquer solução local encontrada pelo problema de otimização é uma solução para o problema!

Outro aspecto interessante reside no fato de LMIs não serem únicas. O uso de transformações de congruência permite estabelecer que:

$$T = \begin{bmatrix} A & B \\ C & D \end{bmatrix} > 0 \Leftrightarrow \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} T \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} > 0 \\ \Leftrightarrow \begin{bmatrix} D & C \\ B & A \end{bmatrix} > 0$$

Múltiplas LMIs podem ser expressadas como uma única LMI. Isso permite considerar várias restrições e solucioná-las simultaneamente. Considere q LMIs a seguir:

$$F_1(x) > 0, F_2(x) < 0, \dots, F_q(x) > 0$$

Uma única LMI pode ser obtida tomando-se:

$$F(x) = \text{diag}\{F_1(x), -F_2(x), \dots, F_q(x)\} > 0$$

Este resultado pode ser comprovado do fato de que os autovalores de um bloco diagonal matriz são iguais à união dos valores próprios do blocos.

Exercício 21.1

Considere a seguinte LMI:

$$F(x) = \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & x_2 \\ x_1 & x_2 & 1 \end{bmatrix} > 0$$

Determine, analiticamente, o conjunto factível.

Apesar de grandes vantagens, a solução de problemas de otimização utilizando LMIs só se tornou viável no início da década de 1990, quando foram desenvolvidos algoritmos eficientes para a solução deste tipo de problemas de otimização (chamados de problemas de Programação Semi-Definida - *Semidefinite Programming*, SDP).

Nesta primeira aula, veremos como derivar LMIs para a estabilidade/estabilização de sistemas lineares a partir do Teorema de Estabilidade de Lyapunov e como escrever e resolver um problema de factibilidade (otimização sem custo) utilizando o YALMIP.

LMIs para análise de estabilidade de sistemas lineares

Consideramos um sistema linear da forma

$$\dot{\mathbf{x}} = A\mathbf{x}$$

e queremos saber se ele é assintoticamente estável. Das aulas de estabilidade de Lyapunov, sabemos que, no caso de um sistema linear, isso é verdade se, e somente se, existe uma função de Lyapunov quadrática que prova a estabilidade do sistema.

Consideramos então uma função da forma

$$V(\mathbf{x}) = \mathbf{x}^T P \mathbf{x} > 0$$

Note que, para garantir que a função seja definida positiva, a matriz P tem de ser definida positiva (e com isto temos nossa primeira LMI)

$$P > 0$$

Tomando a derivada temporal da função de Lyapunov

$$\dot{V}(\mathbf{x}) = \dot{\mathbf{x}}^T P \mathbf{x} + \mathbf{x}^T P \dot{\mathbf{x}} = \mathbf{x}^T A^T P \mathbf{x} + \mathbf{x}^T P A \mathbf{x}$$

$$\dot{V}(\mathbf{x}) = \mathbf{x}^T (A^T P + P A) \mathbf{x} < 0$$

Para garantir que a derivada temporal da função de Lyapunov seja definida negativa temos

$$A^T P + P A < 0$$

Logo, podemos escrever o problema de análise de estabilidade de um sistema linear como, encontrar P tal que

$$P > 0$$

$$A^T P + P A < 0$$

```

1 %considerando que a matriz A já está definida em uma parte anterior
  do código
2 %encontramos o número de estados do sistema
3 n = size(A,1);
4 %declaramos a nossa variável de decisão
5 P = sdpvar(n,n, 'symmetric');
6 %declaramos as nossas restrições
7 %primeiro vamos declarar um "container" vazio para elas
8 restricoes = [];
9 %agora a restrição de que P tem que ser definida positiva

```

```

10 %restricoes = restricoes + (P > 0); %normalmente utilizaríamos esta
    restrição
11 %entretanto, como não existem restrições estritas nos solvers, o uso
    dessa restrição permitiria a solução P=0
12 %logo, ao invés disso, utilizamos P > I
13 restricoes = restricoes + (P > eye(n));
14 %e por fim a restrição de que a derivada temporal da função de
    Lyapunov tem que ser definida negativa
15 restricoes = restricoes + (A'*P + P*A < 0);
16 %em seguida, resolvemos o problema de otimização
17 diagnostico = optimize(restricoes);
18 %visualizamos o resultado de diagnostico para ver se está tudo certo
19 display(diagnostico);
20 %e verificamos se foi possível resolver o problema corretamente
    também pelo resíduo primal
21 primal = checkset(restricoes);
22 display(min(primal)) %para uma solução factível, o resíduo primal
    deve ser positivo
23 %encontramos o valor de P
24 P = double(P);

```

LMI's para síntese de controladores estabilizantes de sistemas lineares

Vamos aproveitar que acabamos de ver as condições de análise de estabilidade de sistemas lineares para aprender sobre um dos aspectos interessantes do uso de LMIs para a análise e síntese de sistemas dinâmicos: As condições de síntese normalmente são construtivas.

Quando dizemos que as condições de síntese são construtivas, neste contexto, queremos dizer que elas são sempre obtidas por meio de modificações às condições de análise.

Suponha que queremos encontrar um controlador, por realimentação de estados, para o sistema

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

Como é um sistema linear, e vamos fazer realimentação de estados, a lei de controle tem a forma

$$\mathbf{u} = \mathbf{K}\mathbf{x}$$

Logo o sistema em malha fechada pode ser escrito como

$$\dot{\mathbf{x}} = (A + BK) \mathbf{x}$$

Aplicando-se os mesmos passos que para as condições de análise (mas colocando $A + BK$ no lugar de A) chegamos às condições

$$\begin{aligned} P &> 0 \\ A^T P + PA + K^T B^T P + PBK &< 0 \end{aligned}$$

Note, entretanto, que nas condições acima aparece um termo com o produto de duas variáveis de decisão (o que faz com que nossas condições não sejam mais LMIs - pois não serão mais lineares ou afins) e a otimização não é capaz de resolver (de forma global) problemas dessa forma. O que podemos fazer nesse caso?

Uma alternativa, muito utilizada, é recorrer às propriedades de sinais de matrizes (vistas na aula 20) e tentar alguma mudança de variáveis que nos leve à LMIs.

Neste caso, podemos multiplicar ambas as condições à esquerda e à direita por P^{-1} para obter

$$\begin{aligned} P^{-1} &> 0 \\ P^{-1} A^T + AP^{-1} + P^{-1} K^T B^T + BKP^{-1} &< 0 \end{aligned}$$

E fazendo as mudanças de variáveis

$$\begin{aligned} X &= P^{-1} \\ Y &= KP^{-1} = KX \end{aligned}$$

Podemos escrever as condições

$$\begin{aligned} X &> 0 \\ AX + X^T A^T + BY + Y^T B^T &< 0 \end{aligned}$$

E as matrizes originais podem ser recuperadas por

$$\begin{aligned} P &= X^{-1} \\ K &= YX^{-1} \end{aligned}$$

```

1 %considerando que a matriz A já está definida em uma parte anterior
  do código
2 %encontramos o número de estados do sistema
3 n = size(A,1);
4 %encontramos o número de entradas de controle
5 m = size(B,2);

```

```

6 %declaramos as nossas variáveis de decisão
7 X = sdpvar(n,n, 'symmetric');
8 Y = sdpvar(m,n, 'full');
9 %declaramos as nossas restrições
10 %primeiro vamos declarar um "container" vazio para elas
11 restricoes = [];
12 %agora a restrição de que X tem que ser definida positiva
13 %restricoes = restricoes + (X > 0); %normalmente utilizaríamos esta
    restrição
14 %entretanto, como não existem restrições estritas nos solvers, o uso
    dessa restrição permitiria a solução X=0
15 %logo, ao invés disso, utilizamos X > I
16 restricoes = restricoes + (X > eye(n));
17 %e por fim a restrição de que a derivada temporal da função de
    Lyapunov tem que ser definida negativa
18 restricoes = restricoes + (A*X + X*A' + B*Y + Y'*B' < 0);
19 %em seguida, resolvemos o problema de otimização
20 diagnostico = optimize(restricoes);
21 %visualizamos o resultado de diagnostico para ver se está tudo certo
22 display(diagnostico);
23 %e verificamos se foi possível resolver o problema corretamente
    também pelo resíduo primal
24 primal = checkset(restricoes);
25 display(min(primal)) %para uma solução factível, o resíduo primal
    deve ser positivo
26 %encontramos os valores de P e K
27 P = inv(double(X));
28 K = double(Y)/double(X);

```

É muito importante se notar que, só faz sentido realizar uma transformação de variáveis que seja bijetora (tenha inversa), porque senão não será possível se recuperar a variável original desejada.

LMI para estabilidade e estabilização de sistemas lineares a tempo discreto

Da mesma forma como chegamos em condições para sistemas a tempo contínuo, podemos chegar em condições para sistemas a tempo discreto.

Consideramos um sistema linear da forma

$$\mathbf{x}_{k+1} = A\mathbf{x}_k$$

e queremos saber se ele é assintoticamente estável. Assim como no caso de um sistema a tempo contínuo, procuramos por uma função de Lyapunov para o sistema. Num paralelo com o caso a tempo contínuo, para sistemas lineares e invariantes no tempo, também vale a afirmação de que um sistema é assintoticamente estável se, e somente se existe uma função de Lyapunov quadrática que comprove a sua estabilidade assintótica.

Consideramos então uma função da forma

$$V(\mathbf{x}) = \mathbf{x}_k^T P \mathbf{x}_k > 0$$

Note que, para garantir que a função seja definida positiva, a matriz P tem de ser definida positiva (e com isto temos nossa primeira LMI)

$$P > 0$$

No caso discreto, ao invés de impor que a derivada temporal da função de Lyapunov seja definida negativa, queremos que a variação seja definida negativa. Logo,

$$\begin{aligned} \Delta V(\mathbf{x}) &= \mathbf{x}_{k+1}^T P \mathbf{x}_{k+1} - \mathbf{x}_k^T P \mathbf{x}_k \\ \Delta V(\mathbf{x}) &= \mathbf{x}^T (A^T P A - P) \mathbf{x} < 0 \end{aligned}$$

Para garantir que a variação da função de Lyapunov seja definida negativa temos

$$A^T P A - P < 0$$

Logo, podemos escrever o problema de análise de estabilidade de um sistema linear como, encontrar P tal que

$$\begin{aligned} P &> 0 \\ A^T P A - P &< 0 \end{aligned}$$

```

1 %considerando que a matriz A já está definida em uma parte anterior
  do código
2 %encontramos o número de estados do sistema
3 n = size(A,1);
4 %declaramos a nossa variável de decisão
5 P = sdpvar(n,n,'symmetric');
6 %declaramos as nossas restrições
7 %primeiro vamos declarar um "container" vazio para elas
8 restricoes = [];
9 %agora a restrição de que P tem que ser definida positiva
10 %restricoes = restricoes + (P > 0); %normalmente utilizaríamos esta
    restrição

```

```

11 %entretanto , como não existem restrições estritas nos solvers , o uso
    dessa restrição permitiria a solução P=0
12 %logo , ao invés disso , utilizamos P > I
13 restricoes = restricoes + (P > eye(n));
14 %e por fim a restrição de que a derivada temporal da função de
    Lyapunov tem que ser definida negativa
15 restricoes = restricoes + (A'*P*A - P < 0);
16 %em seguida , resolvemos o problema de otimização
17 diagnostico = optimize(restricoes);
18 %visualizamos o resultado de diagnostico para ver se está tudo certo
19 display(diagnostico);
20 %e verificamos se foi possível resolver o problema corretamente
    também pelo resíduo primal
21 primal = checkset(restricoes);
22 display(min(primal)) %para uma solução factível , o resíduo primal
    deve ser positivo
23 %encontramos o valor de P
24 P = double(P);

```

No caso da estabilização, suponha que queremos encontrar um controlador, por realimentação de estados, para o sistema

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k$$

Como é um sistema linear, e vamos fazer realimentação de estados, a lei de controle tem a forma

$$\mathbf{u}_k = K\mathbf{x}_k$$

Logo o sistema em malha fechada pode ser escrito como

$$\mathbf{x}_k = (A + BK)\mathbf{x}_k$$

Aplicando-se os mesmos passos que para as condições de análise (mas colocando $A + BK$ no lugar de A) chegamos às condições

$$P > 0$$

$$(A^T + K^T B^T) P (A + BK) - P < 0$$

A segunda condição entretanto não é uma LMI (aparece uma forma quadrática com as variáveis de decisão e uma matriz no meio do quadrado). No caso discreto, antes de fazer a transformação linearizante que fizemos no caso contínuo, precisamos antes de transformar nossa LMI usando o *complemento de Schur*.

Lema 21.1: Complemento de Schur

$$\begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} > 0$$

se, e somente se

$$P_{22} > 0, \quad P_{11} - P_{12}P_{22}^{-1}P_{12}^T > 0$$

ou

$$P_{11} > 0, \quad P_{22} - P_{12}^T P_{11}^{-1} P_{12} > 0$$

A prova deste lema (bem como o enunciado e a prova de outros lemas) pode ser vista no Apêndice A).

Fazendo uso do complemento de Schur, podemos escrever a segunda condição como

$$\begin{bmatrix} -P & A^T + K^T B^T \\ * & -P^{-1} \end{bmatrix} < 0$$

O problema agora que faz com que nossa condição não seja uma LMI é que ela depende de P e de P^{-1} . Uma solução é multiplicá-la à esquerda e à direita por $\begin{bmatrix} P^{-1} & 0 \\ 0 & I \end{bmatrix}$ e fazer as mudanças de variáveis

$$X = P^{-1}$$

$$Y = KP^{-1} = KX$$

Dessa forma, chegamos à condição

$$\begin{matrix} X > 0 \\ \begin{bmatrix} -X & XA^T + Y^T B^T \\ * & -X \end{bmatrix} < 0 \end{matrix}$$

E as matrizes originais podem ser recuperadas por

$$P = X^{-1}$$

$$K = YX^{-1}$$

```

1 %considerando que a matriz A já está definida em uma parte anterior
  do código
2 %encontramos o número de estados do sistema
3 n = size(A,1);
4 %encontramos o número de entradas de controle
5 m = size(B,2);
6 %declaramos as nossas variáveis de decisão

```

```

7 X = sdpvar(n,n,'symmetric');
8 Y = sdpvar(m,n,'full');
9 %declaramos as nossas restrições
10 %primeiro vamos declarar um "container" vazio para elas
11 restricoes = [];
12 %agora a restrição de que X tem que ser definida positiva
13 %restricoes = restricoes + (X > 0); %normalmente utilizaríamos esta
    restrição
14 %entretanto, como não existem restrições estritas nos solvers, o uso
    dessa restrição permitiria a solução X=0
15 %logo, ao invés disso, utilizamos X > I
16 restricoes = restricoes + (X > eye(n));
17 %e por fim a restrição de que a derivada temporal da função de
    Lyapunov tem que ser definida negativa
18 %vamos definir a matriz de blocos usando blkvar (porque aí não
    precisamos escrever a parte simétrica)
19 Q = blkvar;
20 Q(1,1) = -X;
21 Q(1,2) = X*A'+Y'*B';
22 Q(2,2) = -X;
23 Q = sdpvar(Q); %só para ter certeza de que preenchemos certo
24 restricoes = restricoes + (Q < 0);
25 %em seguida, resolvemos o problema de otimização
26 diagnostico = optimize(restricoes);
27 %visualizamos o resultado de diagnostico para ver se está tudo certo
28 display(diagnostico);
29 %e verificamos se foi possível resolver o problema corretamente
    também pelo resíduo primal
30 primal = checkset(restricoes);
31 display(min(primal)) %para uma solução factível, o resíduo primal
    deve ser positivo
32 %encontramos os valores de P e K
33 P = inv(double(X));
34 K = double(Y)/double(X);

```

Exercício 21.2

Resolva novamente o Exercício 21.1, mas usando o Complemento de Schur desta vez.

Exercício 21.3

Considere um sistema linear invariante no tempo descrito por

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{w}, \quad \mathbf{z} = C\mathbf{x} + D\mathbf{w}$$

O ganho \mathcal{L}_2 induzido, ou norma \mathcal{H}_∞ no caso de sistemas lineares, de \mathbf{w} para \mathbf{z} , ignorando o efeito das condições iniciais do sistema, pode ser definido como

$$\sup_{\mathbf{w} \in \mathcal{L}_2} \frac{\|\mathbf{z}(t)\|_2}{\|\mathbf{w}(t)\|_2}.$$

Uma forma de encontrar um limitante superior, γ para este ganho é

$$\|\mathbf{z}(t)\|_2 \leq \gamma \|\mathbf{w}(t)\|_2 \Leftrightarrow \int_0^\infty \frac{1}{\gamma} \mathbf{z}^T(\tau) \mathbf{z}(\tau) d\tau \leq \int_0^\infty \gamma \mathbf{w}^T(\tau) \mathbf{w}(\tau) d\tau$$

Supondo que o sistema seja assintoticamente estável, usamos a seguinte manipulação

$$\int_0^\infty \dot{V}(\mathbf{x}(\tau)) + \frac{1}{\gamma} \mathbf{z}^T(\tau) \mathbf{z}(\tau) - \gamma \mathbf{w}^T(\tau) \mathbf{w}(\tau) d\tau \leq \int_0^\infty \dot{V}(\mathbf{x}(\tau)) d\tau$$

$$\int_0^\infty \dot{V}(\mathbf{x}(\tau)) + \frac{1}{\gamma} \mathbf{z}^T(\tau) \mathbf{z}(\tau) - \gamma \mathbf{w}^T(\tau) \mathbf{w}(\tau) d\tau \leq V(\mathbf{x}(\infty)) - V(\mathbf{x}(0))$$

mas, como estamos considerando o caso em que as condições iniciais são nulas, $V(\mathbf{x}(0)) = 0$. E, como $\mathbf{w} \in \mathcal{L}_2$ e o sistema é assintoticamente estável, $V(\mathbf{x}(\infty)) = 0$.

Logo, sabemos que uma condição suficiente para que o ganho \mathcal{L}_2 induzido seja menor ou igual a γ é que

$$\dot{V} + \frac{1}{\gamma} \mathbf{z}^T \mathbf{z} - \gamma \mathbf{w}^T \mathbf{w} \leq 0$$

Usando esta condição, com uma função de Lyapunov quadrática e o complemento de Schur, encontre um conjunto de restrições LMI que garantam que a norma \mathcal{H}_∞ seja menor ou igual a γ .

Exercício 21.4

Considere um sistema linear invariante no tempo descrito por

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{w}_k, \quad \mathbf{z}_k = C\mathbf{x}_k + D\mathbf{w}_k$$

O ganho ℓ_2 induzido, ou norma \mathcal{H}_∞ no caso de sistemas lineares, de \mathbf{w} para \mathbf{z} , ignorando o efeito das condições iniciais do sistema, pode ser definido como $\sup_{\mathbf{w} \in \ell_2} \frac{\|\mathbf{z}_k\|_2}{\|\mathbf{w}_k\|_2}$.

Uma forma de encontrar um limitante superior, γ para este ganho é

$$\|\mathbf{z}_k\|_2 \leq \gamma \|\mathbf{w}_k\|_2 \Leftrightarrow \sum_{k=0}^{\infty} \frac{1}{\gamma} \mathbf{z}_k^T \mathbf{z}_k \leq \sum_{k=0}^{\infty} \gamma \mathbf{w}_k^T \mathbf{w}_k$$

Supondo que o sistema seja assintoticamente estável, usamos a seguinte manipulação

$$\begin{aligned} \sum_{k=0}^{\infty} \Delta V(\mathbf{x}_k) + \frac{1}{\gamma} \mathbf{z}_k^T \mathbf{z}_k - \gamma \mathbf{w}_k^T \mathbf{w}_k &\leq \sum_{k=0}^{\infty} \Delta V(\mathbf{x}_k) \\ \sum_{k=0}^{\infty} \Delta V(\mathbf{x}_k) + \frac{1}{\gamma} \mathbf{z}_k^T \mathbf{z}_k - \gamma \mathbf{w}_k^T \mathbf{w}_k &\leq V(\mathbf{x}_\infty) - V(\mathbf{x}_0) \end{aligned}$$

mas, como estamos considerando o caso em que as condições iniciais são nulas, $V(\mathbf{x}_0) = 0$. E, como $\mathbf{w} \in \ell_2$ e o sistema é assintoticamente estável, $V(\mathbf{x}_\infty) = 0$.

Logo, sabemos que uma condição suficiente para que o ganho ℓ_2 induzido seja menor ou igual a γ é que

$$\Delta V + \frac{1}{\gamma} \mathbf{z}_k^T \mathbf{z}_k - \gamma \mathbf{w}_k^T \mathbf{w}_k \leq 0$$

Usando esta condição, com uma função de Lyapunov quadrática e o complemento de Schur, encontre um conjunto de restrições LMI que garantam que a norma \mathcal{H}_∞ seja menor ou igual a γ .

Aula 22

Não-Linearidade de Setor

Existem várias estratégias interessantes para se obter um modelo TS de um sistema não-linear, entretanto nesta aula focaremos nossas atenções ao método da **Não-Linearidade de Setor**, por ser a técnica mais comumente utilizada para se encontrar modelos TS exatos dentro de um domínio de análise compacto.

Uma abordagem diferente, e mais voltada para a obtenção de modelos TS de forma numérica, pode ser encontrada no Apêndice C.

Quando trabalhos com modelos TS e LMIs, é comum utilizar um modelo TS composto por um conjunto regras se-então que estabelecem que para um conjunto de variáveis premissas, a dinâmica do sistema pode ser representada por um modelo linear.

Um modelo muito comum, e cuja estrutura pode ser explorada em alguns casos possui a seguinte forma:

$$R_i: \quad \text{Se } x_1 \in \mathcal{M}_1^{\alpha_{i1}} \text{ e } x_2 \in \mathcal{M}_2^{\alpha_{i2}} \text{ e } \dots \text{ e } x_n \in \mathcal{M}_n^{\alpha_{in}}, \\ \text{então } \begin{cases} \dot{\mathbf{x}} = A_i \mathbf{x} + B_i \mathbf{u}, \\ \mathbf{y} = C_i \mathbf{x} + D_i \mathbf{u}, \end{cases}$$

em que α_{ij} é um índice que associa para cada conjunto fuzzy da premissa x_j com a regra i (e.g. um índice $\alpha_{21} = 3$ indica que o terceiro conjunto fuzzy de x_1 é utilizado na segunda regra).

Considerando as funções de pertinência correspondentes as conjuntos $\mathcal{M}_j^{\alpha_{ij}}$

$$0 \leq \omega_j^{\alpha_{ij}}(x_j) \leq 1$$

e considerando uma representação normalizada delas (de modo que sua soma seja sempre igual a 1)

$$w_j^{\alpha_{ij}}(x_j) = \frac{\omega_j^{\alpha_{ij}}(x_j)}{\sum_{\alpha_{ij}=1}^{r_j} \omega_j^{\alpha_{ij}}(x_j)}$$

podemos definir funções de pertinência normalizadas relacionadas com cada regra da forma

$$h_i(\mathbf{x}) = \prod_{j=1}^n w_j^{\alpha_{ij}}(x_j),$$

e o modelo inferido do sistema, pode ser descrito por

$$\begin{aligned}\dot{\mathbf{x}} &= \sum_{i=1}^r h_i(\mathbf{x}) (A_i \mathbf{x} + B_i \mathbf{u}), \\ \mathbf{y} &= \sum_{i=1}^r h_i(\mathbf{x}) (C_i \mathbf{x} + D_i \mathbf{u})\end{aligned}$$

Note que neste modelo, especificamente, consideramos que as variáveis premissas são obrigatoriamente as variáveis de estado do sistema. Este não precisa necessariamente de ser o caso, e um modelo inferido com variáveis premissas arbitrárias, \mathbf{z} , poderia ser escrito como

$$\begin{aligned}\dot{\mathbf{x}} &= \sum_{i=1}^r h_i(\mathbf{z}) (A_i \mathbf{x} + B_i \mathbf{u}), \\ \mathbf{y} &= \sum_{i=1}^r h_i(\mathbf{z}) (C_i \mathbf{x} + D_i \mathbf{u}).\end{aligned}$$

Não-Linearidade de Setor

A não-linearidade de setor é uma técnica que permite encontrar uma representação exata, num domínio compacto, para um sistema não-linear contanto que $\dot{\mathbf{x}} = 0$. Nem sempre a aplicação da não-linearidade de setor leva a modelos no formato apresentado anteriormente (em que apenas as variáveis de estado são premissas, e em que cada conjunto depende apenas de uma variável de estado), mas muitas vezes tal fato não tem tanta importância.

De forma a entender como a técnica funciona, vamos primeiro considerar o caso de uma função escalar. Dentro do domínio desejado, procura-se um setor de modo que

$$f(x) \in [a_1 \ a_2] x.$$

Se for possível reescrever a função como

$$f(x) = g(x)x$$

podemos calcular

$$a_1 = \max g(x)$$

$$a_2 = \min g(x).$$

Dessa forma, podemos reescrever a função como

$$f(x) = \sum_{i=1}^2 w^i a_i x,$$

e encontrar as funções de pertinência w^i como

$$w^1 = \frac{g(x) - a_2}{a_1 - a_2},$$

$$w^2 = \frac{g(x) - a_1}{a_2 - a_1}.$$

Note que essas funções são tais que

$$w^i \geq 0, \quad \forall i$$

$$\sum_{i=1}^2 w^i = 1$$

e portanto podem ser vistas como funções de pertinência normalizadas.

Exemplo 22.1. Considere a função $\text{sen}(x)$ com um domínio $x \in [-\pi, \pi]$ apresentada na Figura 22.1. Podemos reescrever esta função como

$$f(x) = \frac{\text{sen}(x)}{x} x$$

e encontrar o máximo e o mínimo de $\frac{\text{sen}(x)}{x}$ dentro da região de análise. Neste exemplo, temos que os valores máximo e mínimo são 1 e 0, e portanto

$$a_1 = 1$$

$$a_2 = 0$$

Além disso, usando as fórmulas apresentadas (ou o fato de que $w^1(x)a_1x + w^2(x)a_2x = f(x)$ e $w_1(x) + w_2(x) = 1$), temos que

$$w^1 = \frac{\text{sen}(x)}{x}$$

$$w^2 = \frac{x - \text{sen}(x)}{x}$$

Além disso, note na Figura 22.1 que a função $\text{sen}(x)$ está sempre contida dentro dos setores encontrados dentro da região de análise.

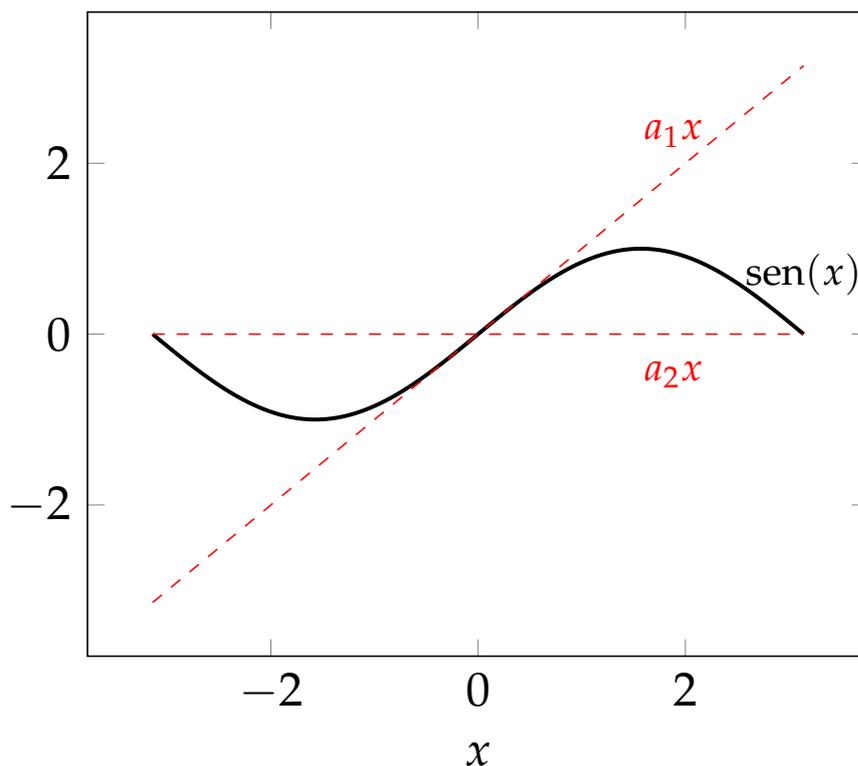


Figura 22.1: Exemplo de modelagem utilizando não linearidade de setor. A função seno, dentro do intervalo de análise apresentado, encontra-se dentro das duas retas apresentadas.

Quando lidamos com sistemas dinâmicos, para encontramos alguma das formas apresentadas anteriormente, primeiramente precisamos de reescrever o sistema na forma

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

ou

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

Entretanto, quando nossas variáveis que causam as variações dos parâmetros dependem dos estados do sistema, tal representação não é única (e portanto as representações em modelos TS de um sistema, mesmo com um conjunto fixo de variáveis de estado estabelecido, não são únicas).

Exemplo 22.2. (Tanaka e Wang 2001, cap. 2, Exemplo 2)

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -x_1 + x_1 x_2^3 \\ -x_2 + (3 + x_2)x_1^3 \end{bmatrix}$$

Devemos encontrar uma representação tal que $\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x})\mathbf{x}$, entretanto, note que essa escolha não é única!

Exemplos:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & x_1 x_2^2 \\ (3 + x_2)x_1^2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 + x_2^3 & 0 \\ 3x_1^2 & -1 + x_1^3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Para o primeiro caso, podemos escrever

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & z_1 \\ z_2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} x_1 x_2^2 \\ (3 + x_2)x_1^2 \end{bmatrix}$$

Supondo $x_1 \in [-1, 1]$ e $x_2 \in [-1, 1]$

$$\begin{aligned} a_1^1 &= \max z_1 = 1, \\ a_2^1 &= \min z_1 = -1, \end{aligned}$$

$$\begin{aligned} a_1^2 &= \max z_2 = 4, \\ a_2^2 &= \min z_2 = 0 \end{aligned}$$

$$\begin{aligned} w_1^1 &= \frac{z_1 + 1}{2}, \\ w_2^1 &= \frac{1 - z_1}{2}, \end{aligned}$$

$$\begin{aligned} w_1^2 &= \frac{z_2}{4}, \\ w_2^2 &= \frac{4 - z_2}{4} \end{aligned}$$

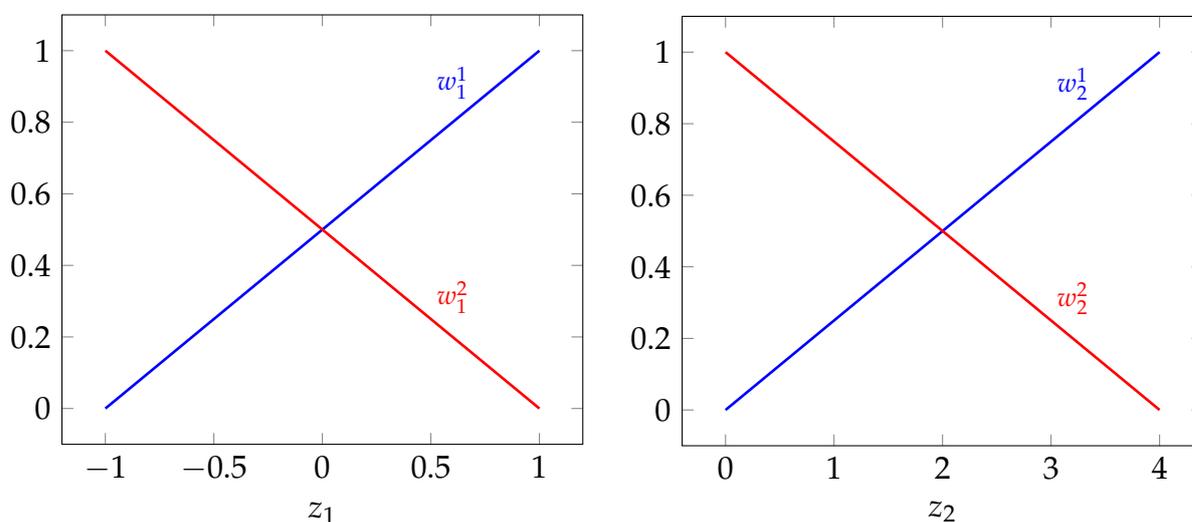


Figura 22.2: Funções de pertinência encontradas para a primeira representação no Exemplo 22.2.

Fazendo uso das funções de pertinência encontradas, o modelo TS do sistema pode ser escrito com as seguintes regras

R_i : Se $z_1 \in \mathcal{M}_1^{\alpha_{i1}}$ e $z_2 \in \mathcal{M}_2^{\alpha_{i2}}$,
então $\dot{\mathbf{x}} = A_i \mathbf{x}$

$$\begin{array}{llll} \alpha_{11} = 1, & \alpha_{21} = 1, & \alpha_{31} = 2, & \alpha_{41} = 2 \\ \alpha_{12} = 1, & \alpha_{22} = 2, & \alpha_{32} = 1, & \alpha_{42} = 2 \\ h_1 = w_1^1 w_2^1, & h_2 = w_1^1 w_2^2, & h_3 = w_1^2 w_2^1, & h_4 = w_1^2 w_2^2 \end{array}$$

E o modelo inferido pode ser escrito como

$$\dot{\mathbf{x}} = \sum_{i=1}^4 h_i A_i \mathbf{x},$$

$$\begin{array}{ll} A_1 = \begin{bmatrix} -1 & 1 \\ 4 & -1 \end{bmatrix} & A_2 = \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix} \\ A_3 = \begin{bmatrix} -1 & -1 \\ 4 & -1 \end{bmatrix} & A_4 = \begin{bmatrix} -1 & -1 \\ 0 & -1 \end{bmatrix} \end{array}$$

Para o segundo caso, podemos escrever

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} z_1 & 0 \\ z_2 & z_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} -1 + x_2^3 \\ 3x_1^2 \\ -1 + x_1^3 \end{bmatrix}$$

Supondo $x_1 \in [-1, 1]$ e $x_2 \in [-1, 1]$

$$\begin{array}{lll} a_1^1 = \max z_1 = 0, & a_1^2 = \max z_2 = 3, & a_1^3 = \max z_3 = 0 \\ a_2^1 = \min z_1 = -2, & a_2^2 = \min z_2 = 0, & a_2^3 = \min z_3 = -2 \end{array}$$

$$\begin{array}{lll} w_1^1 = \frac{z_1 + 2}{2}, & w_2^1 = \frac{z_2}{3}, & w_3^1 = \frac{z_3 + 2}{2} \\ w_1^2 = \frac{-z_1}{2}, & w_2^2 = \frac{3 - z_2}{3}, & w_3^2 = \frac{-z_3}{2} \end{array}$$

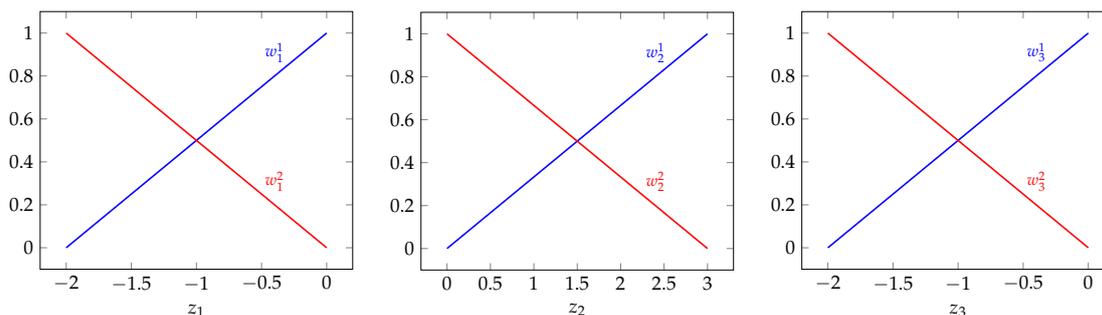


Figura 22.3: Funções de pertinência encontradas para a segunda representação no Exemplo 22.2.

Mudança de variáveis premissas - Curiosidade

Note que neste caso é possível reescrever o modelo do primeiro formato apresentado nesta aula (As variáveis premissas são as variáveis de estado e não tem afirmações da forma " $x_1 \in \mathcal{M}_1^{\alpha_{i1}}$ e $x_1 \in \mathcal{M}_2^{\alpha_{i2}}$ ").

Neste caso, as variáveis premissas z_i são, cada uma, funções de apenas uma variável de estado (diferentemente do caso anterior) e por isso, para deixar o modelo neste formato basta formar novos conjuntos $\overline{\mathcal{M}}_1^{\alpha_{i1}}$ e $\overline{\mathcal{M}}_2^{\alpha_{i2}}$.

Faça

$$\overline{w}_1^1(x_1) = w_2^1 w_3^1,$$

$$\overline{w}_1^3(x_1) = w_2^2 w_3^1,$$

$$\overline{w}_2^1(x_2) = w_1^1,$$

$$\overline{w}_1^2(x_1) = w_2^1 w_3^2,$$

$$\overline{w}_1^4(x_1) = w_2^2 w_3^2,$$

$$\overline{w}_2^2(x_2) = w_1^2,$$

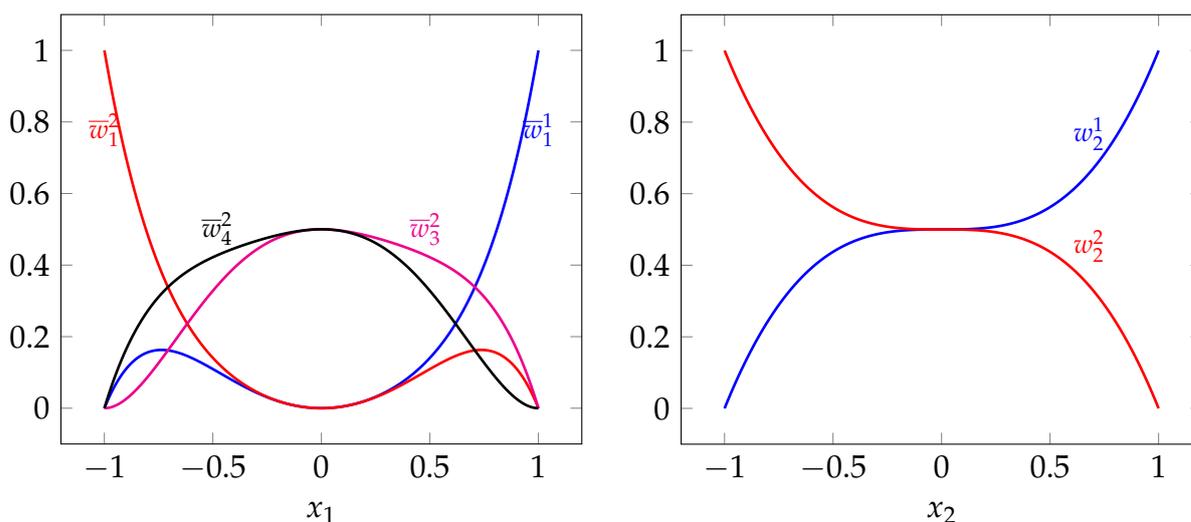


Figura 22.4: Funções de pertinência encontradas para a segunda representação no Exemplo 22.2, com as variáveis de estado como variáveis premissas.

Fazendo uso das funções de pertinência encontradas, o modelo TS do sistema pode ser escrito com as seguintes regras

$$R_i: \text{ Se } x_1 \in \overline{\mathcal{M}}_1^{\alpha_{i1}} \text{ e } x_2 \in \overline{\mathcal{M}}_2^{\alpha_{i2}}, \\ \text{então } \dot{\mathbf{x}} = A_i \mathbf{x}$$

$$\begin{array}{llll} \alpha_{11} = 1, & \alpha_{21} = 1, & \alpha_{31} = 2, & \alpha_{41} = 2 \\ \alpha_{12} = 1, & \alpha_{22} = 2, & \alpha_{32} = 1, & \alpha_{42} = 2 \\ h_1 = \overline{w}_1^1 \overline{w}_2^1, & h_2 = \overline{w}_1^1 \overline{w}_2^2, & h_3 = \overline{w}_1^2 \overline{w}_2^1, & h_4 = \overline{w}_1^2 \overline{w}_2^2 \\ \\ \alpha_{51} = 3, & \alpha_{61} = 3, & \alpha_{71} = 4, & \alpha_{81} = 4 \\ \alpha_{52} = 1, & \alpha_{62} = 2, & \alpha_{72} = 1, & \alpha_{82} = 2 \\ h_5 = \overline{w}_1^3 \overline{w}_2^1, & h_6 = \overline{w}_1^3 \overline{w}_2^2, & h_7 = \overline{w}_1^4 \overline{w}_2^1, & h_8 = \overline{w}_1^4 \overline{w}_2^2 \end{array}$$

E o modelo inferido pode ser escrito como

$$\dot{\mathbf{x}} = \sum_{i=1}^4 h_i A_i \mathbf{x},$$

$$\begin{array}{llll} A_1 = \begin{bmatrix} 0 & 0 \\ 3 & 0 \end{bmatrix} & A_2 = \begin{bmatrix} -2 & 0 \\ 3 & 0 \end{bmatrix} & A_3 = \begin{bmatrix} 0 & 0 \\ 3 & -2 \end{bmatrix} & A_4 = \begin{bmatrix} -2 & 0 \\ 3 & -2 \end{bmatrix} \\ A_5 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & A_6 = \begin{bmatrix} -2 & 0 \\ 0 & 0 \end{bmatrix} & A_7 = \begin{bmatrix} 0 & 0 \\ 0 & -2 \end{bmatrix} & A_8 = \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix} \end{array}$$

De forma a mostrar que ambas as representações TS encontradas acima correspondem exatamente ao sistema dentro do domínio escolhido, podemos plotar o plano de fase do sistema e sua trajetória para algumas condições iniciais.

O código abaixo define uma função que calcula a derivada do sistema utilizando a representação original do sistema. Tal função será utilizada para desenhar o plano de fase do sistema e para simular seu comportamento.

```

1 %Função que retorna o valor da derivada dos estados do sistema , dado
  um valor de x. Temos que colocar t como parâmetro (mesmo não
  utilizando) porque é o formato de função necessário para simular
  usando ode45
2
3 % – Usando a representação original do sistema
4 function dx = derivada_original(t,x)
5     dx = zeros(2,1);
6     dx(1) = -x(1) + x(1)*x(2)^3;
7     dx(2) = -x(2) + (3+x(2))*x(1)^3;
8 end

```

O código abaixo define uma função que calcula a derivada do sistema utilizando a primeira possibilidade para a representação fuzzy.

```

1 %Função que retorna o valor da derivada dos estados do sistema , dado
  um valor de x. Temos que colocar t como parâmetro (mesmo não
  utilizando) porque é o formato de função necessária para simular
  usando ode45
2 % – Usando a representação fuzzy – 1a possibilidade
3 function dx = derivada_pos1(t,x)
4     %Calcula as pertinências do sistema
5     h = pertinencia(x);
6     %Pega as matrizes Ai do sistema
7     A = matrizes_A();
8     %Calcula a derivada pela infêrencia do sistema
9     dx = zeros(2,1);
10    for i = 1:length(A)
11        dx = dx + h(i)*A{i}*x;
12    end
13 end
14
15 %função para calcular as funções de pertinência do sistema
16 function h = pertinencia(x)
17     %Variáveis premissas z
18     z1 = x(1)*x(2)^2;
19     z2 = (3+x(2))*x(1)^2;
20     %Pertinências
21     w11 = (z1+1)/2; w12 = 1- w11;

```

```

22     w21 = (z2)/4; w22 = 1- w21;
23     %Funções de pertinência de cada regra
24     h = zeros(4,1);
25     h(1) = w11*w21; h(2) = w11*w22; h(3) = w12*w21; h(4) = w12*w22;
26 end
27
28 %função que retorna as matrizes Ai do sistema
29 function A = matrizes_A()
30     A = cell(4,1);
31     A{1} = [-1 1; 4 -1];
32     A{2} = [-1 1; 0 -1];
33     A{3} = [-1 -1; 4 -1];
34     A{4} = [-1 -1; 0 -1];
35 end

```

O código abaixo define uma função que calcula a derivada do sistema utilizando a segunda possibilidade para a representação fuzzy.

```

1 %Função que retorna o valor da derivada dos estados do sistema, dado
   um valor de x. Temos que colocar t como parâmetro (mesmo não
   utilizando) porque é o formato de função necessária para simular
   usando ode45
2 % - Usando a representação fuzzy - 2a possibilidade
3 function dx = derivada_pos2(t,x)
4     %Calcula as pertinências do sistema
5     h = pertinencia(x);
6     %Pega as matrizes Ai do sistema
7     A = matrizes_A();
8     %Calcula a derivada pela infêrencia do sistema
9     dx = zeros(2,1);
10    for i = 1:length(A)
11        dx = dx + h(i)*A{i}*x;
12    end
13 end
14
15 %função para calcular as funções de pertinência do sistema
16 function h = pertinencia(x)
17     %Variáveis premissas z
18     z1 = -1 + x(2)^3;
19     z2 = 3*x(1)^2;
20     z3 = -1 + x(1)^3;
21     %Pertinências iniciais
22     w11 = (z1+2)/2; w12 = 1- w11;

```

```

23     w21 = (z2)/3; w22 = 1- w21;
24     w31 = (z3+2)/2; w32 = 1- w31;
25     %Pertinências após a mudança de variáveis premissas (de z para x)
26     W11 = w21.*w31;
27     W12 = w21.*w32;
28     W13 = w22.*w31;
29     W14 = w22.*w32;
30     W21 = w11;
31     W22 = w12;
32     %Funções de pertinência de cada regra
33     h = zeros(8,1);
34     h(1) = W11*W21; h(2) = W11*W22; h(3) = W12*W21; h(4) = W12*W22;
35     h(5) = W13*W21; h(6) = W13*W22; h(7) = W14*W21; h(8) = W14*W22;
36 end
37
38 %função que retorna as matrizes Ai do sistema
39 function A = matrizes_A()
40     A = cell(8,1);
41     A{1} = [0 0; 3 0];
42     A{2} = [-2 0; 3 0];
43     A{3} = [0 0; 3 -2];
44     A{4} = [-2 0; 3 -2];
45     A{5} = zeros(2,2);
46     A{6} = [-2 0; 0 0];
47     A{7} = [0 0; 0 -2];
48     A{8} = [-2 0; 0 -2];
49 end

```

O código a seguir desenha o plano de fase do sistema e o simula de quatro condições iniciais diferentes. Além disso, salva as figuras geradas como pdf.

```

1 %Plota o plano de fase do sistema para as 3 representações diferentes
2
3 %Lembre-se que o exemplo assumiu que  $-1 \leq x_1 \leq 1$ , e  $-1 \leq x_2 \leq 1$ 
4
5 n = 20; %número de amostras de x1 e x2 (número de amostras total = n
6     ^2)
7 x1 = linspace(-1,1,n); %declara n pontos entre -1 e 1
8 x2 = linspace(-1,1,n); %declara n pontos entre -1 e 1
9
10 %declara vetores que vão guardar o vetores de derivada das três
11 %representações
12 dx_orig = zeros(2,n^2);

```

```

12 dx_pos1 = zeros(2,n^2);
13 dx_pos2 = zeros(2,n^2);
14 %declara matriz q vai guardar o x correspondente a cada derivada
15 X = zeros(2,n^2);
16
17 %itera sobre os pontos
18 index = 0;
19 for i = 1:n
20     for j = 1:n
21         index = index + 1;
22         X(:,index) = [x1(i); x2(j)];
23         dx_orig(:,index) = derivada_original(0,X(:,index));
24         dx_pos1(:,index) = derivada_pos1(0,X(:,index));
25         dx_pos2(:,index) = derivada_pos2(0,X(:,index));
26     end
27 end
28
29 %plota o plano de fase do sistema para as três representações
30 figure(1);
31 quiver(X(1,:),X(2,:),dx_orig(1,:),dx_orig(2,:),1);
32 axis([-1 1 -1 1]);
33 xlabel('x_1'); ylabel('x_2');
34
35 figure(2);
36 quiver(X(1,:),X(2,:),dx_pos1(1,:),dx_pos1(2,:),1);
37 axis([-1 1 -1 1]);
38 xlabel('x_1'); ylabel('x_2');
39
40 figure(3);
41 quiver(X(1,:),X(2,:),dx_pos2(1,:),dx_pos2(2,:),1);
42 axis([-1 1 -1 1]);
43 xlabel('x_1'); ylabel('x_2');
44
45 %O plano de fase do sistema varia bastante em módulo, logo fica bem
    ruim de
46 %visualizar. Por isso vamos plotar um outro plano de fase,
    normalizando
47 %para que os vetores tenham todos o mesmo módulo
48 for i = 1:size(dx_orig,2)
49     dx_orig(:,i) = dx_orig(:,i)/norm(dx_orig(:,i));
50     dx_pos1(:,i) = dx_pos1(:,i)/norm(dx_pos1(:,i));

```

```

51     dx_pos2(:,i) = dx_pos2(:,i)/norm(dx_pos2(:,i));
52 end
53
54 figure(4);
55 quiver(X(1,:),X(2,:),dx_orig(1,:),dx_orig(2,:),0.4);
56 axis([-1 1 -1 1]);
57 xlabel('x_1'); ylabel('x_2');
58
59 figure(5);
60 quiver(X(1,:),X(2,:),dx_pos1(1,:),dx_pos1(2,:),0.4);
61 axis([-1 1 -1 1]);
62 xlabel('x_1'); ylabel('x_2');
63
64 figure(6);
65 quiver(X(1,:),X(2,:),dx_pos2(1,:),dx_pos2(2,:),0.4);
66 axis([-1 1 -1 1]);
67 xlabel('x_1'); ylabel('x_2');
68
69 %%Simula o sistema para algumas condições iniciais e plota a evolução
    do sistema ,
70 %no plano de fase
71 %declara as condições iniciais utilizadas
72 x0 = [-0.6 0.6 -0.8 0.8;
73       0.8 0.8 -0.6 -0.6];
74 tf = 10;
75 for i = 1:size(x0,2)
76     [tout_orig,xout_orig] = ode113(@derivada_original,[0 tf],x0(:,i))
77     ;
78     [tout_pos1,xout_pos1] = ode113(@derivada_pos1,[0 tf],x0(:,i));
79     [tout_pos2,xout_pos2] = ode113(@derivada_pos2,[0 tf],x0(:,i));
80     figure(1); hold on;
81     plot(xout_orig(:,1),xout_orig(:,2),'r');
82     figure(2); hold on;
83     plot(xout_pos1(:,1),xout_pos1(:,2),'r');
84     figure(3); hold on;
85     plot(xout_pos2(:,1),xout_pos2(:,2),'r');
86     figure(4); hold on;
87     plot(xout_orig(:,1),xout_orig(:,2),'r');
88     figure(5); hold on;
89     plot(xout_pos1(:,1),xout_pos1(:,2),'r');
90     figure(6); hold on;

```

```
90     plot(xout_pos2(:,1),xout_pos2(:,2),'r');
91 end
92
93 %Salva as figuras para apresentar no slide
94 figure(1);
95 set(gcf,'PaperUnits','centimeters');
96 set(gcf,'PaperSize',[10 10]);
97 set(gcf,'PaperPosition',[0 0 10 10]);
98 print -dpdf exem_Tanaka_orig;
99
100 figure(2);
101 set(gcf,'PaperUnits','centimeters');
102 set(gcf,'PaperSize',[10 10]);
103 set(gcf,'PaperPosition',[0 0 10 10]);
104 print -dpdf exem_Tanaka_pos1;
105
106 figure(3);
107 set(gcf,'PaperUnits','centimeters');
108 set(gcf,'PaperSize',[10 10]);
109 set(gcf,'PaperPosition',[0 0 10 10]);
110 print -dpdf exem_Tanaka_pos2;
111
112 figure(4);
113 set(gcf,'PaperUnits','centimeters');
114 set(gcf,'PaperSize',[10 10]);
115 set(gcf,'PaperPosition',[0 0 10 10]);
116 print -dpdf exem_Tanaka_orig_norm;
117
118 figure(5);
119 set(gcf,'PaperUnits','centimeters');
120 set(gcf,'PaperSize',[10 10]);
121 set(gcf,'PaperPosition',[0 0 10 10]);
122 print -dpdf exem_Tanaka_pos1_norm;
123
124 figure(6);
125 set(gcf,'PaperUnits','centimeters');
126 set(gcf,'PaperSize',[10 10]);
127 set(gcf,'PaperPosition',[0 0 10 10]);
128 print -dpdf exem_Tanaka_pos2_norm;
```

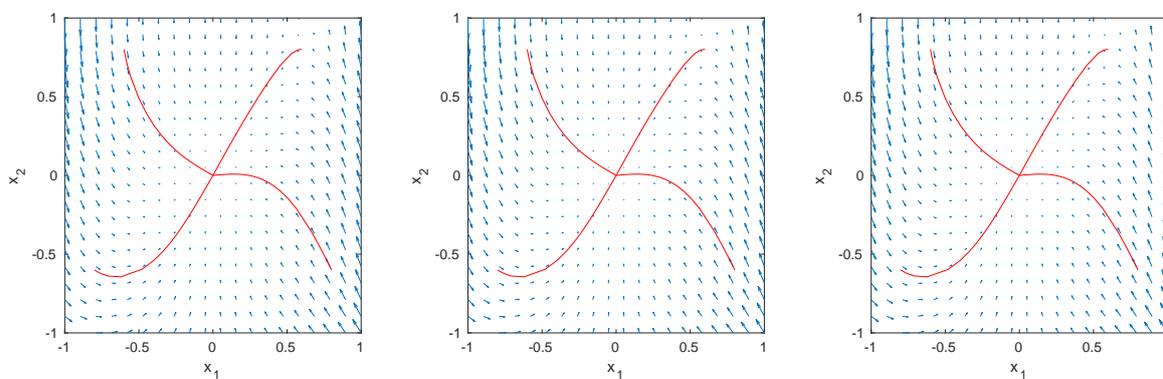


Figura 22.5: Plano de fase para a representação original, a primeira representação TS e a segunda representação TS. As curvas em vermelho representam 4 trajetórias partindo de diferentes condições iniciais

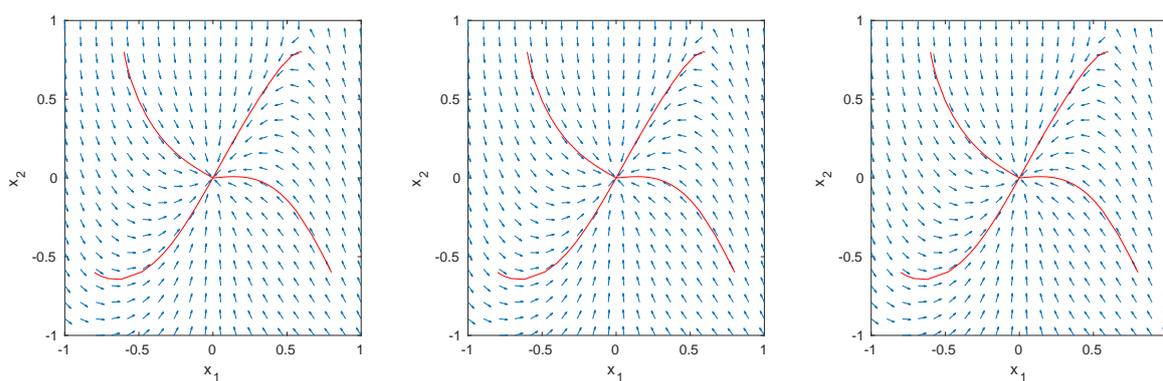


Figura 22.6: Plano de fase normalizado para a representação original, a primeira representação TS e a segunda representação TS. As curvas em vermelho representam 4 trajetórias partindo de diferentes condições iniciais

Exercício 22.1

Considere o modelo de um pêndulo invertido sobre um carrinho (ignorando a dinâmica do carrinho) apresentado no Exercício 19.4:

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{g \operatorname{sen}(x_1) - \frac{amlx_2^2 \operatorname{sen}(2x_1)}{2} - a \cos(x_1)u}{\frac{4l}{3} - aml \cos^2(x_1)}\end{aligned}$$

Encontre uma representação TS para o sistema (Apresente as matrizes de estado encontradas bem como as funções de pertinência normalizadas).

Exercício 22.2

Considere agora o modelo de um pêndulo invertido sobre um carrinho (ignorando a dinâmica do carrinho) com atrito, apresentado no item 3 do Exercício 19.4:

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{g \operatorname{sen}(x_1) - \frac{amlx_2^2 \operatorname{sen}(2x_1)}{2} - a \cos(x_1)u + v}{\frac{4l}{3} - aml \cos^2(x_1)}\end{aligned}$$

$$v = \begin{cases} 0, & \text{se } x_2 = 0 \\ -v_{\text{seco}} \frac{x_2}{|x_2|} - kx_2, & \text{caso contrário} \end{cases}$$

É possível se encontrar uma representação TS (nos moldes das que foram apresentadas nesta aula) para este sistema?

Se sim, apresente tal representação (Apresente as matrizes de estado encontradas bem como as funções de pertinência normalizadas).

Caso contrário, justifique.

Aula 23

Controle PDC

Na Aula de hoje, focaremos nossa atenção em uma das estratégias mais comuns para a estabilidade e estabilização utilizando funções de Lyapunov Quadráticas - o Controle PDC (Compensação Paralela Distribuída - do inglês Parallel Distributed Compensation).

Para introduzir o assunto, vamos começar a aula com desigualdades para o teste de estabilidade quadrática de sistemas TS e estabilização utilizando uma lei de controle linear

Sistemas a tempo contínuo

Consideramos um sistema TS da forma

$$\dot{\mathbf{x}} = \sum_{i=1}^r h_i(\boldsymbol{\theta}) A_i \mathbf{x} \quad (23.1)$$

em que \mathbf{x} são as variáveis de estados e $\boldsymbol{\theta}$ são as variáveis premissas (que podem depender dos estados).

Considerando uma função de Lyapunov da forma

$$V(\mathbf{x}) = \mathbf{x}^T P \mathbf{x} > 0$$

sua derivada temporal será dada por

$$\dot{V}(\mathbf{x}) = \dot{\mathbf{x}}^T P \mathbf{x} + \mathbf{x}^T P \dot{\mathbf{x}} = \sum_{i=1}^r h_i(\boldsymbol{\theta}) \mathbf{x}^T (A_i^T P + P A_i) \mathbf{x}$$

Uma condição *suficiente* para garantir que a derivada temporal da função de Lyapunov seja definida negativa é

$$\sum_{i=1}^r h_i(\boldsymbol{\theta}) (A_i^T P + P A_i) < 0$$

Note, porém, que esta condição tem que ser válida para todos os valores de θ e é portanto uma condição de dimensão infinita. Entretanto, como as funções de pertinência representam uma soma convexa e como as LMIs são convexas, se

$$A_i^T P + P A_i < 0, \quad \forall i$$

então a condição original será satisfeita.

Logo, podemos verificar a estabilidade do sistema não-linear representado pelo modelo TS em (23.1), como o problema de encontrar uma matriz P tal que

$$P > 0$$

$$A_i^T P + P A_i < 0, \quad \forall i$$

Note que, diferentemente do caso linear, não podemos fazer nenhuma afirmação em relação à estabilidade da origem do sistema se não formos capazes de encontrar tal matriz (não é necessário que exista uma função de Lyapunov quadrática para um sistema não-linear).

O código abaixo apresenta como tal teste poderia ser feito utilizando o YALMIP no matlab. Considera-se que as matrizes A do sistema estão disponíveis em uma célula.

```

1 %considerando que a matriz A já está definida em uma parte anterior
  do código
2 %encontramos o número de regras do sistema
3 r = size(A,1);
4 %encontramos o número de estados do sistema
5 n = size(A{1},1);
6 %declaramos a nossa variável de decisão
7 P = sdpvar(n,n,'symmetric');
8 %declaramos as nossas restrições
9 %primeiro vamos declarar um "container" vazio para elas
10 restricoes = [];
11 %agora a restrição de que P tem que ser definida positiva
12 %restricoes = restricoes + (P > 0); %normalmente utilizaríamos esta
  restrição
13 %entretanto, como não existem restrições estritas nos solvers, o uso
  dessa restrição permitiria a solução P=0
14 %logo, ao invés disso, utilizamos P > I
15 restricoes = restricoes + (P > eye(n));
16 %e por fim a restrição de que a derivada temporal da função de
  Lyapunov tem que ser definida negativa
17 for i=1:r
18   restricoes = restricoes + (A{i}'*P + P*A{i} < 0);
19 end

```

```

20 %em seguida , resolvemos o problema de otimização
21 diagnostico = optimize(restricoes);
22 %visualizamos o resultado de diagnostico para ver se está tudo certo
23 display(diagnostico);
24 %e verificamos se foi possível resolver o problema corretamente
    também pelo resíduo primal
25 primal = checkset(restricoes);
26 display(min(primal)) %para uma solução factível , o resíduo primal
    deve ser positivo
27 %encontramos o valor de P
28 P = double(P);

```

Consideramos agora um sistema TS da forma

$$\dot{\mathbf{x}} = \sum_{i=1}^r h_i(\boldsymbol{\theta}) (A_i \mathbf{x} + B_i \mathbf{u})$$

Supondo uma lei de controle linear por realimentação de estados, temos

$$\mathbf{u} = K\mathbf{x}$$

e podemos escrever a malha fechada do sistema como

$$\dot{\mathbf{x}} = \sum_{i=1}^r h_i(\boldsymbol{\theta}) (A_i + B_i K) \mathbf{x}$$

Aplicando-se os mesmos passos que para as condições de análise (mas colocando $A_i + B_i K$ no lugar de A_i) chegamos às condições

$$\mathbf{x}^T P \mathbf{x} > 0$$

$$\sum_{i=1}^r h_i(\boldsymbol{\theta}) \mathbf{x}^T \left(A_i^T P + P A_i + K^T B_i^T P + P B_i K \right) \mathbf{x} < 0$$

Assim como vimos para sistemas lineares, tais restrições não são LMIs (devido aos termos $P B_i K$). Dessa forma, fazemos uma mudança de variáveis no vetores da forma quadrática

$$\mathbf{x} = P^{-1} \bar{\mathbf{x}}$$

Além disso, fazemos as modificações linearizantes de variáveis

$$X = P^{-1}$$

$$Y = K P^{-1}$$

e obtemos as condições

$$\bar{\mathbf{x}}^T \mathbf{X} \bar{\mathbf{x}} > 0$$

$$\sum_{i=1}^r h_i(\boldsymbol{\theta}) \bar{\mathbf{x}}^T \left(\mathbf{X} \mathbf{A}_i^T + \mathbf{A}_i \mathbf{X} + \mathbf{Y}^T \mathbf{B}_i^T + \mathbf{B}_i \mathbf{Y} \right) \bar{\mathbf{x}} < 0$$

Logo, podemos buscar um controlador linear estabilizante para o sistema, buscando por matrizes \mathbf{X} e \mathbf{Y} tal que

$$\mathbf{X} > 0$$

$$\mathbf{X} \mathbf{A}_i^T + \mathbf{A}_i \mathbf{X} + \mathbf{Y}^T \mathbf{B}_i^T + \mathbf{B}_i \mathbf{Y} < 0, \quad \forall i$$

e as matrizes originais podem ser recuperadas por

$$\mathbf{P} = \mathbf{X}^{-1}$$

$$\mathbf{K} = \mathbf{Y} \mathbf{X}^{-1}$$

```

1 %considerando que as matrizes A e B já estão definidas em uma parte
  anterior do código
2 %encontramos o número de regras do sistema
3 r = size(A,1);
4 %encontramos o número de estados do sistema
5 n = size(A{1},1);
6 %encontramos o número de entradas de controle
7 m = size(B{1},2);
8 %declaramos as nossas variáveis de decisão
9 X = sdpvar(n,n,'symmetric');
10 Y = sdpvar(m,n,'full');
11 %declaramos as nossas restrições
12 %primeiro vamos declarar um "container" vazio para elas
13 restricoes = [];
14 %agora a restrição de que X tem que ser definida positiva
15 %restricoes = restricoes + (X > 0); %normalmente utilizaríamos esta
  restrição
16 %entretanto, como não existem restrições estritas nos solvers, o uso
  dessa restrição permitiria a solução X=0
17 %logo, ao invés disso, utilizamos X > I
18 restricoes = restricoes + (X > eye(n));
19 %e por fim a restrição de que a derivada temporal da função de
  Lyapunov tem que ser definida negativa
20 for i = 1:r
21     restricoes = restricoes + (A{i}*X + X*A{i}' + B{i}*Y + Y'*B{i}' <
      0);

```

```

22 end
23 %em seguida , resolvemos o problema de otimização
24 diagnostico = optimize(restricoes);
25 %visualizamos o resultado de diagnostico para ver se está tudo certo
26 display(diagnostico);
27 %e verificamos se foi possível resolver o problema corretamente
    também pelo resíduo primal
28 primal = checkset(restricoes);
29 display(min(primal)) %para uma solução factível , o resíduo primal
    deve ser positivo
30 %encontramos os valores de P e K
31 P = inv(double(X));
32 K = double(Y)/double(X);

```

Note, entretanto, que estas condições ignoram uma informação muito importante que temos nos modelos TS (e que basicamente diferenciam as LMIs para modelos TS e para sistemas com incerteza politópica) - **possuímos o conhecimento das funções de pertinência**.

Considerando que as variáveis premissas também são medidas, podemos utilizar uma lei de controle com as mesmas funções de pertinência que o sistema.

Quando esta lei tem a forma

$$\mathbf{u} = \sum_{j=1}^r h_j(\boldsymbol{\theta}) K_j \mathbf{x}$$

ela recebe o nome de lei de controle PDC. Note que esta família de leis de controle é mais geral do que as leis de controle lineares, uma vez que pode-se recuperar a lei de controle linear no caso em que $K_j = K \forall j$, porque

$$\mathbf{u} = \sum_{j=1}^r h_j(\boldsymbol{\theta}) K \mathbf{x} = K \mathbf{x} \sum_{j=1}^r h_j(\boldsymbol{\theta}) = K \mathbf{x}$$

Seguindo os passos da lei de controle linear, e definindo as matrizes linearizantes

$$\begin{aligned}
 X &= P^{-1} \\
 Y_j &= K_j P^{-1}
 \end{aligned}$$

obtemos as condições

$$\begin{aligned}
 &X > 0 \\
 &\sum_{i=1}^r \sum_{j=1}^r h_i(\boldsymbol{\theta}) h_j(\boldsymbol{\theta}) \left(X A_i^T + A_i X + Y_j^T B_i^T + B_i Y_j \right) < 0
 \end{aligned}$$

Definindo

$$Q_{ij} = XA_i^T + A_iX + Y_j^T B_i^T + B_iY_j$$

poderíamos utilizar uma forma semelhante à que fizemos até agora com um somatório fuzzy simples e impor que

$$Q_{ij} < 0, \quad \forall i, j$$

Note entretanto que tal condição não seria boa, pois seria equivalente às

$$XA_i^T + A_iX + Y_j^T B_i^T + B_iY_j < 0, \quad \forall i, j$$

o que, no fundo, seria equivalente a impor a existência de um controlador linear para o sistema. Ao invés disso, podemos fazer uso das condições suficientes para somatórios fuzzy duplos (algumas das quais são apresentadas no Apêndice B).

Uma estratégia simples, mas bastante utilizada, é apresentada no lema B.1, e nos diz que uma condição suficiente para o somatório duplo é fazer

$$\begin{aligned} Q_{ii} &< 0, \quad \forall i \\ Q_{ij} + Q_{ji} &< 0, \quad \forall i, j > i \end{aligned}$$

Logo, podemos buscar um controlador PDC estabilizante para o sistema, buscando por matrizes X e Y_j tal que

$$\begin{aligned} X &> 0 \\ XA_i^T + A_iX + Y_i^T B_i^T + B_iY_i &< 0, \quad \forall i \\ XA_i^T + A_iX + Y_j^T B_i^T + B_iY_j + XA_j^T + A_jX + Y_i^T B_j^T + B_jY_i &< 0, \quad \forall i, j > i \end{aligned}$$

e as matrizes originais podem ser recuperadas por

$$\begin{aligned} P &= X^{-1} \\ K_j &= Y_j X^{-1} \end{aligned}$$

```

1 %considerando que as matrizes A e B já estão definidas em uma parte
  anterior do código
2 %encontramos o número de regras do sistema
3 r = size(A,1);
4 %encontramos o número de estados do sistema
5 n = size(A{1},1);
6 %encontramos o número de entradas de controle
7 m = size(B{1},2);
8 %declaramos as nossas variáveis de decisão
9 X = sdpvar(n,n, 'symmetric');
10 Y = cell(r,1);
11 for i = 1:r
12     Y{i} = sdpvar(m,n, 'full');
13 end
14 %declaramos as nossas restrições
15 %primeiro vamos declarar um "container" vazio para elas
16 restricoes = [];
17 %agora a restrição de que X tem que ser definida positiva
18 %restricoes = restricoes + (X > 0); %normalmente utilizaríamos esta
  restrição
19 %entretanto, como não existem restrições estritas nos solvers, o uso
  dessa restrição permitiria a solução X=0
20 %logo, ao invés disso, utilizamos X > I
21 restricoes = restricoes + (X > eye(n));
22 %e por fim a restrição de que a derivada temporal da função de
  Lyapunov tem que ser definida negativa
23 for i = 1:r
24     Qii = A{i}*X + X*A{i}' + B{i}*Y{i} + Y{i}'*B{i}';
25     restricoes = restricoes + (Qii < 0);
26     for j = (i+1):r
27         Qij = A{i}*X + X*A{i}' + B{i}*Y{j} + Y{j}'*B{i}';
28         Qji = A{j}*X + X*A{j}' + B{j}*Y{i} + Y{i}'*B{j}';
29         restricoes = restricoes + (Qij + Qji < 0);
30     end
31 end
32 %em seguida, resolvemos o problema de otimização
33 diagnostico = optimize(restricoes);
34 %visualizamos o resultado de diagnostico para ver se está tudo certo
35 display(diagnostico);
36 %e verificamos se foi possível resolver o problema corretamente
  também pelo resíduo primal

```

```

37 primal = checkset(restricoes);
38 display(min(primal)) %para uma solução factível, o resíduo primal
    deve ser positivo
39 %encontramos os valores de P e K
40 P = inv(double(X));
41 K = cell(r,1);
42 for i = 1:r
43     K{i} = double(Y{i})/double(X);
44 end

```

Sistemas a tempo discreto

Consideramos um sistema TS da forma

$$\mathbf{x}_{k+1} = \sum_{i=1}^r h_i(\boldsymbol{\theta}) A_i \mathbf{x}_k$$

e uma função de Lyapunov da forma

$$V(\mathbf{x}) = \mathbf{x}_k^T P \mathbf{x}_k > 0$$

A variação da função de Lyapunov é dada por

$$\begin{aligned} \Delta V(\mathbf{x}) &= \mathbf{x}_{k+1}^T P \mathbf{x}_{k+1} - \mathbf{x}_k^T P \mathbf{x}_k \\ \Delta V(\mathbf{x}) &= \mathbf{x}^T \left(\left(\sum_{i=1}^r h_i(\boldsymbol{\theta}) A_i^T \right) P \left(\sum_{j=1}^r h_j(\boldsymbol{\theta}) A_j \right) - P \right) \mathbf{x} < 0 \end{aligned}$$

Para garantir que a variação da função de Lyapunov seja definida negativa temos

$$\left(\left(\sum_{i=1}^r h_i(\boldsymbol{\theta}) A_i^T \right) P \left(\sum_{j=1}^r h_j(\boldsymbol{\theta}) A_j \right) - P \right) < 0$$

que pode ser reescrita utilizando o complemento de Schur como

$$\sum_{i=1}^r h_i(\boldsymbol{\theta}) \begin{bmatrix} -P & A_i^T P \\ * & -P \end{bmatrix} < 0$$

Logo, podemos escrever o problema de análise de estabilidade como encontrar P tal que

$$\begin{aligned} P &> 0 \\ \begin{bmatrix} -P & A_i^T P \\ * & -P \end{bmatrix} &< 0, \quad \forall i \end{aligned}$$

```

1 %considerando que a matriz A já está definida em uma parte anterior
  do código
2 %encontramos o número de regras do sistema
3 r = size(A,1);
4 %encontramos o número de estados do sistema
5 n = size(A{1},1);
6 %declaramos a nossa variável de decisão
7 P = sdpvar(n,n, 'symmetric');
8 %declaramos as nossas restrições
9 %primeiro vamos declarar um "container" vazio para elas
10 restricoes = [];
11 %agora a restrição de que P tem que ser definida positiva
12 %restricoes = restricoes + (P > 0); %normalmente utilizaríamos esta
  restrição
13 %entretanto, como não existem restrições estritas nos solvers, o uso
  dessa restrição permitiria a solução P=0
14 %logo, ao invés disso, utilizamos P > I
15 restricoes = restricoes + (P > eye(n));
16 %e por fim a restrição de que a derivada temporal da função de
  Lyapunov tem que ser definida negativa
17 for i=1:r
18   Q = blkvar;
19   Q(1,1) = -P;
20   Q(1,2) = A{i}'*P;
21   Q(2,2) = -P;
22   Q = sdpvar(Q);
23   restricoes = restricoes + (Q < 0);
24 end
25 %em seguida, resolvemos o problema de otimização
26 diagnostico = optimize(restricoes);
27 %visualizamos o resultado de diagnostico para ver se está tudo certo
28 display(diagnostico);
29 %e verificamos se foi possível resolver o problema corretamente
  também pelo resíduo primal
30 primal = checkset(restricoes);
31 display(min(primal)) %para uma solução factível, o resíduo primal
  deve ser positivo
32 %encontramos o valor de P
33 P = double(P);

```

Consideramos agora um sistema TS da forma

$$\mathbf{x}_{k+1} = \sum_{i=1}^r h_i(\boldsymbol{\theta}) (A_i \mathbf{x}_k + B_i \mathbf{u}_k)$$

Supondo uma lei de controle linear por realimentação de estados, temos

$$\mathbf{u}_k = K \mathbf{x}_k$$

e podemos escrever a malha fechada do sistema como

$$\mathbf{x}_{k+1} = \sum_{i=1}^r h_i(\boldsymbol{\theta}) (A_i + B_i K) \mathbf{x}_k$$

Aplicando-se os mesmos passos que para as condições de análise (mas colocando $A_i + B_i K$ no lugar de A_i) chegamos às condições

$$\begin{aligned} & \mathbf{x}^T P \mathbf{x} > 0 \\ & \sum_{i=1}^r h_i(\boldsymbol{\theta}) \mathbf{x}^T \left(\left(\sum_{i=1}^r h_i(\boldsymbol{\theta}) A_i^T \right) P \left(\sum_{j=1}^r h_j(\boldsymbol{\theta}) A_j \right) - P \right) \mathbf{x} < 0 \end{aligned}$$

Assim como vimos para sistemas lineares, tais restrições não são LMIs (devido aos termos $P B_i K$). Dessa forma, fazemos uma mudança de variáveis no vetores da forma quadrática

$$\mathbf{x} = P^{-1} \bar{\mathbf{x}}$$

Além disso, fazendo as modificações linearizantes de variáveis

$$\begin{aligned} X &= P^{-1} \\ Y &= K P^{-1} \end{aligned}$$

e fazendo uso do complemento de Schur, obtemos as condições

$$\begin{aligned} & X > 0 \\ & \sum_{i=1}^r h_i(\boldsymbol{\theta}) \begin{bmatrix} -X & X A_i^T + Y^T B_i^T \\ * & -X \end{bmatrix} < 0 \end{aligned}$$

Logo, podemos buscar um controlador linear estabilizante para o sistema, buscando por matrizes X e Y tal que

$$\begin{aligned} & X > 0 \\ & \begin{bmatrix} -X & X A_i^T + Y^T B_i^T \\ * & -X \end{bmatrix} < 0, \quad \forall i \end{aligned}$$

e as matrizes originais podem ser recuperadas por

$$\begin{aligned} P &= X^{-1} \\ K &= Y X^{-1} \end{aligned}$$

```

1 %considerando que as matrizes A e B já estão definidas em uma parte
  anterior do código
2 %encontramos o número de regras do sistema
3 r = size(A,1);
4 %encontramos o número de estados do sistema
5 n = size(A{1},1);
6 %encontramos o número de entradas de controle
7 m = size(B{1},2);
8 %declaramos as nossas variáveis de decisão
9 X = sdpvar(n,n, 'symmetric');
10 Y = sdpvar(m,n, 'full');
11 %declaramos as nossas restrições
12 %primeiro vamos declarar um "container" vazio para elas
13 restricoes = [];
14 %agora a restrição de que X tem que ser definida positiva
15 restricoes = restricoes + (X > eye(n));
16 %e por fim a restrição de que a derivada temporal da função de
  Lyapunov tem que ser definida negativa
17 for i = 1:r
18   Q = blkvar;
19   Q(1,1) = -X;
20   Q(1,2) = X*A{i}' + Y'*B{i}';
21   Q(2,2) = -X;
22   Q = sdpvar(Q);
23   restricoes = restricoes + (Q < 0);
24 end
25 %em seguida, resolvemos o problema de otimização
26 diagnostico = optimize(restricoes);
27 %visualizamos o resultado de diagnostico para ver se está tudo certo
28 display(diagnostico);
29 %e verificamos se foi possível resolver o problema corretamente
  também pelo resíduo primal
30 primal = checkset(restricoes);
31 display(min(primal)) %para uma solução factível, o resíduo primal
  deve ser positivo
32 %encontramos os valores de P e K
33 P = inv(double(X));
34 K = double(Y)/double(X);

```

De forma similar à que fizemos para sistemas a tempo contínuo, podemos utilizar

uma lei de controle PDC da forma

$$\mathbf{u}_k = \sum_{j=1}^r h_j(\boldsymbol{\theta}) K_j \mathbf{x}_k$$

e utilizando a mesma estratégia suficiente para o somatório fuzzy duplo

$$\begin{aligned} Q_{ii} &< 0, \quad \forall i \\ Q_{ij} + Q_{ji} &< 0, \quad \forall i, j > i \end{aligned}$$

Logo, podemos buscar um controlador PDC estabilizante para o sistema, buscando por matrizes X e Y_j tal que

$$\begin{aligned} & X > 0 \\ & \begin{bmatrix} -X & XA_i^T + Y_i^T B_i^T \\ * & -X \end{bmatrix} < 0, \quad \forall i \\ \begin{bmatrix} -X & XA_i^T + Y_j^T B_i^T \\ * & -X \end{bmatrix} + \begin{bmatrix} -X & XA_j^T + Y_i^T B_j^T \\ * & -X \end{bmatrix} < 0, \quad \forall i, j > i \end{aligned}$$

e as matrizes originais podem ser recuperadas por

$$\begin{aligned} P &= X^{-1} \\ K_j &= Y_j X^{-1} \end{aligned}$$

```

1 %considerando que as matrizes A e B já estão definidas em uma parte
  anterior do código
2 %encontramos o número de regras do sistema
3 r = size(A,1);
4 %encontramos o número de estados do sistema
5 n = size(A{1},1);
6 %encontramos o número de entradas de controle
7 m = size(B{1},2);
8 %declaramos as nossas variáveis de decisão
9 X = sdpvar(n,n, 'symmetric');
10 Y = cell(r,1);
11 for i = 1:r
12     Y{i} = sdpvar(m,n, 'full');
13 end
14 %declaramos as nossas restrições
15 %primeiro vamos declarar um "container" vazio para elas
16 restricoes = [];
17 %agora a restrição de que X tem que ser definida positiva
18 %restricoes = restricoes + (X > 0); %normalmente utilizaríamos esta
  restrição
19 %entretanto, como não existem restrições estritas nos solvers, o uso
  dessa restrição permitiria a solução X=0
20 %logo, ao invés disso, utilizamos X > I
21 restricoes = restricoes + (X > eye(n));
22 %e por fim a restrição de que a derivada temporal da função de
  Lyapunov tem que ser definida negativa
23 for i = 1:r
24     Qii = blkvar;
25     Qii(1,1) = -X;
26     Qii(1,2) = X*A{i}' + Y{i}'*B{i}';
27     Qii(2,2) = -X;
28     Qii = sdpvar(Qii);
29     restricoes = restricoes + (Qii < 0);
30     for j = (i+1):r
31         Qij = blkvar;
32         Qij(1,1) = -X;
33         Qij(1,2) = X*A{i}' + Y{j}'*B{i}';
34         Qij(2,2) = -X;
35         Qij = sdpvar(Qij);
36         Qji = blkvar;
37         Qji(1,1) = -X;

```

```
38     Qji(1,2) = X*A{j}' + Y{i}'*B{j}';
39     Qji(2,2) = -X;
40     Qji = sdpcvar(Qji);
41     restricoes = restricoes + (Qij + Qji < 0);
42     end
43 end
44 %em seguida, resolvemos o problema de otimização
45 diagnostico = optimize(restricoes);
46 %visualizamos o resultado de diagnostico para ver se está tudo certo
47 display(diagnostico);
48 %e verificamos se foi possível resolver o problema corretamente
    também pelo resíduo primal
49 primal = checkset(restricoes);
50 display(min(primal)) %para uma solução factível, o resíduo primal
    deve ser positivo
51 %encontramos os valores de P e K
52 P = inv(double(X));
53 K = cell(r,1);
54 for i = 1:r
55     K{i} = double(Y{i})/double(X);
56 end
```

Exercício 23.1

Como vimos no exercício 21.3, podemos encontrar um limitante superior γ para a norma \mathcal{L}_2 induzida de \mathbf{w} para \mathbf{z} de um sistema por meio da desigualdade

$$\dot{V} + \frac{1}{\gamma} \mathbf{z}^T \mathbf{z} - \gamma \mathbf{w}^T \mathbf{w} \leq 0$$

com \dot{V} a derivada temporal de uma função de Lyapunov do sistema.

Considere o sistema não-linear descrito por um modelo TS

$$\begin{aligned} \dot{\mathbf{x}} &= \sum_{i=1}^r h_i(\mathbf{x}) (A_i \mathbf{x} + B_i \mathbf{u}) \\ \mathbf{y} &= \sum_{i=1}^r h_i(\mathbf{x}) (C_i \mathbf{x} + D_i \mathbf{u}) \end{aligned}$$

Queremos que o comportamento do sistema em relação à uma referência desejada para \mathbf{y} , seja próximo ao de um modelo de referência linear

$$\begin{aligned} \dot{\mathbf{x}}_m &= A_m \mathbf{x}_m + B_m \mathbf{r} \\ \mathbf{y} &= C_m \mathbf{x} \end{aligned}$$

Uma abordagem inicial possível seria a de se criar um sistema aumentado

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_m \end{bmatrix} &= \sum_{i=1}^r h_i(\mathbf{x}) \left(\begin{bmatrix} A_i & 0 \\ 0 & A_m \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_m \end{bmatrix} + \begin{bmatrix} B_i \\ 0 \end{bmatrix} \mathbf{u} + \begin{bmatrix} 0 \\ B_m \end{bmatrix} \mathbf{r} \right) \\ (\mathbf{y} - \mathbf{y}_m) &= \sum_{i=1}^r h_i(\mathbf{x}) \left(\begin{bmatrix} C_i & -C_m \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_m \end{bmatrix} + D_i \mathbf{u} \right) \end{aligned}$$

e fazer

$$\mathbf{w} = \mathbf{r}, \quad \mathbf{z} = (\mathbf{y} - \mathbf{y}_m)$$

Considerando uma lei de controle PDC da forma

$$\mathbf{u} = \sum_{j=1}^r h_j(\mathbf{x}) \begin{bmatrix} K_j & F_j \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_m \end{bmatrix}$$

e usando uma função de Lyapunov quadrática, encontre condições LMI para a síntese de um controlador que minimize o custo garantido (minimize um limitante superior) da norma \mathcal{L}_2 induzida de \mathbf{w} (a entrada de referência) para \mathbf{z} (o erro de seguimento).

Exercício 23.2

De forma a garantir um erro em regime estacionário nulo para uma entrada em degrau, é necessário que exista um comportamento integral (no sistema ou no controlador).

Uma forma de garantir isso em um controlador como o do exercício anterior, é aumentando o sistema com um erro integral entre a saída do sistema e a saída do modelo de referência.

Definindo este erro integral com a dinâmica

$$\dot{\ell} = \mathbf{y} - \mathbf{y}_m = \sum_{i=1}^r h_i(\mathbf{x}) \left(\begin{bmatrix} C_i & -C_m \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_m \end{bmatrix} + D_i \mathbf{u} \right)$$

podemos escrever o sistema aumentado como

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}} \\ \dot{\ell} \end{bmatrix} = \sum_{i=1}^r h_i(\mathbf{x}) \left(\begin{bmatrix} A_i & 0 & 0 \\ 0 & A_m & 0 \\ C_i & -C_m & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_m \\ \ell \end{bmatrix} + \begin{bmatrix} B_i \\ 0 \\ D_i \end{bmatrix} \mathbf{u} + \begin{bmatrix} 0 \\ B_m \\ 0 \end{bmatrix} \mathbf{r} \right)$$

Entretanto, escolhendo como saída do sistema o mesmo erro de seguimento que no exercício anterior podemos ter problemas de observabilidade (a integral pode não ter nenhum efeito sobre o comportamento “ótimo” da saída). Dessa forma escolhemos uma nova saída virtual dada por

$$\lambda \ell + (\mathbf{y} - \mathbf{y}_m) = \sum_{i=1}^r h_i(\mathbf{x}) \left(\begin{bmatrix} C_i & -C_m & \lambda I \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_m \\ \ell \end{bmatrix} + D_i \mathbf{u} \right)$$

de modo que se pudessemos forçar tal saída para zero, a dinâmica do erro integral seria $\dot{\ell} = -\lambda \ell$.

Fazendo

$$\mathbf{u} = \sum_{j=1}^r h_j(\mathbf{x}) \begin{bmatrix} K_j & F_j & G_j \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_m \\ \ell \end{bmatrix}, \quad \mathbf{w} = \mathbf{r}, \quad \mathbf{z} = \lambda \ell + (\mathbf{y} - \mathbf{y}_m)$$

e usando uma função de Lyapunov quadrática, encontre condições LMI para a síntese de um controlador que minimize o custo garantido (minimize um limitante superior) da norma \mathcal{L}_2 induzida de \mathbf{w} (a entrada de referência) para \mathbf{z} (a saída virtual).

Exercício 23.3

Encontre as condições LMI equivalentes às dos exercícios anteriores para o caso de um sistema TS a tempo discreto.

Aula 24

Observadores e o princípio da separação

Quando os estados do sistema não estão disponíveis por meio de medições, podemos tentar estimá-los. Uma forma de se realizar tal estimação é pelo uso de observadores de estado. Além disso, veremos que, em certos casos, um princípio da separação similar ao de sistemas lineares pode ser utilizado para modelos TS - o controlador e o observador podem ser sintonizados separadamente sem que um atrapalhe a estabilidade do outro.

Sistemas a tempo contínuo - premissas medidas

Consideramos primeiro o caso em que as variáveis premissas são todas sinais externos, ou variáveis de estado disponíveis para medição - o caso das **variáveis premissas medidas**.

Para esta aula, consideramos sistemas descritos por

$$\begin{aligned}\dot{\mathbf{x}} &= \sum_{i=1}^r h_i(\boldsymbol{\theta}) (A_i \mathbf{x} + B_i \mathbf{u}) \\ \mathbf{y} &= \sum_{i=1}^r h_i(\boldsymbol{\theta}) (C_i \mathbf{x} + D_i \mathbf{u})\end{aligned}$$

Como estamos considerando o caso em que as variáveis premissas são medidas, podemos definir um observador da forma

$$\begin{aligned}\dot{\hat{\mathbf{x}}} &= \sum_{i=1}^r h_i(\boldsymbol{\theta}) (A_i \hat{\mathbf{x}} + B_i \mathbf{u} + L_i (\hat{\mathbf{y}} - \mathbf{y})) \\ \hat{\mathbf{y}} &= \sum_{i=1}^r h_i(\boldsymbol{\theta}) (C_i \hat{\mathbf{x}} + D_i \mathbf{u})\end{aligned}$$

Definindo o erro de estimação como

$$\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x}$$

podemos escrever a dinâmica do erro de estimação como

$$\begin{aligned}\dot{\mathbf{e}} &= \sum_{i=1}^r \sum_{j=1}^r h_i(\boldsymbol{\theta}) h_j(\boldsymbol{\theta}) (A_i \hat{\mathbf{x}} + \cancel{B_i \mathbf{u}} + L_j (C_i \mathbf{x} + \cancel{D_i \mathbf{u}} - C_i \mathbf{x} - \cancel{D_i \mathbf{u}}) - A_i \mathbf{x} - \cancel{B_i \mathbf{u}}) \\ \dot{\mathbf{e}} &= \sum_{i=1}^r \sum_{j=1}^r h_i(\boldsymbol{\theta}) h_j(\boldsymbol{\theta}) (A_i + L_j C_i) \mathbf{e}\end{aligned}$$

Note que, para este caso, a dinâmica do erro não depende dos estados do sistema nem do estimador e também não depende da entrada de controle utilizada.

Fazendo uso da função de Lyapunov quadrática

$$V(\mathbf{e}) = \mathbf{e}^T P \mathbf{e}$$

temos que sua derivada temporal será dada por

$$\begin{aligned}\dot{V}(\mathbf{e}) &= \dot{\mathbf{e}}^T P \mathbf{e} + \mathbf{e}^T P \dot{\mathbf{e}} \\ \dot{V}(\mathbf{e}) &= \mathbf{e}^T \sum_{i=1}^r \sum_{j=1}^r h_i(\boldsymbol{\theta}) h_j(\boldsymbol{\theta}) (A_i^T P + C_i^T L_j^T P + P A_i + P L_j C_i) \mathbf{e}\end{aligned}$$

definindo a variável linearizante

$$H_j = P L_j$$

temos que condições suficientes para a estabilidade do erro de estimação são

$$\begin{aligned}P &> 0, \\ \sum_{i=1}^r \sum_{j=1}^r h_i(\boldsymbol{\theta}) h_j(\boldsymbol{\theta}) (A_i^T P + C_i^T H_j^T + P A_i + H_j C_i) &< 0\end{aligned}$$

utilizando as condições suficientes para somatório fuzzy duplo, temos que condições suficientes para a estabilidade do erro de estimação é que existam matrizes P e H_j tal que

$$\begin{aligned}P &> 0 \\ A_i^T P + C_i^T H_i^T + P A_i + H_i C_i &< 0, \quad \forall i \\ A_i^T P + C_i^T H_j^T + P A_i + H_j C_i + A_j^T P + C_j^T H_i^T + P A_j + H_i C_j &< 0, \quad \forall i, j > i\end{aligned}$$

Suponha agora que desejamos utilizar uma lei de controle PDC para o sistema. Tal lei de controle normalmente seria escrita como

$$\mathbf{u} = \sum_{j=1}^r h_j(\boldsymbol{\theta}) K_j \mathbf{x}$$

entretanto como todos os estados não estão disponíveis para a medição, fazemos

$$\mathbf{u} = \sum_{j=1}^r h_j(\boldsymbol{\theta}) K_j \hat{\mathbf{x}}$$

que por sua vez pode ser reescrita como

$$\mathbf{u} = \sum_{j=1}^r h_j(\boldsymbol{\theta}) (K_j \mathbf{x} + K_j \mathbf{e})$$

logo podemos escrever a dinâmica do sistema (aumentado, considerando estados do sistema e o erro de estimação) como

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{e}} \end{bmatrix} = \sum_{i=1}^r \sum_{j=1}^r h_i(\boldsymbol{\theta}) h_j(\boldsymbol{\theta}) \begin{pmatrix} \begin{bmatrix} A_i + B_i K_j & B_i K_j \\ 0 & A_i + L_j C_i \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix} \end{pmatrix}$$

Considere a função de Lyapunov

$$V(\mathbf{x}, \mathbf{e}) = \mathbf{x}^T P_1 \mathbf{x} + \mathbf{e}^T P_2 \mathbf{e} = \begin{bmatrix} \mathbf{x}^T & \mathbf{e}^T \end{bmatrix} \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix}$$

Sua derivada temporal pode ser escrita como

$$\dot{V} = \begin{bmatrix} \mathbf{x}^T & \mathbf{e}^T \end{bmatrix} \sum_{i=1}^r \sum_{j=1}^r h_i(\boldsymbol{\theta}) h_j(\boldsymbol{\theta}) Q_{ij} \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix}$$

$$Q_{ij} = \begin{bmatrix} P_1 A_i + P_1 B_i K_j + A_i^T P_1 + K_j^T B_i^T P_1 & P_1 B_i K_j \\ * & A_i^T P_2 + C_i^T L_j^T P_2 + P_2 A_i + P_2 L_j C_i \end{bmatrix}$$

Note entretanto que se as condições LMI apresentadas para o observador logo acima são atendidas, temos que $\exists \lambda > 0$ tal que

$$\sum_{i=1}^r \sum_{j=1}^r h_i(\boldsymbol{\theta}) h_j(\boldsymbol{\theta}) \left(A_i^T P + C_i^T L_j^T P + P A_i + P L_j C_i \right) < -\lambda I$$

Além disso, definindo um $\rho > 0$ arbitrário e multiplicando a desigualdade, temos

$$\sum_{i=1}^r \sum_{j=1}^r h_i(\boldsymbol{\theta}) h_j(\boldsymbol{\theta}) \left(A_i^T \rho P + C_i^T L_j^T \rho P + \rho P A_i + \rho P L_j C_i \right) < -\rho \lambda I$$

Definindo $P_2 = \rho P$ e $\bar{\lambda} = \rho \lambda$, temos

$$\sum_{i=1}^r \sum_{j=1}^r h_i(\boldsymbol{\theta}) h_j(\boldsymbol{\theta}) \left(A_i^T P_2 + C_i^T L_j^T P_2 + P_2 A_i + P_2 L_j C_i \right) < -\bar{\lambda} I$$

Além disso, como $\bar{\lambda}$ pode ser escolhido arbitrariamente (independentemente dos ganhos do observador encontrados), temos que, pelo Complemento de Schur, que uma condição para que o controlador encontrado estabilize assintoticamente o sistema é que

$$\sum_{i=1}^r \sum_{j=1}^r h_i(\boldsymbol{\theta}) h_j(\boldsymbol{\theta}) \left(P_1 A_i + P_1 B_i K_j + A_i^T P_1 + K_j^T B_i^T P_1 \right) < 0$$

que é a condição que vimos para a síntese de um controlador PDC estabilizante na última aula.

De modo geral, quando estamos lidando com o caso em que as variáveis premissas são todas medidas, o controlador e o observador de estados (da forma apresentada nesta aula) podem ser sintonizados separadamente sem comprometimentos em relação à estabilidade do sistema em malha fechada. Este fato é conhecido como o *Princípio da Separação*.

Sistemas a tempo contínuo - premissas estimadas

O caso das **variáveis premissas estimadas** já é bem mais complicado de se lidar, e, para o caso geral, não existem garantias de que o *Princípio da Separação* seja válido.

Consideramos agora, sistemas descritos por

$$\begin{aligned} \dot{\mathbf{x}} &= \sum_{i=1}^r h_i(\mathbf{x}) (A_i \mathbf{x} + B_i \mathbf{u}) \\ \mathbf{y} &= \sum_{i=1}^r h_i(\mathbf{x}) C_i \mathbf{x} \end{aligned}$$

Como estamos considerando o caso em que as variáveis premissas são estimadas, podemos definir um observador da forma

$$\begin{aligned} \dot{\hat{\mathbf{x}}} &= \sum_{i=1}^r h_i(\hat{\mathbf{x}}) (A_i \hat{\mathbf{x}} + B_i \mathbf{u} + L_i (\hat{\mathbf{y}} - \mathbf{y})) \\ \hat{\mathbf{y}} &= \sum_{i=1}^r h_i(\hat{\mathbf{x}}) C_i \hat{\mathbf{x}} \end{aligned}$$

Definindo o erro de estimação como

$$\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x}$$

podemos escrever a dinâmica do erro de estimação como

$$\begin{aligned} \dot{\mathbf{e}} &= \sum_{i=1}^r h_i(\hat{\mathbf{x}}) (A_i \hat{\mathbf{x}} + B_i \mathbf{u}) - \sum_{i=1}^r h_i(\mathbf{x}) (A_i \mathbf{x} + B_i \mathbf{u}) + \sum_{j=1}^r h_j(\hat{\mathbf{x}}) L_j (\hat{\mathbf{y}} - \mathbf{y}) \\ \dot{\mathbf{e}} &= \sum_{i=1}^r \sum_{j=1}^r h_i(\hat{\mathbf{x}}) h_j(\hat{\mathbf{x}}) (A_i + L_j C_i) \mathbf{e} + \sum_{j=1}^r h_j(\hat{\mathbf{x}}) \sum_{i=1}^r (h_i(\hat{\mathbf{x}}) - h_i(\mathbf{x})) ((A_i + L_j C_i) \mathbf{x} + B_i \mathbf{u}) \end{aligned} \tag{24.1}$$

Dissipatividade

Note, porém, que $\Delta \mathbf{h} = \mathbf{h}(\hat{\mathbf{x}}) - \mathbf{h}(\mathbf{x})$ possui duas propriedades interessantes

$$\sum_{i=1}^r \Delta h_i = 0$$

$$\Delta h_i \in [-1, 1]$$

A primeira propriedade nos diz que a diferença das funções de pertinência pertence ao plano de soma zero, enquanto que a segunda propriedade nos diz que ela pertence ao hiperretângulo $[-1, 1]^r$ (que é definido pelo produto cartesiano de $[-1, 1]$ r vezes).

Curiosamente, a interseção de um hiperretângulo (ou de qualquer politopo) com um plano, sempre resulta em um politopo (quando ela existe). Considerando que este politopo é formado por ℓ vértices \mathbf{v}_k , a diferença entre as pertinências pode ser escrita como

$$\Delta \mathbf{h} = \sum_{k=1}^{\ell} \alpha_k \mathbf{v}_k$$

com

$$\alpha_k \in [0, 1], \forall k$$

$$\sum_{k=1}^{\ell} \alpha_k = 1$$

e a dinâmica do erro em (24.1) pode ser escrita como

$$\dot{\mathbf{e}} = \sum_{i=1}^r \sum_{j=1}^r \sum_{k=1}^{\ell} h_i(\hat{\mathbf{x}}) h_j(\hat{\mathbf{x}}) \alpha_k \left((A_i + L_j C_i) \mathbf{e} + \sum_{s=1}^r v_{ks} ((A_s + L_j C_s) \mathbf{x} + B_i \mathbf{u}) \right)$$

Supondo uma lei de controle *linear* para o sistema, dada por

$$\mathbf{u} = K \hat{\mathbf{x}}$$

como $\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x}$, podemos reescrevê-la como

$$\mathbf{u} = K \mathbf{x} + K \mathbf{e}$$

e a dinâmica do sistema em malha fechada pode ser escrita como

$$\dot{\mathbf{x}} = \sum_{i=1}^r h_i(\mathbf{x}) ((A_i + B_i K) \mathbf{x} + B_i K \mathbf{e})$$

Diferentemente do caso em que as premissas eram medidas, da forma como apresentamos a dinâmica do erro de estimação e dos estados do sistema, o princípio da separação não é válido e o projeto do controlador e observador não podem ser totalmente desacoplados.

Escrever a dinâmica do sistema aumentado (utilizando os estados do sistema e o erro de estimação) também não é uma boa ideia, pois não seria possível encontrar uma transformação linearizante (por conta das matrizes K e L_j).

Dessa forma, tentaremos chegar em um conjunto de restrições conjuntas para o observador e controlador (utilizados juntamente) a partir da idéia de interconexão neutra de sistemas dissipativos (Willems 1972).

Definição 24.1: Sistema Dinâmico Dissipativo

Um sistema dinâmico é chamado de dissipativo em relação à uma taxa de fornecimento (*supply rate*) $w(\mathbf{u}, \mathbf{y})$ (uma função das entradas do sistema, \mathbf{u} , e das saídas do sistema, \mathbf{y}) se existe uma função não negativa dos estados, chamada de função de armazenamento (*storage function*) que satisfaz a desigualdade de dissipação

$$S(\mathbf{x}(0)) + \int_0^t w(\tau) d\tau \geq S(\mathbf{x}(t))$$

ou escrita de outra forma

$$\dot{S} \leq w$$

Propriedade 24.1: Estabilidade de Sistemas Dissipativos

O ponto de equilíbrio \mathbf{x}^* de um sistema dissipativo é estável se a função de armazenamento é contínua e atinge um mínimo local isolado em \mathbf{x}^* . Além disso, $S(\mathbf{x})$ pode ser usada como função de Lyapunov do sistema.

Definição 24.2: Interconexão neutra de sistemas

Dizemos que a interconexão entre um conjunto de sistemas é neutra se dadas as funções de fornecimento $w_i(\mathbf{u}_i, \mathbf{y}_i)$ dos sistemas, temos

$$\sum_{i=1}^{n_s} w_i(\mathbf{u}_i, \mathbf{y}_i) = 0$$

A interconexão neutra de sistemas é interessante quando ela é aplicada à sistemas dissipativos. Note que, se para cada sistema temos

$$\dot{S}_i - w_i < 0$$

se tomarmos a soma das desigualdades de dissipação teremos

$$\sum_{i=1}^{n_s} \dot{S}_i - \sum_{i=1}^{n_s} w_i < 0$$

mas como a interconexão é neutra, temos que

$$\sum_{i=1}^{n_s} w_i = 0$$

o que nos leva a $\sum_{i=1}^{n_s} \dot{S}_i < 0$. Logo, podemos tomar como função de Lyapunov do sistema $V = \sum_{i=1}^{n_s} S_i$ e teremos que $\dot{V} < 0$, garantindo a estabilidade.

Logo, uma estratégia possível para a síntese do controlador e observador de estados é projetá-los de modo que o erro de estimação seja dissipativo em relação à uma função $w(\mathbf{x}, \mathbf{u}, \mathbf{e})$ e que os estados do sistema sejam dissipativos em relação à $-w(\mathbf{x}, \mathbf{u}, \mathbf{e})$.

Para esta aula, vamos considerar as seguintes funções de armazenamento

$$S_e = \mathbf{e}^T P \mathbf{e}$$

$$S_x = \mathbf{x}^T X^{-1} \mathbf{x}$$

e as funções de fornecimento

$$w_e = -\mathbf{e}^T R^{-1} \mathbf{e} + \begin{bmatrix} \mathbf{x}^T & \mathbf{u}^T \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix}$$

$$w_x = \mathbf{e}^T R^{-1} \mathbf{e} - \begin{bmatrix} \mathbf{x}^T & \mathbf{u}^T \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix}$$

$$w_x = -w_e$$

de modo que nossa função de Lyapunov será dada por

$$V(\mathbf{e}, \mathbf{x}) = S_e + S_x = \mathbf{e}^T P \mathbf{e} + \mathbf{x}^T X^{-1} \mathbf{x}$$

A condição para o erro de estimação leva a

$$S_e - w_e < 0$$

$$\dot{\mathbf{e}}^T P \mathbf{e} + \mathbf{e}^T P \dot{\mathbf{e}} + \mathbf{e}^T R^{-1} \mathbf{e} - \begin{bmatrix} \mathbf{x}^T & \mathbf{u}^T \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{u}} \end{bmatrix} < 0$$

$$\sum_{i=1}^r \sum_{j=1}^r \sum_{k=1}^{\ell} h_i(\hat{\mathbf{x}}) h_j(\hat{\mathbf{x}}) \alpha_k \begin{bmatrix} \mathbf{e}^T & \mathbf{x}^T & \mathbf{u}^T \end{bmatrix} \begin{bmatrix} A_i^T P + P A_i + C_i^T L_j^T P + P L_j C_i + R^{-1} & * & * \\ \sum_{s=1}^r v_{ks} (A_s^T P + C_s^T L_j^T P) & -I & * \\ \sum_{s=1}^r v_{ks} B_s^T P & 0 & -I \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \mathbf{x} \\ \mathbf{u} \end{bmatrix} < 0$$

Fazendo $H_j = P L_j$ e usando o complemento de Schur, temos a condição

$$\sum_{i=1}^r \sum_{j=1}^r \sum_{k=1}^{\ell} h_i(\hat{\mathbf{x}}) h_j(\hat{\mathbf{x}}) \alpha_k \begin{bmatrix} A_i^T P + P A_i + C_i^T H_j^T + H_j C_i & * & * & * \\ \sum_{s=1}^r v_{ks} (A_s^T P + C_s^T H_j^T) & -I & * & * \\ \sum_{s=1}^r v_{ks} B_s^T P & 0 & -I & * \\ I & 0 & 0 & -R \end{bmatrix} < 0$$

Em relação às condições relativas aos estados do sistema, note que, usando $\mathbf{u} = K\mathbf{x} + K\mathbf{e}$, podemos escrever

$$w_x = \mathbf{e}^T R^{-1} \mathbf{e} - \begin{bmatrix} \mathbf{x}^T & \mathbf{u}^T \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix}$$

$$w_x = \mathbf{e}^T R^{-1} \mathbf{e} - \begin{bmatrix} \mathbf{x}^T & \mathbf{e}^T \end{bmatrix} \begin{bmatrix} I & K^T \\ 0 & K^T \end{bmatrix} \begin{bmatrix} I & 0 \\ K & K \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix}$$

logo

$$S_x - w_x < 0$$

$$\dot{\mathbf{x}}^T X^{-1} \mathbf{x} + \mathbf{x}^T X^{-1} \dot{\mathbf{x}} - \mathbf{e}^T R^{-1} \mathbf{e} + \begin{bmatrix} \mathbf{x}^T & \mathbf{e}^T \end{bmatrix} \begin{bmatrix} I & K^T \\ 0 & K^T \end{bmatrix} \begin{bmatrix} I & 0 \\ K & K \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix} < 0$$

$$\sum_{i=1}^r h_i(\mathbf{x}) \begin{bmatrix} \mathbf{x}^T & \mathbf{e}^T \end{bmatrix} Q_i \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix} < 0$$

com

$$Q_i = \begin{bmatrix} A_i^T X^{-1} + X^{-1} A_i + K^T B_i^T X^{-1} + X^{-1} B_i K & X^{-1} B_i K \\ * & -R^{-1} \end{bmatrix} + \begin{bmatrix} I & K^T \\ 0 & K^T \end{bmatrix} \begin{bmatrix} I & 0 \\ K & K \end{bmatrix}$$

Fazendo a transformação de congruência $\mathbf{x} = X\bar{\mathbf{x}}$ e $\mathbf{e} = X\bar{\mathbf{e}}$ (equivalente a multiplicar Q_i à esquerda e à direita por $\begin{bmatrix} X & 0 \\ 0 & X \end{bmatrix}$), $Y = KX$, utilizando o complemento de Schur e o Lema A.1 (com $\mu = 1$), as condições relativas aos estados do sistema podem ser escritas como

$$\sum_{i=1}^r h_i(\mathbf{x}) \begin{bmatrix} A_i X + X A_i^T + B_i Y + Y^T B_i^T & B_i Y & X & Y^T \\ * & -2X + R & 0 & Y^T \\ * & * & -I & 0 \\ * & * & * & -I \end{bmatrix} < 0$$

Logo, condições suficientes para a existência de um conjunto controlador/observador que estabiliza o sistema é que existam matrizes $X = X^T$, Y , $P = P^T$, H_j e R tal que

$$\sum_{i=1}^r \sum_{j=1}^r \sum_{k=1}^{\ell} h_i(\hat{\mathbf{x}}) h_j(\hat{\mathbf{x}}) \alpha_k \begin{bmatrix} A_i^T P + P A_i + C_i^T H_j^T + H_j C_i & * & * & * \\ \sum_{s=1}^r v_{ks} (A_s^T P + C_s^T H_j^T) & -I & * & * \\ \sum_{s=1}^r v_{ks} B_s^T P & 0 & -I & * \\ I & 0 & 0 & -R \end{bmatrix} < 0$$

$$\sum_{i=1}^r h_i(\mathbf{x}) \begin{bmatrix} A_i X + X A_i^T + B_i Y + Y^T B_i^T & B_i Y & X & Y^T \\ * & -2X + R & 0 & Y^T \\ * & * & -I & 0 \\ * & * & * & -I \end{bmatrix} < 0$$

o ganho do controlador pode ser recuperado por $K = YX^{-1}$ e os ganhos do observador podem ser recuperados por $L_j = P^{-1} H_j$.

Teorema do Valor Médio

Uma abordagem mais comum (Guerra et al. 2018; Ichalal, Arioui e Mammar 2011) no caso das premissas estimadas é fazer uso do Teorema do Valor Médio (ou alguma de suas variações).

Teorema 24.1: Teorema do Valor Médio

Seja G um subconjunto aberto convexo de \mathbb{R}^n e $f : G \rightarrow \mathbb{R}$ uma função diferenciável, então para \mathbf{a} e \mathbf{b} em G temos

$$f(\mathbf{a}) - f(\mathbf{b}) = \langle \nabla f(\mathbf{c}), (\mathbf{a} - \mathbf{b}) \rangle$$

com $\mathbf{c} = \alpha \mathbf{a} + (1 - \alpha) \mathbf{b}$ um ponto no segmento de reta entre \mathbf{a} e \mathbf{b} .

Voltando ao comportamento do erro de estimação na equação (24.1), temos

$$\dot{\mathbf{e}} = \sum_{i=1}^r \sum_{j=1}^r h_i(\hat{\mathbf{x}}) h_j(\hat{\mathbf{x}}) (A_i + L_j C_i) \mathbf{e} + \sum_{j=1}^r h_j(\hat{\mathbf{x}}) \sum_{i=1}^r (h_i(\hat{\mathbf{x}}) - h_i(\mathbf{x})) ((A_i + L_j C_i) \mathbf{x} + B_i \mathbf{u})$$

mas, usando o Teorema do valor médio, temos que

$$h_i(\hat{\mathbf{x}}) - h_i(\mathbf{x}) = \nabla h_i(\mathbf{z}_i) \mathbf{e}$$

e, pelas propriedades das funções de pertinência, temos que

$$\sum_{i=1}^r (h_i(\hat{\mathbf{x}}) - h_i(\mathbf{x})) = 0$$

$$\sum_{i=1}^r (h_i(\hat{\mathbf{x}}) - h_i(\mathbf{x})) \left(\sum_{k=1}^r \sum_{\ell=1}^r h_k(\hat{\mathbf{x}}) h_\ell(\hat{\mathbf{x}}) P^{-1} (\Lambda_{k\ell} \mathbf{x} + \Gamma_{k\ell} \mathbf{u}) \right) = 0$$

logo a dinâmica do erro de observação pode ser reescrita como

$$\dot{\mathbf{e}} = \sum_{i=1}^r \sum_{j=1}^r h_i(\hat{\mathbf{x}}) h_j(\hat{\mathbf{x}}) \left(A_i + L_j C_i + \sum_{k=1}^r \left((A_k + L_j C_k + P^{-1} \Lambda_{ij}) \mathbf{x} + (B_k + P^{-1} \Gamma_{ij}) \mathbf{u} \right) \nabla h_k(\mathbf{z}_k) \right) \mathbf{e}$$

Tomando uma função de Lyapunov da forma

$$V(\mathbf{e}) = \mathbf{e}^T P \mathbf{e}$$

sua derivada temporal é dada por

$$\dot{V} = \dot{\mathbf{e}}^T P \mathbf{e} + \mathbf{e}^T P \dot{\mathbf{e}}$$

$$\dot{V} = \sum_{i=1}^r \sum_{j=1}^r h_i(\hat{\mathbf{x}}) h_j(\hat{\mathbf{x}}) \mathbf{e}^T Q_{ij} \mathbf{e}$$

com

$$Q_{ij} = \left(A_i + L_j C_i + \sum_{k=1}^r \left((A_k + L_j C_k + P^{-1} \Lambda_{ij}) \mathbf{x} + (B_k + P^{-1} \Gamma_{ij}) \mathbf{u} \right) \nabla h_k(\mathbf{z}_k) \right)^T P + (*)$$

com o * indicando que o transposto do termo é somado a ele.

Supondo que o vetor de estados do sistema, as entradas de controle e o gradiente das funções de pertinência são limitados e podem ser descritos como pertencendo a um politopo, podemos escrevê-los como

$$\begin{aligned} \mathbf{x} &= \sum_{\ell=1}^{n_x} \alpha_{\ell} \mathbf{x}_{\ell} \\ \mathbf{u} &= \sum_{m=1}^{n_u} \beta_m \mathbf{u}_m \\ \nabla h_k(\mathbf{z}_k) &= \sum_{n_k=1}^{n_{d_k}} \gamma_{n_k} \mathbf{v}_{n_k} \end{aligned}$$

com

$$\begin{aligned} \sum_{\ell=1}^{n_x} \alpha_{\ell} &= 1, \quad \alpha_{\ell} \in [0, 1] \\ \sum_{m=1}^{n_u} \beta_m &= 1, \quad \beta_m \in [0, 1] \\ \sum_{n_k=1}^{n_{d_k}} \gamma_{n_k} &= 1, \quad \gamma_{n_k} \in [0, 1] \end{aligned}$$

Definindo

$$H_j = PL_j$$

podemos reescrever a derivada temporal da função de Lyapunov como

$$\dot{V} = \sum_{i=1}^r \sum_{j=1}^r \sum_{\ell=1}^{n_x} \sum_{m=1}^{n_u} \underbrace{\sum_{n_1=1}^{n_{d_1}} \cdots \sum_{n_r=1}^{n_{d_r}}}_{r \text{ vezes}} h_i(\hat{\mathbf{x}}) h_j(\hat{\mathbf{x}}) \alpha_\ell \beta_m \underbrace{\gamma_{n_1} \cdots \gamma_{n_r}}_{r \text{ vezes}} \mathbf{e}^T \bar{Q}_{ij\ell mn_1 \dots n_r} \mathbf{e}$$

com

$$\bar{Q}_{ij\ell mn_1 \dots n_r} = A_i^T P + P A_i + C_i^T H_j^T + H_j C_i + \sum_{k=1}^r \left(((P A_k + H_j C_k + \Lambda_{ij}) \mathbf{x}_\ell + (P B_k + \Gamma_{ij}) \mathbf{u}_m) v_{n_k} + v_{n_k}^T \left(\mathbf{x}_\ell^T (A_k^T P + C_k^T H_j^T + \Lambda_{ij}^T) + \mathbf{u}_m^T (B_k^T P + \Gamma_{ij}^T) \right) \right)$$

E, utilizando condições suficientes para o somatório duplo (em i e em j), temos que uma condição suficiente para que exista um observador de estados com erro de estimação assintoticamente estável (dado que os estados, as entradas e os gradientes das pertinências permaneçam dentro do politopo utilizado para análise) é que as LMIs abaixo sejam satisfeitas.

$$P > 0$$

$$\bar{Q}_{i\ell mn_1 \dots n_r} < 0, \quad \forall i, \ell, m, n_1, \dots, n_r$$

$$\bar{Q}_{ij\ell mn_1 \dots n_r} + \bar{Q}_{ji\ell mn_1 \dots n_r} < 0, \quad \forall i, j > i, \ell, m, n_1, \dots, n_r$$

Exercício 24.1

Mostre que, se os limites utilizados no observador de estados com premissas estimadas usando o Teorema do Valor Médio forem respeitados, o princípio da separação vale para uma *lei de controle linear*.

Exercício 24.2

Encontre condições LMI similares para um sistema TS a *tempo discreto*.

Exercício 24.3

Considere o modelo encontrado para o pêndulo invertido no Exercício 22.1.

- a) Encontre uma lei de controle PDC para o sistema e a simule em funcionamento.
- b) Suponha que apenas o ângulo x_1 é medido. Utilizando os resultados para o caso em que as variáveis premissas são medidas, projete um observador de estados para o sistema e verifique seu comportamento em conjunto com a lei de controle PDC previamente encontrada.
- c) Supondo ainda que apenas o ângulo x_1 é medido. Encontre um observador de estados para o sistema tal que as premissas estimadas sejam utilizadas ao invés das premissas medidas. Projete uma lei de controle linear para o sistema e verifique seu comportamento em conjunto com o observador de estados. Apresente os resultados de simulação.

Aula 25

Funções de Lyapunov Fuzzy

Como estamos lidando com sistemas não-lineares, a busca por uma função de Lyapunov quadrática para o sistema introduz um certo conservadorismo em nossas condições de análise e síntese. Dessa forma, na aula de hoje vamos focar nossa atenção na família de funções de Lyapunov fuzzy, definidas a partir das funções de pertinência do sistema.

Diferente das outras aulas, vamos começar nossa aula a partir das condições para sistemas a tempo discreto, e em seguida prosseguiremos para os sistemas a tempo contínuo.

Ao longo desta aula, vamos considerar uma notação mais curta, de modo que fique mais fácil de escrever as condições propostas. Desta forma, para um conjunto de matrizes Q_i , de funções de pertinência $h_i(\cdot)$ e de variáveis premissas θ_k (no caso discreto) ou $\theta(t)$ (no caso contínuo), escrevemos

$$\begin{aligned} \sum_{i=1}^r h_i(\theta_k) Q_i &= Q_{\theta_k}, & \left(\sum_{i=1}^r h_i(\theta_k) Q_i \right)^{-1} &= Q_{\theta_k}^{-1}, \\ \sum_{i=1}^r h_i(\theta_{k+1}) Q_i &= Q_{\theta_{k+1}}, & \sum_{i=1}^r h_i(\theta_{k-1}) Q_i &= Q_{\theta_{k-1}}, \\ \sum_{i=1}^r h_i(\theta(t)) Q_i &= Q_{\theta}, & \sum_{i=1}^r \dot{h}_i(\theta(t)) Q_i &= \dot{Q}_{\theta} \end{aligned}$$

Sistemas a tempo discreto

Consideramos um sistema TS da forma

$$\mathbf{x}_{k+1} = A_{\theta_k} \mathbf{x}_k + B_{\theta_k} \mathbf{u}_k$$

e uma função de Lyapunov da forma

$$V(\mathbf{x}) = \mathbf{x}_k^T P_{\theta_k}^{-1} \mathbf{x}_k > 0$$

Lei de Controle PDC

Note que se utilizarmos uma lei de controle PDC da forma

$$\mathbf{u}_k = K_{\theta_k} \mathbf{x}_k$$

o sistema em malha fechada será descrito por

$$\mathbf{x}_{k+1} = (A_{\theta_k} + B_{\theta_k} K_{\theta_k}) \mathbf{x}_k$$

A variação da função de Lyapunov é dada por

$$\begin{aligned} \Delta V(\mathbf{x}) &= \mathbf{x}_{k+1}^T P_{\theta_{k+1}}^{-1} \mathbf{x}_{k+1} - \mathbf{x}_k^T P_{\theta_k}^{-1} \mathbf{x}_k \\ \Delta V(\mathbf{x}) &= \mathbf{x}_k^T \left((A_{\theta_k} + B_{\theta_k} K_{\theta_k})^T P_{\theta_{k+1}}^{-1} (A_{\theta_k} + B_{\theta_k} K_{\theta_k}) - P_{\theta_k}^{-1} \right) \mathbf{x}_k < 0 \end{aligned}$$

Para garantir que a variação da função de Lyapunov seja definida negativa temos

$$(A_{\theta_k} + B_{\theta_k} K_{\theta_k})^T P_{\theta_{k+1}}^{-1} (A_{\theta_k} + B_{\theta_k} K_{\theta_k}) - P_{\theta_k}^{-1} < 0$$

que pode ser reescrita utilizando o complemento de Schur como

$$\begin{bmatrix} -P_{\theta_k}^{-1} & A_{\theta_k}^T + K_{\theta_k}^T B_{\theta_k}^T \\ * & -P_{\theta_{k+1}}^{-1} \end{bmatrix} < 0 \quad (25.1)$$

Se a matriz P_{θ_k} não variasse com as funções de pertinência, poderíamos fazer uma transformação de congruência, multiplicando à esquerda e à direita por $\text{diag}(P, I)$, e definindo $Y_j = K_j P$ teríamos condições LMI para a síntese de um controlador estabilizante.

Entretanto, como $P_{\theta_k} = \sum_{i=1}^r h_i(\theta_k) P_i$, se fizermos uma transformação de congruência, multiplicando à esquerda e à direita por $\text{diag}(P_{\theta_k}, I)$, teríamos o produto

$$K_{\theta_k} P_{\theta_k} = \sum_{j=1}^r \sum_{\ell=1}^r h_j(\theta_k) h_{\ell}(\theta_k) K_j P_{\ell}$$

e não podemos fazer a transformação de variáveis $Y_{j\ell} = K_j P_{\ell}$ uma vez que a partir das variáveis de decisão $Y_{j\ell}$ não seríamos capazes de recuperar as matrizes K_j (pois não

existe garantia de que os mesmos ganhos K_j sejam encontrados para diferentes índices ℓ).

Desta forma, consideramos uma transformação de congruência em que as condições LMI apresentadas em (25.1) são multiplicadas à esquerda por $\text{diag}(G^T, I)$ e à direita por $\text{diag}(G, I)$, levando à

$$\begin{bmatrix} -G^T P_{\theta_k}^{-1} G & G^T A_{\theta_k}^T + G^T K_{\theta_k}^T B_{\theta_k}^T \\ * & -P_{\theta_{k+1}} \end{bmatrix} < 0$$

Utilizando o Lema A.1, com $\mu = 1$, sabemos que $-G^T P_{\theta_k}^{-1} G \leq -G^T - G + P_{\theta_k}$, e definindo $Y_j = K_j G$, temos

$$\begin{bmatrix} -G^T - G + P_{\theta_k} & G^T A_{\theta_k}^T + Y_{\theta_k}^T B_{\theta_k}^T \\ * & -P_{\theta_{k+1}} \end{bmatrix} < 0$$

que podemos escrever, com os somatórios nebulosos, como

$$\sum_{i=1}^r \sum_{j=1}^r \sum_{\ell=1}^r h_i(\theta_k) h_j(\theta_k) h_\ell(\theta_{k+1}) \begin{bmatrix} -G^T - G + P_i & G^T A_i^T + Y_j^T B_i^T \\ * & -P_\ell \end{bmatrix} < 0$$

de modo que condições suficientes para a síntese de leis de controle estabilizantes PDC, são dadas por

$$\begin{aligned} P_i &> 0 \quad \forall i \\ Q_{i\ell} &< 0 \quad \forall i, \ell \\ Q_{ij\ell} + Q_{jil} &< 0 \quad \forall i, j > i, \ell \end{aligned}$$

com

$$Q_{ij\ell} = \begin{bmatrix} -G^T - G + P_i & G^T A_i^T + Y_j^T B_i^T \\ * & -P_\ell \end{bmatrix}$$

e os ganhos do controlador são recuperados por meio de $K_j = Y_j G^{-1}$.

Lei de Controle não-PDC

Note que, utilizando a matriz G constante, fomos capazes de recuperar as matrizes da lei de controle PDC, e por isso chegamos em um conjunto de condições LMI de síntese. Entretanto, tais condições são, de certa forma, conservadoras, pois exigem que a matriz G seja a mesma para todas as regras.

De modo a contornar esta ideia, uma nova lei de controle, denominada de controle não-PDC, foi proposta em Guerra e Vermeiren 2004. Esta lei é definida como

$$\begin{aligned} \mathbf{u}_k &= \left(\sum_{i=1}^r h_i(\theta_k) K_i \right) \left(\sum_{j=1}^r h_j(\theta_k) G_j \right)^{-1} \mathbf{x}_k \\ \mathbf{u}_k &= K_{\theta_k} G_{\theta_k}^{-1} \mathbf{x}_k \end{aligned}$$

A principal diferença desta lei de controle para a lei de controle PDC é que ela **depende da inversa de um somatório com as funções de pertinência**. Apesar de qualquer lei de controle nebuloso com funções de pertinência diferentes no controlador daquelas utilizadas no sistema possa ser chamada de uma lei de controle não-PDC, é usual atribuir este nome à lei de controle apresentada.

Com esta lei de controle, o sistema em malha fechada passa a ser descrito por

$$\mathbf{x}_{k+1} = \left(A_{\theta_k} + B_{\theta_k} K_{\theta_k} G_{\theta_k}^{-1} \right) \mathbf{x}_k$$

e a partir da mesma função de Lyapunov que acabamos de utilizar, após um complemento de Schur, temos que as seguintes condições são suficientes para garantir a negatividade da variação da função de Lyapunov

$$\begin{bmatrix} -P_{\theta_k}^{-1} & A_{\theta_k}^T + G_{\theta_k}^{-T} K_{\theta_k}^T B_{\theta_k}^T \\ * & -P_{\theta_{k+1}} \end{bmatrix} < 0$$

Consideramos uma transformação de congruência em que estas condições LMI são multiplicadas à esquerda por $\text{diag}(G_{\theta_k}^T, I)$ e à direita por $\text{diag}(G_{\theta_k}, I)$, levando à

$$\begin{bmatrix} -G_{\theta_k}^T P_{\theta_k}^{-1} G_{\theta_k} & G_{\theta_k}^T A_{\theta_k}^T + K_{\theta_k}^T B_{\theta_k}^T \\ * & -P_{\theta_{k+1}} \end{bmatrix} < 0$$

Utilizando o Lema A.1, com $\mu = 1$, sabemos que $-G_{\theta_k}^T P_{\theta_k}^{-1} G_{\theta_k} \leq -G_{\theta_k}^T - G_{\theta_k} + P_{\theta_k}$ levando a

$$\begin{bmatrix} -G_{\theta_k}^T - G_{\theta_k} + P_{\theta_k} & G_{\theta_k}^T A_{\theta_k}^T + K_{\theta_k}^T B_{\theta_k}^T \\ * & -P_{\theta_{k+1}} \end{bmatrix} < 0$$

que podemos escrever, com os somatórios nebulosos, como

$$\sum_{i=1}^r \sum_{j=1}^r \sum_{\ell=1}^r h_i(\boldsymbol{\theta}_k) h_j(\boldsymbol{\theta}_k) h_\ell(\boldsymbol{\theta}_{k+1}) \begin{bmatrix} -G_i^T - G_i + P_i & G_j^T A_i^T + K_j^T B_i^T \\ * & -P_\ell \end{bmatrix} < 0$$

de modo que condições suficientes para a síntese de leis de controle estabilizantes PDC, são dadas por

$$\begin{aligned} P_i &> 0 \quad \forall i \\ Q_{i\ell} &< 0 \quad \forall i, \ell \\ Q_{ij\ell} + Q_{jil} &< 0 \quad \forall i, j > i, \ell \end{aligned}$$

com

$$Q_{ij\ell} = \begin{bmatrix} -G_i^T - G_i + P_i & G_j^T A_i^T + K_j^T B_i^T \\ * & -P_\ell \end{bmatrix}$$

Lei de Controle não-PDC com funções de pertinência atrasadas

Um conservadorismo que ainda existe em nossas condições é o fato de que as matrizes da função de Lyapunov no instante futuro, $P_{\theta_{k+1}}$, aparecem variando com um índice independente de todas as outras matrizes da condição. Deste modo, de forma a encontrar condições menos conservadoras, os autores em Lendek, Guerra e Lauber 2015 propuseram o uso de uma função de Lyapunov fuzzy com as funções de pertinência atrasadas

$$V(\mathbf{x}_k) = \mathbf{x}_k^T P_{\theta_{k-1}}^{-1} \mathbf{x}_k$$

de modo que sua variação é dada por

$$\Delta V = \mathbf{x}_{k+1}^T P_{\theta_k}^{-1} \mathbf{x}_{k+1} - \mathbf{x}_k^T P_{\theta_{k-1}}^{-1} \mathbf{x}_k$$

Note que ao fazer isto, a variação da função de Lyapunov passa a depender das funções de pertinência avaliadas no instante atual e no instante anterior. Com apenas esta mudança, o conservadorismo das condições seria equivalente às condições que obtivemos anteriormente. Entretanto, como a dependência que temos em outro instante de tempo das pertinências é de um valor passado, podemos definir nossa lei de controle como

$$\mathbf{u}_k = \left(\sum_{i=1}^r \sum_{j=1}^r h_i(\boldsymbol{\theta}_k) h_j(\boldsymbol{\theta}_{k-1}) K_{ij} \right) \left(\sum_{\kappa=1}^r \sum_{\ell=1}^r h_{\kappa}(\boldsymbol{\theta}_k) h_{\ell}(\boldsymbol{\theta}_{k-1}) G_{\kappa\ell} \right)^{-1} \mathbf{x}_k$$

$$\mathbf{u}_k = K_{\theta_k \theta_{k-1}} G_{\theta_k \theta_{k-1}}^{-1} \mathbf{x}_k$$

Com esta lei de controle, o sistema em malha fechada passa a ser descrito por

$$\mathbf{x}_{k+1} = \left(A_{\theta_k} + B_{\theta_k} K_{\theta_k \theta_{k-1}} G_{\theta_k \theta_{k-1}}^{-1} \right) \mathbf{x}_k$$

que, substituindo na variação da função de Lyapunov, e utilizando o complemento de Schur, leva ao seguinte conjunto de condições suficientes para garantir a negatividade da variação da função de Lyapunov

$$\begin{bmatrix} -P_{\theta_{k-1}}^{-1} & A_{\theta_k}^T + G_{\theta_k \theta_{k-1}}^{-T} K_{\theta_k \theta_{k-1}}^T B_{\theta_k}^T \\ * & -P_{\theta_k} \end{bmatrix} < 0$$

Consideramos uma transformação de congruência em que estas condições LMI são multiplicadas à esquerda por $\text{diag}(G_{\theta_k \theta_{k-1}}^T, I)$ e à direita por $\text{diag}(G_{\theta_k \theta_{k-1}}, I)$, levando à

$$\begin{bmatrix} -G_{\theta_k \theta_{k-1}}^T P_{\theta_{k-1}}^{-1} G_{\theta_k \theta_{k-1}} & G_{\theta_k \theta_{k-1}}^T A_{\theta_k}^T + K_{\theta_k \theta_{k-1}}^T B_{\theta_k}^T \\ * & -P_{\theta_k} \end{bmatrix} < 0$$

Assim como fizemos anteriormente, utilizamos o fato de que

$$-G_{\theta_k \theta_{k-1}}^T P_{\theta_{k-1}}^{-1} G_{\theta_k \theta_{k-1}} \leq -G_{\theta_k \theta_{k-1}}^T - G_{\theta_k \theta_{k-1}} + P_{\theta_{k-1}}$$

levando a

$$\begin{bmatrix} -G_{\theta_k \theta_{k-1}}^T - G_{\theta_k \theta_{k-1}} + P_{\theta_{k-1}} & G_{\theta_k \theta_{k-1}}^T A_{\theta_k}^T + K_{\theta_k \theta_{k-1}}^T B_{\theta_k}^T \\ * & -P_{\theta_k} \end{bmatrix} < 0$$

Podemos escrever esta condição, com os somatórios nebulosos, como

$$\sum_{i=1}^r \sum_{j=1}^r \sum_{\ell=1}^r h_i(\theta_k) h_j(\theta_k) h_\ell(\theta_{k-1}) \begin{bmatrix} -G_{i\ell}^T - G_{i\ell} + P_\ell & G_{j\ell}^T A_i^T + K_{j\ell}^T B_i^T \\ * & -P_i \end{bmatrix} < 0$$

de modo que condições suficientes para a síntese de leis de controle estabilizantes PDC, são dadas por

$$\begin{aligned} P_i &> 0 \quad \forall i \\ Q_{i\ell} &< 0 \quad \forall i, \ell \\ Q_{ij\ell} + Q_{jil} &< 0 \quad \forall i, j > i, \ell \end{aligned}$$

com

$$Q_{ij\ell} = \begin{bmatrix} -G_{i\ell}^T - G_{i\ell} + P_\ell & G_{j\ell}^T A_i^T + K_{j\ell}^T B_i^T \\ * & -P_i \end{bmatrix}$$

Sistemas a tempo contínuo

No caso dos sistemas a tempo contínuo, consideramos um sistema descrito por

$$\dot{\mathbf{x}} = A_\theta \mathbf{x} + B_\theta \mathbf{u}$$

e, inicialmente, vamos considerar uma função de Lyapunov fuzzy candidata da forma

$$V(\mathbf{x}) = \mathbf{x}^T P_\theta \mathbf{x}$$

Com uma função dessa forma, sua derivada temporal será

$$\dot{V} = \dot{\mathbf{x}}^T P_\theta \mathbf{x} + \mathbf{x}^T P_\theta \dot{\mathbf{x}} + \mathbf{x}^T \dot{P}_\theta \mathbf{x}$$

e precisamos de lidar com o termo $\mathbf{x}^T \dot{P}_\theta \mathbf{x}$ de alguma forma em nossas condições.

Existem, na literatura, diferentes formas de se lidar com essa derivada temporal das funções de pertinência, mas nesta aula vamos focar nossa atenção na estratégia proposta em Mozelli et al. 2009. Note que a propriedade de que a soma das funções de pertinência é sempre igual a 1, implica em

$$\sum_{i=1}^r h_i(\theta) = 1 \Rightarrow \sum_{i=1}^r \dot{h}_i(\theta) = 0$$

de modo que

$$\sum_{i=1}^r \dot{h}_i X = \left(\sum_{i=1}^r \dot{h}_i \right) X = 0$$

o que nos permitirá inserir variáveis de folga em nossas condições para lidar com as derivadas temporais das funções de pertinência. Dessa forma, podemos escrever

$$\dot{P}_\theta = \sum_{i=1}^r \dot{h}_i P_i = \sum_{i=1}^r \dot{h}_i (P_i + X)$$

Considerando que as derivadas temporais das funções de pertinência atendem aos limites

$$|\dot{h}_i(\theta)| \leq \phi_i$$

se exigirmos que

$$P_i + X \geq 0$$

podemos escrever que

$$\dot{P}_\theta \leq \sum_{i=1}^r |\dot{h}_i| (P_i + X) \leq \sum_{i=1}^r \phi_i (P_i + X)$$

E portanto, uma forma de lidar com o termo da derivada temporal das funções de pertinência é dado por

$$\begin{aligned} \dot{P}_\theta &\leq P_\phi \\ P_\phi &= \sum_{i=1}^r \phi_i (P_i + X) \end{aligned}$$

Agora que sabemos como podemos lidar com este termo, vamos primeiro considerar o caso em que as variáveis premissas do nosso modelo são dadas por sinais externos, que não dependem da dinâmica do sistema.

Variáveis premissas são sinais externos - Controle PDC

Neste caso, já possuímos as ferramentas que precisamos para encontrar as condições de síntese. Vamos considerar uma lei de controle PDC da forma

$$\mathbf{u} = K_\theta \mathbf{x}$$

levando a uma dinâmica de malha fechada da forma

$$\dot{\mathbf{x}} = (A_\theta + B_\theta K_\theta) \mathbf{x}$$

Se substituirmos essa dinâmica diretamente na derivada temporal da função de Lyapunov, obtemos

$$\mathbf{x}^T \left((A_\theta + B_\theta K_\theta)^T P_\theta + P_\theta (A_\theta + B_\theta K_\theta) + \dot{P}_\theta \right) \mathbf{x} < 0$$

Entretanto, como nosso objetivo é encontrar uma lei de controle PDC, encontramos um problema, pois não é fácil encontrar uma transformação de congruência que nos permita lidar com o produto entre os ganhos da lei de controle e da matriz de parâmetros da função de Lyapunov. Desta forma, é tradicional utilizar alguma estratégia para “separar a matriz P_θ das matrizes da dinâmica do sistema”.

Existem várias estratégias diferentes disponíveis na literatura para realizar isto, como as estratégias com *Termo Nulo* (Mozelli, Palhares e Avellar 2009), *Lema de Finsler* ou *Sistemas Descritores* (Mozelli, Palhares e Mendes 2010), ou mesmo *Condições similares à de sistemas discretos para sistemas contínuos* (Shaked 2001).

Nesta aula, focaremos nossa atenção na abordagem utilizando o Lema de Finsler (Lema A.6), cuja prova está apresentada no Apêndice A. Voltando à condição para a derivada temporal da função de Lyapunov candidata, podemos escrevê-la como

$$\dot{V} = \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{bmatrix}^T \begin{bmatrix} \dot{P}_\theta & P_\theta \\ * & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{bmatrix} < 0$$

e sabemos que o vetor da forma quadrática possui a relação

$$\begin{bmatrix} A_\theta + B_\theta K_\theta & -I \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{bmatrix} = 0$$

Note que a condição desejada para a derivada temporal tem a forma

$$\mathbf{x}^T Q \mathbf{x} < 0, \quad \forall \mathbf{x} | B\mathbf{x} = 0, \mathbf{x} \neq 0$$

da afirmação 1. do Lema de Finsler (Lema A.6), com

$$Q = \begin{bmatrix} \dot{P}_\theta & P_\theta \\ * & 0 \end{bmatrix}, \quad B = \begin{bmatrix} A_\theta + B_\theta K_\theta & -I \end{bmatrix}$$

de modo que ela é equivalente à condição 4. do Lema, que tem a forma

$$\exists X | Q + XB + B^T X^T < 0$$

Para que as condições que encontremos sejam tratáveis, iremos abrir mão na equivalência entre as condições e iremos escolher uma matriz X com uma estrutura específica (garantindo ainda a suficiência da condição que iremos obter). Escolhemos

$$X = \begin{bmatrix} N^{-T} \\ \mu N^{-T} \end{bmatrix}$$

de modo que nossa condição suficiente para que a derivada temporal da função de Lyapunov seja negativa é dada por

$$\begin{bmatrix} \dot{P}_\theta & P_\theta \\ * & 0 \end{bmatrix} + \begin{bmatrix} N^{-T} \\ \mu N^{-T} \end{bmatrix} \begin{bmatrix} A_\theta + B_\theta K_\theta & -I \end{bmatrix} + \begin{bmatrix} A_\theta^T + K_\theta^T B_\theta^T \\ -I \end{bmatrix} \begin{bmatrix} N^{-1} & \mu N^{-1} \end{bmatrix} < 0$$

$$\begin{bmatrix} \dot{P}_\theta + N^{-T} (A_\theta + B_\theta K_\theta) + (A_\theta^T + K_\theta^T B_\theta^T) N^{-1} & P_\theta - N^{-T} + \mu (A_\theta^T + K_\theta^T B_\theta^T) N^{-1} \\ * & -\mu (N^{-T} + N^{-1}) \end{bmatrix} < 0$$

Fazendo uma transformação de congruência, multiplicando nossa condição à esquerda por $\text{diag}(N^T, I)$ e à direita por $\text{diag}(N, I)$, e definindo $W_\theta = N^T P_\theta N$ e $Y_\theta = K_\theta N$, chegamos na condição

$$\begin{bmatrix} \dot{W}_\theta + A_\theta N + N^T A_\theta^T + B_\theta Y_\theta + Y_\theta^T B_\theta^T & W_\theta - N + \mu (N^T A_\theta^T + Y_\theta^T B_\theta^T) \\ * & -\mu (N^T + N) \end{bmatrix} < 0$$

Definindo $W_\phi = \sum_{i=1}^r \phi_i (W_i + X)$, e exigindo que $W_i + X \geq 0$, temos a seguinte condição suficiente

$$\begin{bmatrix} W_\phi + A_\theta N + N^T A_\theta^T + B_\theta Y_\theta + Y_\theta^T B_\theta^T & W_\theta - N + \mu (N^T A_\theta^T + Y_\theta^T B_\theta^T) \\ * & -\mu (N^T + N) \end{bmatrix} < 0$$

De modo que, dado um escalar μ e os limitantes ϕ_i , tal que $|\dot{h}_i| \leq \phi_i$, temos as seguintes condições suficientes de síntese para uma lei de controle PDC

$$\begin{aligned} W_i &> 0, \forall i \\ W_i + X &\geq 0, \forall i, \\ Q_{ii} &< 0, \forall i \\ Q_{ij} + Q_{ji} &< 0 \forall i, j > i \end{aligned}$$

com

$$Q_{ij} = \begin{bmatrix} W_\phi + A_i N + N^T A_i^T + B_i Y_j + Y_j^T B_i^T & W_i - N + \mu (N^T A_i^T + Y_j^T B_i^T) \\ * & -\mu (N^T + N) \end{bmatrix}$$

$$W_\phi = \sum_{i=1}^r \phi_i (W_i + X)$$

e os ganhos de controle são encontrados por $K_i = Y_i N^{-1}$.

Variáveis premissas dependem dos estados - Controle PDC

No caso em que as variáveis premissas dependem dos estados, a dinâmica do sistema em malha fechada afeta o limitante utilizado para a derivada temporal das funções de pertinência, de modo que, se não utilizarmos algumas condições extras para garantir que os limitantes escolhidos para as funções de pertinência são respeitados dentro de uma certa região, podemos acabar com um controlador cuja validade só é garantida em uma região bem pequena em torno da origem. Entretanto, de forma a lidar com estes limitantes, abrimos mão de garantirmos uma propriedade global para o sistema, para garantirmos uma propriedade local.

Nesta aula, vamos focar nossa atenção na estratégia proposta por (Lee, Park e Joo 2012). A ideia é que garantimos que o limitante escolhido é respeitado dentro do conjunto de nível

$$V(\mathbf{x}) \leq 1$$

e condições extras são adicionadas para tentar crescer este conjunto o máximo possível, dentro de uma região de análise estabelecida.

Considere os limites desejados para as derivadas temporais da função de Lyapunov

$$\begin{aligned} |\dot{h}_i| &\leq \phi_i \\ \dot{h}_i^2 &\leq \phi_i^2 \\ \frac{\dot{h}_i^2}{\phi_i^2} &\leq 1 \end{aligned}$$

de modo que, se garantirmos que

$$\frac{\dot{h}_i^2}{\phi_i^2} \leq V(\mathbf{x})$$

sabemos que os limitantes escolhidos serão respeitados quando $V(\mathbf{x}) \leq 1$ (o conjunto de nível desejado).

Como as variáveis premissas, neste caso, são funções do estado, podemos escrever que

$$\dot{h}_i = \nabla h_i \dot{\mathbf{x}}$$

em que ∇h_i é o gradiente da função de pertinência h_i em relação às variáveis de estado. Dessa forma, nossa condição desejada pode ser escrita, para este caso, como

$$-\mathbf{x}^T P_\theta \mathbf{x} + \frac{1}{\phi_i^2} \mathbf{x}^T \left(A_\theta^T + K_\theta^T B_\theta^T \right) \nabla h_i^T \nabla h_i (A_\theta + B_\theta K_\theta) \mathbf{x} \leq 0$$

que, com uma transformação de congruência do tipo $\mathbf{x} = N\mathbf{w}$, leva às condições

$$-W_\theta + \frac{1}{\phi_i^2} \left(N^T A_\theta^T + Y_\theta^T B_\theta^T \right) \nabla h_i^T \nabla h_i (A_\theta N + B_\theta Y_\theta) \leq 0$$

e, finalmente, por meio de um complemento de Schur, leva a

$$\begin{bmatrix} -W_\theta & (N^T A_\theta^T + Y_\theta^T B_\theta^T) \nabla h_i^T \\ * & -\phi_i^2 \end{bmatrix} \leq 0$$

Note que a condição que chegamos depende do gradiente das funções de pertinência, e, se considerarmos uma representação politópica para o gradiente, podemos substituí-lo pelos vértices de sua representação.

Como estamos lidando com uma análise local agora, o conjunto de nível unitário da função de Lyapunov servirá como uma estimativa do domínio de atração do sistema em malha fechada (e pelas condições que acabamos de derivar, apenas garantimos que a nossa análise é válida dentro desta região). Desta forma, é importante adicionar condições para garantir que este conjunto esteja contido dentro da região de análise. A ideia neste caso normalmente é impor que nas bordas da região, o valor da função de Lyapunov seja maior ou igual a 1 (de modo que, por continuidade da função, e o fato de que na origem ela vale 0, o conjunto desejado estará contido na região).

Considerando que nossa região de análise, Ω , é definida por um hiperretângulo, da forma

$$\begin{aligned} \Omega_j &= \{\mathbf{x} \in \mathbb{R}^n \mid |x_j| \leq \bar{x}_j\} \\ \Omega &= \bigcap_{j=1}^n \Omega_j \end{aligned}$$

se garantirmos que

$$V(\mathbf{x}) \geq \frac{x_j^2}{\bar{x}_j^2}$$

teremos que $V(\mathbf{x}) \geq 1$ quando o estado estiver na borda da região de análise. De modo a escrevermos esta condição, definimos os vetores indicadores \mathbf{e}_j^T dados pela j -ésima linha da matriz identidade $n \times n$, de modo que $x_j = \mathbf{e}_j^T \mathbf{x}$. Sendo assim, nossa condição pode ser escrita como

$$\begin{aligned} -P_\theta + \frac{1}{\bar{x}_j^2} \mathbf{e}_j \mathbf{e}_j^T &\leq 0 \\ \begin{bmatrix} -P_\theta & \mathbf{e}_j \\ * & -\bar{x}_j^2 \end{bmatrix} &\leq 0 \end{aligned}$$

Fazendo uma transformação de congruência, multiplicando à esquerda por $\text{diag}(N^T, 1)$, e à direita $\text{diag}(N, 1)$, chegamos nas condições

$$\begin{bmatrix} -W_\theta & N^T \mathbf{e}_j \\ * & -\bar{x}_j^2 \end{bmatrix} \leq 0$$

Já que já garantimos que a estimativa do domínio de atração está contida na região de análise, basta que adicionemos alguma condição de modo a crescer esta região. Uma forma muito comum de se fazer isso, é tentarmos crescer uma bola contida dentro da região.

Considerando uma bola definida por

$$\begin{aligned} \mathbf{x}^T \mathbf{x} &\leq \beta \\ \frac{1}{\beta} \mathbf{x}^T \mathbf{x} &\leq 1 \end{aligned}$$

queremos que este conjunto esteja dentro do conjunto de nível unitário da função de Lyapunov. De modo que queremos que

$$V(\mathbf{x}) - 1 \leq \frac{1}{\beta} \mathbf{x}^T \mathbf{x} - 1$$

pois isto implica que $V(\mathbf{x}) \leq 1$ sempre que $\mathbf{x}^T \mathbf{x} \leq 1$. O que nos leva à condição

$$P_\theta \leq \frac{1}{\beta} I$$

que com uma transformação de congruência, multiplicando por N^T à esquerda, e N à direita, temos

$$W_\theta \leq \frac{1}{\beta} N^T N$$

Infelizmente, neste caso em que estamos aplicando estas condições, não exista uma forma direta de impormos esta condição sem adicionarmos um certo conservadorismo. De modo que a condição fique tratável, consideramos a desigualdade $-\frac{1}{\beta} N^T N \leq -N^T - N + \beta I$. De modo que uma condição *suficiente* para tentarmos crescer a estimativa do domínio de atração é **maximizando** β e adicionando a condição

$$W_\theta - N^T - N + \beta I \leq 0$$

Juntando tudo, vemos que, além das condições utilizadas para o caso das premissas independentes dos estados, precisamos das seguintes condições extras

$$\begin{aligned} \begin{bmatrix} -W_\theta & (N^T A_\theta^T + Y_\theta^T B_\theta^T) \nabla h_i^T \\ * & -\phi_i^2 \end{bmatrix} &\leq 0 \\ \begin{bmatrix} -W_\theta & N^T \mathbf{e}_j \\ * & -\bar{x}_j^2 \end{bmatrix} &\leq 0 \\ W_\theta - N^T - N + \beta I &\leq 0 \end{aligned}$$

Controle Não-PDC

Para finalizar esta aula, vamos considerar o caso em que temos uma lei de controle não-PDC.

Para isso, vamos considerar uma função de Lyapunov candidata da forma

$$V(\mathbf{x}) = \mathbf{x}^T P_\theta^{-1} \mathbf{x}$$

e a propriedade de que $\dot{P}_\theta^{-1} = -P_\theta^{-1} \dot{P}_\theta P_\theta^{-1}$. Esta propriedade é válida pois

$$\begin{aligned} P_\theta P_\theta^{-1} &= I \\ \dot{P}_\theta P_\theta^{-1} + P_\theta \dot{P}_\theta^{-1} &= 0 \\ \dot{P}_\theta^{-1} &= -P_\theta^{-1} \dot{P}_\theta P_\theta^{-1} \end{aligned}$$

A derivada temporal da função de Lyapunov candidata é dada por

$$\dot{V} = \dot{\mathbf{x}}^T P_\theta^{-1} \mathbf{x} + \mathbf{x}^T P_\theta^{-1} \dot{\mathbf{x}} + \mathbf{x}^T \dot{P}_\theta^{-1} \mathbf{x} = \dot{\mathbf{x}}^T P_\theta^{-1} \mathbf{x} + \mathbf{x}^T P_\theta^{-1} \dot{\mathbf{x}} - \mathbf{x}^T P_\theta^{-1} \dot{P}_\theta P_\theta^{-1} \mathbf{x}$$

Considerando uma lei de controle da forma

$$\mathbf{u} = K_\theta P_\theta^{-1} \mathbf{x}$$

a dinâmica do sistema em malha fechada é dada por

$$\dot{\mathbf{x}} = \left(A_\theta + B_\theta K_\theta P_\theta^{-1} \right) \mathbf{x}$$

substituindo na derivada temporal do sistema, temos que condições suficientes são dadas por

$$\left(A_\theta + B_\theta K_\theta P_\theta^{-1} \right)^T P_\theta^{-1} + P_\theta^{-1} \left(A_\theta + B_\theta K_\theta P_\theta^{-1} \right) - P_\theta^{-1} \dot{P}_\theta P_\theta^{-1} < 0$$

Fazendo uma transformação de congruência, multiplicando a condição à esquerda e à direita por P_θ , obtemos

$$A_\theta P_\theta + P_\theta A_\theta^T + B_\theta K_\theta + K_\theta^T B_\theta^T - \dot{P}_\theta < 0$$

Considerando um conjunto de limites para as derivadas temporais das funções de pertinência, dados, da forma $|\dot{h}_i| < \phi_i$, como vimos anteriormente, podemos definir

$$\begin{aligned} P_i + X &\geq 0 \\ P_\phi &= \sum_{i=1}^r \phi_i (P_i + X) \\ -\dot{P}_\theta &\leq P_\phi \end{aligned}$$

de modo que, para os sistemas em que as variáveis premissas não dependem dos estados, temos que condições suficientes para a síntese de uma lei de controle não-PDC são dadas por

$$\begin{aligned} P_\theta &> 0 \\ P_i + X &\geq 0, \forall i \\ A_\theta P_\theta + P_\theta A_\theta^T + B_\theta K_\theta + K_\theta^T B_\theta^T + P_\phi &< 0 \end{aligned}$$

Seguindo os mesmos passos realizados na seção anterior, para os sistemas cujas variáveis premissas dependem dos estados, iremos impor que $\frac{h_i^2}{\phi^2} \leq V(\mathbf{x})$, que $V(\mathbf{x}) \geq 1$ nas bordas da região de análise, e um conjunto de condições para impor que uma bola (que tentamos crescer) está contida dentro do conjunto de nível unitário da função de Lyapunov.

Para $\frac{h_i^2}{\phi^2} \leq V(\mathbf{x})$, temos

$$\begin{aligned} -P_\theta^{-1} + \frac{1}{\phi_i^2} (A_\theta^T + P_\theta^{-1} K_\theta^T B_\theta^T) \nabla h_i^T \nabla h_i (A_\theta + B_\theta K_\theta P_\theta^{-1}) &\leq 0 \\ \begin{bmatrix} -P_\theta^{-1} & (A_\theta^T + P_\theta^{-1} K_\theta^T B_\theta^T) \nabla h_i^T \\ * & -\phi_i^2 \end{bmatrix} &\leq 0 \\ \begin{bmatrix} -P_\theta & (P_\theta A_\theta^T + K_\theta^T B_\theta^T) \nabla h_i^T \\ * & -\phi_i^2 \end{bmatrix} &\leq 0 \end{aligned}$$

em que a última condição é obtida por meio de uma transformação de congruência.

Para garantir que $V(\mathbf{x}) \geq 1$ nas bordas da região de análise, fazemos

$$\begin{aligned} -P_\theta^{-1} + \frac{1}{\bar{x}_j^2} \mathbf{e}_j \mathbf{e}_j^T &\leq 0 \\ \begin{bmatrix} -P_\theta^{-1} & \mathbf{e}_j \\ * & -\bar{x}_j^2 \end{bmatrix} &\leq 0 \\ \begin{bmatrix} -P_\theta & P_\theta \mathbf{e}_j \\ * & -\bar{x}_j^2 \end{bmatrix} &\leq 0 \end{aligned}$$

em que a última condição é obtida por meio de uma transformação de congruência.

De modo a impormos que uma bola está contida dentro do conjunto de nível unitário da função de Lyapunov (que tentamos crescer), fazemos

$$V(\mathbf{x}) - 1 \leq \frac{1}{\beta} \mathbf{x}^T \mathbf{x} - 1$$

$$V(\mathbf{x}) \leq \frac{1}{\beta} \mathbf{x}^T \mathbf{x}$$

$$P_\theta^{-1} \leq \frac{1}{\beta} I$$

$$P_\theta \geq \beta I$$

e tentamos **maximizar** o valor de β .

Juntando as três condições extras, temos que, para os sistemas cujas variáveis premissas dependem dos estados do sistema, as seguintes condições extras são suficientes para garantir que o limite escolhido para as derivadas das funções de pertinência é respeitado dentro do conjunto de nível unitário da função de Lyapunov, e este conjunto é aumentado, respeitando os limites da região de análise, se **maximizarmos** β e considerarmos

$$\begin{bmatrix} -P_\theta & (P_\theta A_\theta^T + K_\theta^T B_\theta^T) \nabla h_i^T \\ * & -\phi_i^2 \end{bmatrix} \leq 0$$

$$\begin{bmatrix} -P_\theta & P_\theta \mathbf{e}_j \\ * & -\bar{x}_j^2 \end{bmatrix} \leq 0$$

$$P_\theta \geq \beta I$$

Unidade IV

**Controle Adaptativo Baseado em
Aproximação**

Aula 26

Controle Adaptativo Baseado em Aproximação

Nesta parte da matéria, vamos focar nossos estudos em como podemos utilizar sistemas não-lineares como aproximadores universais para modelar dinâmicas desconhecidas em um sistema não-linear de forma adaptativa.

Nesta aula, vamos ver como o controle adaptativo poderia ser aplicado em um sistema escalar com parâmetros desconhecidos, e em seguida vamos estender a abordagem para utilizar um aproximador universal e aplicar em sistemas com dinâmica desconhecida.

Como vimos anteriormente, uma técnica muito comum para o controle de sistemas não-lineares é a linearização por realimentação de estados.

Como exemplo, considere o sistema escalar (com um único estado)

$$\dot{x} = f(x) + g(x)u$$

A lei de controle

$$u = \frac{-f(x) - ax}{g(x)}$$

faz com que a dinâmica do sistema em malha fechada seja

$$\dot{x} = -ax$$

e o sistema será assintoticamente estável em malha fechada (vale a pena observar que tal lei de controle só é válida nos pontos em que $g(x) \neq 0$).

Note que essa lei só é possível quando a dinâmica do sistema é perfeitamente conhecida (já que ela cancela a dinâmica e a substitui pela dinâmica desejada).

Controle adaptativo

Quando conhecemos a dinâmica, mas desconhecemos alguns parâmetros, podemos fazer uso de técnicas de controle adaptativo para sistemas não-lineares.

Como exemplo, considere o sistema escalar (com um único estado) e um parâmetro desconhecido θ

$$\dot{x} = \theta f(x) + g(x)u$$

Usando a lei de controle

$$u = \frac{-\hat{\theta}f(x) - ax}{g(x)}$$

temos que a malha fechada será dada por

$$\dot{x} = -ax + (\theta - \hat{\theta})f(x)$$

Definindo o erro de adaptação

$$\tilde{\theta} = \theta - \hat{\theta}$$

podemos reescrever a dinâmica da malha fechada como

$$\dot{x} = -ax + \tilde{\theta}f(x)$$

Fazendo uso da função de Lyapunov

$$V(x, \tilde{\theta}) = \frac{1}{2}x^2 + \frac{1}{2\gamma}\tilde{\theta}^2$$

sua derivada temporal é dada por

$$\begin{aligned}\dot{V}(x, \tilde{\theta}) &= x\dot{x} - \frac{1}{\gamma}\tilde{\theta}\dot{\tilde{\theta}}, \\ &= -ax^2 + x\tilde{\theta}f(x) + \frac{1}{\gamma}\tilde{\theta}\dot{\tilde{\theta}}\end{aligned}$$

Escolhendo a lei de adaptação

$$\dot{\tilde{\theta}} = \gamma x f(x)$$

a derivada temporal da função de Lyapunov será dada por

$$\dot{V}(x, \tilde{\theta}) = -ax^2$$

Como a derivada temporal é semidefinida negativa, provamos que o sistema é estável em malha fechada. Definindo

$$\begin{aligned}\varphi(t) &= \dot{V}(x, \tilde{\theta}) = -ax^2 \\ \dot{\varphi} &= -2ax(-ax + \tilde{\theta}f(x)) = 2a^2x^2 - 2ax\tilde{\theta}f(x)\end{aligned}$$

Supondo que $f(x)$ é limitado, e como o sistema é estável em malha fechada, $\dot{\varphi}$ é limitado, e φ é uniformemente contínuo. Note que

$$\lim_{t \rightarrow \infty} \int_0^t \varphi(s) ds = V(\infty) - V(0)$$

Como $V(x, \tilde{\theta})$ é limitado abaixo, e $\dot{V}(x, \tilde{\theta}) \leq 0$, $V(\infty)$ existe e é limitado. Logo, pelo Lema de Barbalat, $\varphi(t) \rightarrow 0$, e com isso $x \rightarrow 0$ (provando que os estados do sistema convergem para a origem do espaço de estados em malha fechada).

Controle adaptativo utilizando aproximadores TS

Um problema um pouco diferente deste é quando desconhecemos a dinâmica que descreve o comportamento do sistema, ao invés de desconhecermos apenas alguns parâmetros. Neste caso, não é possível se aplicar as técnicas de controle adaptativo para sistemas não-lineares. Uma solução possível neste caso, é fazer uso de aproximadores universais na lei de controle para estimar a arte desconhecida da dinâmica. Esta é a ideia desta unidade.

Conforme vimos anteriormente, modelos Takagi-Sugeno (TS) possuem esta propriedade de aproximadores universais. Além disso, fixando-se as funções de pertinência do modelo, temos uma representação linear nos parâmetros.

Lembrando das aulas anteriores, sabemos que modelos TS normalmente são representados por regras da forma

Regra k : Se x_1 é A_1^k e x_2 é A_2^k e ... e x_n é A_n^k , então $\mathbf{y} = \mathbf{f}_k(x_1, x_2, \dots, x_n, \theta)$

Com funções de pertinência $\mu_{ki}(x_k)$ associadas à cada conjunto M_{ki} , podemos definir as funções de pertinência associadas à cada regra como

$$\mu_i(\mathbf{x}) = \prod_{k=1}^n \mu_{ki}(x_k),$$

e as funções de pertinências normalizadas de cada regra como

$$h_i(\mathbf{x}) = \frac{\mu_i(\mathbf{x})}{\sum_{j=1}^r \mu_j(\mathbf{x})}.$$

Tais funções respeitam as propriedades (partição de unidade)

$$h_i(\mathbf{x}) \geq 0, \quad \sum_{i=1}^r h_i(\mathbf{x}) = 1,$$

e a partir delas podemos fazer a inferência do modelo como

$$\mathbf{f}_h(\mathbf{x}) = \sum_{i=1}^r h_i(\mathbf{x}) \mathbf{f}_i(x_1, x_2, \dots, x_n, \theta).$$

Dessa forma, podemos interpretar um modelo TS como a combinação de várias funções f_i com pesos h_i para gerar uma função desejada.

Considerando que estamos aproximando uma função com “uma única saída” ($f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$), e que as funções utilizadas no conseqüente, $f_i(x)$, são lineares nos parâmetros, podemos escrever

$$\begin{aligned} f_h(\mathbf{x}) &= \sum_{i=1}^r h_i(\mathbf{x}) \boldsymbol{\eta}_i(\mathbf{x})^T \boldsymbol{\theta}_i, \\ &= \begin{bmatrix} h_1 \boldsymbol{\eta}_1^T & h_2 \boldsymbol{\eta}_2^T & \dots & h_r \boldsymbol{\eta}_r^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_1 \\ \boldsymbol{\theta}_2 \\ \vdots \\ \boldsymbol{\theta}_r \end{bmatrix}, \\ &= \bar{\mathbf{h}}^T(\mathbf{x}) \boldsymbol{\theta} \end{aligned}$$

É muito comum o caso em que as funções $f_i(x)$ são constantes, e para elas podemos escrever

$$\begin{aligned} f_h(\mathbf{x}) &= \sum_{i=1}^r h_i(\mathbf{x}) \theta_i, \\ &= \begin{bmatrix} h_1 & h_2 & \dots & h_r \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_r \end{bmatrix}, \\ &= \bar{\mathbf{h}}^T(\mathbf{x}) \boldsymbol{\theta} \end{aligned}$$

Voltando para o controle adaptativo, considerando um sistema escalar (um único estado) com dinâmica

$$\dot{x} = f_0(x) + f(x) + g(x)u$$

em que $f_0(x)$ é conhecida, e $f(x)$ é desconhecida. A ideia aqui é aproximar

$$f(x) \approx \bar{\mathbf{h}}^T(x) \boldsymbol{\theta}$$

e utilizar um lei de adaptação em malha fechada para estimar os parâmetros $\boldsymbol{\theta}$ do aproximador. Ignorando, na aula de hoje, o erro de aproximação entre a função desconhecida e o aproximador utilizado (porque mesmo com os melhores parâmetros possíveis, na maioria dos casos, existiria algum erro), podemos usar a lei de controle

$$u = \frac{-f_0(x) - \bar{\mathbf{h}}^T(x) \hat{\boldsymbol{\theta}} - ax}{g(x)}.$$

A malha fechada (desconsiderando o erro de aproximação) pode ser escrita como

$$\dot{x} = -ax + \bar{\mathbf{h}}^T(x) (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) = -ax + \bar{\mathbf{h}}^T(x) \tilde{\boldsymbol{\theta}}$$

Fazendo uso da função de Lyapunov

$$\begin{aligned} V(x, \tilde{\theta}) &= \frac{1}{2}x^2 + \frac{1}{2}\tilde{\theta}^T \Gamma^{-1} \tilde{\theta}, \\ \dot{V} &= x\dot{x} + \tilde{\theta}^T \Gamma^{-1} \dot{\tilde{\theta}}, \\ &= -ax^2 + \tilde{\theta}^T \left(-\Gamma^{-1} \dot{\tilde{\theta}} + \bar{\mathbf{h}}(x)x \right) \end{aligned}$$

usando a lei de adaptação

$$\dot{\tilde{\theta}} = \Gamma \bar{\mathbf{h}}(x)x$$

a derivada temporal da função de Lyapunov se reduz a

$$\dot{V} = -ax^2$$

e usando um desenvolvimento bem próximo ao da lei de controle adaptativo para sistemas não-lineares, conseguimos mostrar que o sistema é assintoticamente estável usando o Lema de Barbalat.

Aula 27

Controle Adaptativo por Aproximação - Lidando com o erro de aproximação e seguimento de trajetória

Na última aula vimos como, utilizando linearização por realimentação de estados, podemos estabilizar um sistema em torno da origem do espaço de estados.

Vimos também que, quando desconhecemos alguns parâmetros da dinâmica do sistema, podemos utilizar uma lei de controle adaptativa para compensar por este desconhecimento.

Por último, vimos que, quando desconhecemos a dinâmica do sistema ao invés de alguns parâmetros, podemos utilizar aproximadores universais (no nosso caso modelos TS) para compensar por este desconhecimento. Entretanto, consideramos que a dinâmica incerta pode ser perfeitamente representada pelo aproximador escolhido (o que na vida real é difícil de ser verdade).

Lidando com o erro de aproximação

Como na aula anterior, considere um sistema escalar (um único estado) com dinâmica

$$\dot{x} = f_0(x) + f(x) + g(x)u$$

em que $f_0(x)$ é conhecida, e $f(x)$ é desconhecida. Entretanto, vamos considerar que a função desconhecida $f(x)$ pode ser escrita como

$$f(x) = \bar{\mathbf{h}}^T(x)\boldsymbol{\theta}^* + \varepsilon(x)$$

em que consideramos o erro de aproximação $\varepsilon(x)$, dado por

$$\varepsilon(x) = f(x) - \bar{\mathbf{h}}^T(x)\boldsymbol{\theta}^*$$

com $\boldsymbol{\theta}^*$ o parâmetro que resulta no menor erro máximo de representação da função. Ou seja,

$$\boldsymbol{\theta}^* = \arg \min \left(\max_x |f(x) - \bar{\mathbf{h}}^T(x)\boldsymbol{\theta}| \right)$$

Com essa representação, sabemos que

$$|\varepsilon(x)| \leq \max_x |f(x) - \bar{\mathbf{h}}^T(x)\boldsymbol{\theta}| = e_f$$

Usando a mesma lei de controle da aula passada

$$u = \frac{-f_0(x) - \bar{\mathbf{h}}^T(x)\hat{\boldsymbol{\theta}} - ax}{g(x)}$$

e substituindo na equação do sistema, a dinâmica do sistema pode ser escrita como

$$\begin{aligned} \dot{x} &= -ax + \bar{\mathbf{h}}^T(x) (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) + \varepsilon(x) \\ &= -ax + \bar{\mathbf{h}}^T(x)\tilde{\boldsymbol{\theta}} + \varepsilon(x) \end{aligned}$$

Fazendo uso da mesma função de Lyapunov da aula passada

$$\begin{aligned} V(x, \tilde{\boldsymbol{\theta}}) &= \frac{1}{2}x^2 + \frac{1}{2}\tilde{\boldsymbol{\theta}}^T\Gamma^{-1}\tilde{\boldsymbol{\theta}} \\ \dot{V} &= x\dot{x} - \tilde{\boldsymbol{\theta}}^T\Gamma^{-1}\dot{\tilde{\boldsymbol{\theta}}} \\ \dot{V} &= -ax^2 + \varepsilon(x)x + \tilde{\boldsymbol{\theta}}^T \left(-\Gamma^{-1}\dot{\tilde{\boldsymbol{\theta}}} + \mathbf{h}(x)x \right) \end{aligned}$$

Usando a lei de adaptação

$$\dot{\tilde{\boldsymbol{\theta}}} = \Gamma\mathbf{h}(x)x$$

a derivada temporal da função de Lyapunov se reduz a

$$\begin{aligned} \dot{V} &= -ax^2 + \varepsilon(x)x \\ \dot{V} &\leq -ax^2 + e_f|x| \end{aligned}$$

Note que, que quando levamos o erro de aproximação em consideração não chegamos mais em uma derivada temporal semidefinida negativa para a função de Lyapunov escolhida, e não podemos mais concluir que o sistema é estável (no sentido de Lyapunov) por esta função de Lyapunov.

Uma condição *suficiente* para que a derivada da função de Lyapunov seja negativa é

$$\begin{aligned} -ax^2 + e_f|x| &< 0 \\ ax^2 &> e_f|x| \\ |x| &> \frac{e_f}{a} \end{aligned}$$

Ou seja, sabemos que na região em que $|x| > e_f/a$ a derivada temporal da função de Lyapunov será negativa e o sistema convergirá em direção à origem. Entretanto, nosso aproximador também tem um limite em sua região de validade (nesse exemplo vamos considerar uma região do tipo $|x| < \beta$), e com isso, só podemos afirmar que o sistema converge em direção à origem, e melhora os parâmetros adaptados, na região

$$\frac{e_f}{a} < |x| < \beta$$

A estabilidade do sistema (no sentido de Lyapunov), no exemplo da última aula, também nos garantia uma outra propriedade importante, a de que os parâmetros sendo adaptados ficam limitados em malha fechada.

Neste caso que estamos considerando, na região $|x| < e_f/a$, não existe nenhuma garantia de que os parâmetros sendo adaptados fiquem limitados (porque o erro entre os parâmetros reais e os parâmetros adaptados pode crescer indefinidamente sem que o estado deixe a região $|x| < e_f/a$), e dizemos que ocorre uma *deriva dos parâmetros*.

A deriva dos parâmetros pode levar à erros numéricos, a ações de controle muito grandes e no pior dos casos à instabilidade do sistema. Para evitarmos a deriva dos parâmetros, podemos utilizar modificações que deixem a lei de adaptação robusta, como a modificação de projeção e a zona morta (as quais veremos na próxima aula).

Seguimento de Trajetória

Vamos considerar agora que, ao invés de levar o estado para a origem, desejamos que ele siga um sinal de referência, x_d , e que a derivada temporal deste sinal de referência, \dot{x}_d , está disponível.

Considerando um sistema descrito por

$$\dot{x} = f(x) + g(x)u$$

se utilizarmos a lei de controle

$$u = \frac{-a(x - x_d) + \dot{x}_d - f(x)}{g(x)}$$

e definirmos o erro de seguimento como $e = x - x_d$, a malha fechada do sistema é dada por

$$\begin{aligned}\dot{x} &= -a(x - x_d) + \dot{x}_d \\ \dot{x} - \dot{x}_d &= -a(x - x_d) \\ \dot{e} &= -ae\end{aligned}$$

e vemos que a dinâmica do erro é assintoticamente estável em malha fechada, para $a > 0$ (ou seja, o erro de seguimento é limitado e $\lim_{t \rightarrow \infty} e = 0$).

Considere agora que a dinâmica do sistema é dada por

$$\dot{x} = f_0(x) + f(x) + (g_0(x) + g(x)) u$$

em que $f_0(x)$ e $g_0(x)$ são funções conhecidas, e $f(x)$ e $g(x)$ são desconhecidas.

Assim como no caso da estabilização, vamos utilizar aproximadores (com modelos TS) para representar essas funções incertas.

Utilizando os aproximadores, podemos escrever

$$f(x) = \mathbf{h}_f^T(x)\boldsymbol{\theta}_f^* + \varepsilon_f(x)$$

$$g(x) = \mathbf{h}_g^T(x)\boldsymbol{\theta}_g^* + \varepsilon_g(x)$$

em que consideramos que os erros de aproximação $\varepsilon_f(x)$ e $\varepsilon_g(x)$ são dados por

$$\varepsilon_f(x) = f(x) - \mathbf{h}_f^T(x)\boldsymbol{\theta}_f^*$$

$$\varepsilon_g(x) = g(x) - \mathbf{h}_g^T(x)\boldsymbol{\theta}_g^*$$

com $\boldsymbol{\theta}_f^*$ e $\boldsymbol{\theta}_g^*$ os parâmetros que resultam no menor erro máximo de representação das funções. Ou seja,

$$\boldsymbol{\theta}_f^* = \arg \min \left(\max_x |f(x) - \mathbf{h}_f^T(x)\boldsymbol{\theta}_f| \right)$$

$$\boldsymbol{\theta}_g^* = \arg \min \left(\max_x |g(x) - \mathbf{h}_g^T(x)\boldsymbol{\theta}_g| \right)$$

Com essa representação, sabemos que

$$|\varepsilon_f(x)| \leq \max_x |f(x) - \mathbf{h}_f^T(x)\boldsymbol{\theta}_f^*| = e_f$$

$$|\varepsilon_g(x)| \leq \max_x |g(x) - \mathbf{h}_g^T(x)\boldsymbol{\theta}_g^*| = e_g$$

$$|\varepsilon_f(x) + \varepsilon_g(x)| \leq e_f + e_g = \delta$$

Logo, a dinâmica do sistema pode ser escrita como

$$\dot{x} = f_0(x) + \mathbf{h}_f^T(x)\boldsymbol{\theta}_f^* + \varepsilon_f(x) + \left(g_0(x) + \mathbf{h}_g^T(x)\boldsymbol{\theta}_g^* + \varepsilon_g(x) \right) u$$

Utilizando uma lei de controle da forma

$$u = \frac{-f_0(x) - \mathbf{h}_f^T(x)\hat{\boldsymbol{\theta}}_f - ae + \dot{x}_d}{g_0(x) + \mathbf{h}_g^T(x)\hat{\boldsymbol{\theta}}_g}$$

obtemos a malha fechada

$$\dot{x} = f_0(x) + \mathbf{h}_f^T(x)\boldsymbol{\theta}_f^* + \varepsilon_f(x) + \left(g_0(x) + \mathbf{h}_g^T(x)\boldsymbol{\theta}_g^* + \varepsilon_g(x) \right) \left(\frac{-f_0(x) - \mathbf{h}_f^T(x)\hat{\boldsymbol{\theta}}_f - ae + \dot{x}_d}{g_0(x) + \mathbf{h}_g^T(x)\hat{\boldsymbol{\theta}}_g} \right)$$

$$\begin{aligned} \dot{x} &= f_0(x) + \mathbf{h}_f^T(x)\boldsymbol{\theta}_f^* + \varepsilon_f(x) \\ &+ \left(g_0(x) + \mathbf{h}_g^T(x)\boldsymbol{\theta}_g^* - \mathbf{h}_g^T(x)\hat{\boldsymbol{\theta}}_g + \mathbf{h}_g^T(x)\hat{\boldsymbol{\theta}}_g + \varepsilon_g(x) \right) \left(\frac{-f_0(x) - \mathbf{h}_f^T(x)\hat{\boldsymbol{\theta}}_f - ae + \dot{x}_d}{g_0(x) + \mathbf{h}_g^T(x)\hat{\boldsymbol{\theta}}_g} \right) \end{aligned}$$

$$\begin{aligned} \dot{x} &= f_0(x) + \mathbf{h}_f^T(x)\boldsymbol{\theta}_f^* + \varepsilon_f(x) + \left(g_0(x) + \mathbf{h}_g^T(x)\hat{\boldsymbol{\theta}}_g \right) \left(\frac{-f_0(x) - \mathbf{h}_f^T(x)\hat{\boldsymbol{\theta}}_f - ae + \dot{x}_d}{g_0(x) + \mathbf{h}_g^T(x)\hat{\boldsymbol{\theta}}_g} \right) \\ &+ \left(\mathbf{h}_g^T(x)\boldsymbol{\theta}_g^* - \mathbf{h}_g^T(x)\hat{\boldsymbol{\theta}}_g + \varepsilon_g(x) \right) u \end{aligned}$$

O que nos leva à uma dinâmica do erro em malha fechada

$$\dot{e} = -ae + \mathbf{h}_f^T(x) \left(\boldsymbol{\theta}_f^* - \hat{\boldsymbol{\theta}}_f \right) + \varepsilon_f(x) + \left(\mathbf{h}_g^T(x) \left(\boldsymbol{\theta}_g^* - \hat{\boldsymbol{\theta}}_g \right) + \varepsilon_g(x) \right) u$$

Escolhendo a função de Lyapunov

$$V(e, \tilde{\boldsymbol{\theta}}_f, \tilde{\boldsymbol{\theta}}_g) = \frac{1}{2}e^2 + \frac{1}{2}\tilde{\boldsymbol{\theta}}_f^T \Gamma_f^{-1} \tilde{\boldsymbol{\theta}}_f + \frac{1}{2}\tilde{\boldsymbol{\theta}}_g^T \Gamma_g^{-1} \tilde{\boldsymbol{\theta}}_g$$

sua derivada temporal será dada por

$$\begin{aligned} \dot{V} &= e\dot{e} - \tilde{\boldsymbol{\theta}}_f^T \Gamma_f^{-1} \dot{\tilde{\boldsymbol{\theta}}}_f - \tilde{\boldsymbol{\theta}}_g^T \Gamma_g^{-1} \dot{\tilde{\boldsymbol{\theta}}}_g \\ \dot{V} &= -ae^2 + \tilde{\boldsymbol{\theta}}_f^T \mathbf{h}_f(x)e + e\varepsilon_f(x) + \tilde{\boldsymbol{\theta}}_g^T \mathbf{h}_g(x)eu + \varepsilon_g(x)eu - \tilde{\boldsymbol{\theta}}_f^T \Gamma_f^{-1} \dot{\tilde{\boldsymbol{\theta}}}_f - \tilde{\boldsymbol{\theta}}_g^T \Gamma_g^{-1} \dot{\tilde{\boldsymbol{\theta}}}_g \end{aligned}$$

Usando as leis de adaptação

$$\begin{aligned} \dot{\tilde{\boldsymbol{\theta}}}_f &= \Gamma_f \mathbf{h}_f(x)e \\ \dot{\tilde{\boldsymbol{\theta}}}_g &= \Gamma_g \mathbf{h}_g(x)eu \end{aligned}$$

a derivada temporal da função de Lyapunov escolhida passa a ser dada por

$$\dot{V} = -ae^2 + e\varepsilon_f(x) + \varepsilon_g(x)eu$$

Se $\varepsilon_f(x) = 0$ e $\varepsilon_g(x) = 0$, teríamos $\dot{V} = -ae^2 \leq 0$, e usando o Lema de Barbalat (como na última aula), poderíamos provar a estabilidade assintótica do erro ($\lim_{t \rightarrow \infty} e = 0$).

Na próxima aula, veremos como lidar com o caso em que $\varepsilon_f(x) \neq 0$ e $\varepsilon_g(x) \neq 0$.

Aula 28

Modificações robustas das leis de adaptação

Na última aula, vimos que quando existe um erro de aproximação (ou seja, quando a função aproximada não pode ser perfeitamente representada pelo aproximador escolhido) perdemos a garantia de estabilidade assintótica do sistema em malha fechada. Além disso, sem nenhuma modificação, pode ocorrer deriva dos parâmetros e os parâmetros estimados podem crescer sem limites.

Na aula de hoje, veremos modificações que podem ser utilizadas nas leis de adaptação para garantir que os parâmetros fiquem contidos em uma região desejada (projeção) e impedir que o erro de adaptação cresça sem limites (zona morta).

Modificação de Projeção

Na modificação de projeção, consideramos que nossos parâmetros estão contidos em um conjunto convexo S fechado com uma fronteira suave descrito por

$$S = \{\boldsymbol{\theta} \in \mathbb{R}^{q_d} \mid \kappa(\boldsymbol{\theta}) \leq 0\}$$

A ideia dessa modificação é *projetar* a lei de adaptação para dentro do conjunto quando o parâmetro atinge a borda, prevenindo que ele deixe a região desejada.

Ou seja, se tivéssemos uma lei de adaptação da forma

$$\dot{\boldsymbol{\theta}} = \Gamma \boldsymbol{\zeta} e$$

ela seria modificada para

$$\hat{\theta} = P_S(\Gamma \xi e) = \begin{cases} \Gamma \xi e, & \text{se } \hat{\theta} \in S \text{ ou } (\hat{\theta} \in \partial S \text{ e } \nabla \kappa^T \Gamma \xi e \leq 0) \\ \Gamma \xi e - \Gamma \frac{\nabla \kappa \nabla \kappa^T}{\nabla \kappa^T \Gamma \nabla \kappa} \Gamma \xi e, & \text{caso contrário} \end{cases}$$

Ou seja, a lei de adaptação permanece a mesma se o parâmetro está dentro da região desejada, ou se ele está na fronteira da região, mas “apontando” para dentro. Caso contrário, utiliza-se uma projeção que retira do vetor a componente que está “apontando” para fora do conjunto.

Um caso muito comum ocorre quando escolhemos uma matriz de aprendizado diagonal $\Gamma = \text{diag}(\gamma_1, \dots, \gamma_{q_d})$, e temos limites impostos separadamente sobre cada parâmetro, da forma $\theta_i \in [\underline{\theta}_i, \bar{\theta}_i]$. Neste caso, é fácil mostrar que a projeção toma a forma

$$\hat{\theta} = \begin{cases} \gamma_i \xi_i e, & \text{se } \hat{\theta}_i \in [\underline{\theta}_i, \bar{\theta}_i], \text{ ou } (\hat{\theta}_i = \underline{\theta}_i \text{ e } \gamma_i \xi_i e \geq 0) \text{ ou } (\hat{\theta}_i = \bar{\theta}_i \text{ e } \gamma_i \xi_i e \leq 0) \\ 0, & \text{caso contrário} \end{cases}$$

Uma propriedade muito importante da modificação de projeção é que ela normalmente mantém as características de estabilidade da lei de adaptação antes da projeção.

Para ver isso, note que, enquanto $\hat{\theta}$ está no interior do conjunto S não ocorre nenhuma modificação na análise de estabilidade. Considerando que a parte da função de Lyapunov relativa ao erro de estimação é dada por $\tilde{\theta}^T \Gamma^{-1} \tilde{\theta}$, quando $\hat{\theta} \in \partial S$ e $\nabla \kappa^T \Gamma \xi e > 0$ temos um termo adicional na derivada temporal da função de Lyapunov dado por

$$\tilde{\theta}^T \frac{\nabla \kappa \nabla \kappa^T}{\nabla \kappa^T \Gamma \nabla \kappa} \Gamma \xi e$$

Entretanto, como S é convexo e consideramos que o parâmetro ótimo $\theta^* \in S$, temos que $\tilde{\theta}^T \nabla \kappa = (\theta^* - \tilde{\theta})^T \nabla \kappa \leq 0$, mas como este termo só aparece quando $\nabla \kappa^T \Gamma \xi e > 0$, temos que

$$\tilde{\theta}^T \frac{\nabla \kappa \nabla \kappa^T}{\nabla \kappa^T \Gamma \nabla \kappa} \Gamma \xi e \leq 0$$

e nada muda na análise de estabilidade.

Modificação de Zona Morta

Um dos problemas que vimos nas leis de adaptação até agora é que, quando existe um erro de aproximação, existe uma faixa de valores para os quais as leis de adaptação podem aumentar indefinidamente o erro de aproximação (deriva dos parâmetros).

A ideia simples da modificação de zona morta, é que **desligamos** a lei de adaptação na região em que não garantimos convergência pela função de Lyapunov (como não sabemos exatamente a região, normalmente escolhemos uma região um pouco maior).

Como a modificação de zona morta não modifica o comportamento do sistema fora da região de zona morta, a análise para esta região permanece a mesma. Entretanto, dentro da zona morta, como a lei de adaptação está desligada, garantimos que o erro de adaptação não cresce.

Voltando ao problema de seguimento de trajetória

Na última aula, vimos que se tivermos um sistema descrito por

$$\dot{x} = f_0(x) + f(x) + (g_0(x) + g(x))u$$

em que $f_0(x)$ e $g_0(x)$ são funções conhecidas, e $f(x)$ e $g(x)$ são funções desconhecidas, considerando que ao usar aproximadores podemos descrever as funções desconhecidas como

$$f(x) = \mathbf{h}_f^T(x)\boldsymbol{\theta}_f^* + \varepsilon_f(x)$$

$$g(x) = \mathbf{h}_g^T(x)\boldsymbol{\theta}_g^* + \varepsilon_g(x)$$

A lei de controle

$$u = \frac{-f_0(x) - \mathbf{h}_f^T(x)\hat{\boldsymbol{\theta}}_f - ae + \dot{x}_d}{g_0(x) + \mathbf{h}_g^T(x)\hat{\boldsymbol{\theta}}_g}$$

juntamente com as leis de adaptação

$$\dot{\hat{\boldsymbol{\theta}}}_f = \Gamma_f \mathbf{h}_f(x)e$$

$$\dot{\hat{\boldsymbol{\theta}}}_g = \Gamma_g \mathbf{h}_g(x)eu$$

garantem a estabilidade assintótica do erro de seguimento quando não existe nenhum erro de aproximação (e $g_0(x) + \mathbf{h}_g^T(x)\hat{\boldsymbol{\theta}}_g > 0$ ou $g_0(x) + \mathbf{h}_g^T(x)\hat{\boldsymbol{\theta}}_g < 0$). Entretanto, no caso em que existem os erros de aproximação, vimos que, considerando a função de Lyapunov

$$V(e, \tilde{\boldsymbol{\theta}}_f, \tilde{\boldsymbol{\theta}}_g) = \frac{1}{2}e^2 + \frac{1}{2}\tilde{\boldsymbol{\theta}}_f^T \Gamma_f^{-1} \tilde{\boldsymbol{\theta}}_f + \frac{1}{2}\tilde{\boldsymbol{\theta}}_g^T \Gamma_g^{-1} \tilde{\boldsymbol{\theta}}_g$$

temos

$$\dot{V} = -ae^2 + e\varepsilon_f(x) + \varepsilon_g(x)eu$$

e pode ocorrer deriva dos parâmetros (além de não termos considerado nenhuma forma de garantir $g_0(x) + \mathbf{h}_g^T(x)\hat{\boldsymbol{\theta}}_g > 0$ ou $g_0(x) + \mathbf{h}_g^T(x)\hat{\boldsymbol{\theta}}_g < 0$).

Mas, com as modificações vistas nesta aula, podemos fazer algumas melhorias e termos melhores garantias. Primeiramente, vamos fazer uma modificação de projeção

nas leis de adaptação de forma que podemos garantir que os parâmetros adaptados (e com isso o erro de adaptação) sempre estarão limitados.

Sendo assim, devemos escolher conjunto convexos para descrever as regiões que contém os parâmetros $\hat{\theta}_f$ e $\hat{\theta}_g$. É importante escolher a região dos parâmetros de $g(x)$ de tal forma a garantir que $g_0(x) + \mathbf{h}_g^T(x)\hat{\theta}_g > 0$ ou $g_0(x) + \mathbf{h}_g^T(x)\hat{\theta}_g < 0$, para garantir que a lei de controle existe em toda a região sendo analisada.

Consideramos então que as leis de adaptação passam a ser dadas por

$$\begin{aligned}\dot{\hat{\theta}}_f &= P_{S_f}(\Gamma_f \mathbf{h}_f(x)e) \\ \dot{\hat{\theta}}_g &= P_{S_g}(\Gamma_g \mathbf{h}_g(x)eu)\end{aligned}$$

com S_f e S_g os conjuntos convexos que contém os parâmetros estimados.

Como vimos anteriormente, a modificação de projeção não modifica as propriedades de estabilidade do sistema, porém garante que o erro de adaptação será sempre limitado.

Além disso, considerando que

$$0 \leq |\varepsilon_f(x) + \varepsilon_g(x)u| \leq \delta$$

podemos escrever que

$$\dot{V} \leq -ae^2 + |e|\delta$$

e temos que $\dot{V} \leq 0$ quando $|e| > \delta/a$. A ideia é então utilizar uma modificação de zona morta na região $\delta/a + \phi$.

Definindo a função

$$d(e, \tilde{\phi}) = \begin{cases} e, & \text{se } |e| \geq \tilde{\phi} \\ 0, & \text{caso contrário} \end{cases}$$

e $\tilde{\phi} = \delta/a + \delta$, podemos escrever a lei de adaptação com projeção e zona morta como

$$\begin{aligned}\dot{\hat{\theta}}_f &= P_{S_f}(\Gamma_f \mathbf{h}_f(x)d(e, \tilde{\phi})) \\ \dot{\hat{\theta}}_g &= P_{S_g}(\Gamma_g \mathbf{h}_g(x)d(e, \tilde{\phi})u)\end{aligned}$$

Com ambas modificações, podemos mostrar que

- $e, x, \hat{\theta}_f, \hat{\theta}_g$ são limitados
- $e(t)$ é uniformemente finalmente limitado (UUB) \rightarrow o tempo total em que $|e| > \tilde{\phi}$ é finito

Suponha que a condição $|e| > \tilde{\phi}$ seja satisfeita para $t \in (t_{s_i}, t_{f_i})$, $i = 1, 2, 3, \dots$, com $t_{s_i} < t_{f_i} \leq t_{s_{i+1}}$, e $|e| \leq \tilde{\phi}$ para $t \in (t_{f_i}, t_{s_{i+1}})$, e $t_{s_{i+1}}$ pode ser infinito para algum i .

A derivada temporal da função de Lyapunov pode ser reescrita, para $t \in (t_{s_i}, t_{f_i})$, como

$$\begin{aligned}\dot{V} &\leq -ae^2 + |e|\delta \\ \dot{V} &\leq -a|e| \left(|e| - \frac{\delta}{a} \right) \\ \dot{V} &\leq -a\tilde{\phi}\phi\end{aligned}$$

Integrando no tempo dos dois lados da desigualdade

$$V(t_{f_i}) \leq V(t_{s_i}) - a\phi\tilde{\phi}(t_{f_i} - t_{s_i})$$

Note que, devido à zona morta, enquanto $t \in (t_{f_i}, t_{s_{i+1}})$, a adaptação dos parâmetros está desligada. Além disso, $|e(t_{f_i})| = |e(t_{s_{i+1}})| = \tilde{\phi}$, logo $V(t_{f_i}) = V(t_{s_{i+1}})$, logo

$$\begin{aligned}V(t_{f_i}) &\leq V(t_{s_i}) - a\phi\tilde{\phi}(t_{f_i} - t_{s_i}) \\ V(t_{f_i}) &\leq V(t_{f_{i-1}}) - a\phi\tilde{\phi}(t_{f_i} - t_{s_i}) \\ V(t_{f_i}) &\leq V(t_{f_{i-2}}) - a\phi\tilde{\phi}((t_{f_i} - t_{s_i}) + (t_{f_{i-1}} - t_{s_{i-1}})) \\ &\vdots \\ V(t_{f_i}) &\leq V(0) - a\phi\tilde{\phi} \sum_{k=1}^i (t_{f_k} - t_{s_k})\end{aligned}$$

Como $V(t_{f_i}) \geq 0$

$$\sum_{k=1}^i (t_{f_k} - t_{s_k}) \leq \frac{V(0)}{a\phi\tilde{\phi}}$$

o que prova que o tempo fora da zona morta é finito.

Aula 29

Filtros de geração de trajetória e feedback linearization (Input-State) com aproximação adaptativa

Filtros de geração de trajetória

Até o momento, para todas as estratégias desta unidade para o seguimento de trajetória, supomos que o sinal da derivada temporal da referência desejada está disponível.

Além disso, em casos que desejamos controlar sistemas de ordem mais alta (como veremos na aula de hoje) ordens mais altas das derivadas temporais do sinal de referência são necessárias.

Uma estratégia muito utilizada para se obter esses sinais é a de se utilizar um filtro de geração de trajetória. Nesta estratégia, o sinal de referência desejada passa primeiro por um filtro (estável e de ganho unitário na banda de passagem), e implementamos o filtro de tal forma que seus estados sejam as derivadas temporais do sinal de saída do filtro.

Note que, neste caso, passamos a seguir a trajetória dada pelo sinal filtrado e não pelo sinal original. Entretanto, utilizando-se um filtro com banda passante muito maior do que a maior frequência do sinal de referência desejado, a distorção introduzida pelo filtro será pequena.

Considerando que o sinal de referência desejado é dado por $y_c(t)$ e que a saída do

filtro é dada por $y_d(t)$, normalmente utilizamos um filtro da forma

$$\dot{\mathbf{z}}(t) = \begin{bmatrix} 0 & 1 & 0 \dots & 0 & \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{r-1} \end{bmatrix} \mathbf{z}(t) + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ a_0 \end{bmatrix}$$

$$\mathbf{z}(t) = \begin{bmatrix} y_d(t) \\ \dot{y}_d(t) \\ \vdots \\ y_d^{(r-2)} \\ y_d^{(r-1)} \end{bmatrix}$$

que tem função de transferência

$$\frac{Y_d(s)}{Y_c(s)} = \frac{a_0}{s^r + a_{r-1}s^{r-1} + \dots + a_2s^2 + a_1s + a_0}$$

Logo, durante o projeto deste filtro, deve-se ter certeza de que os parâmetros a_i do filtro definem um filtro estável.

Feedback linearization (Input-State ou Input-Output desprezando a dinâmica zero) com aproximação adaptativa

Consideramos o caso do sistema representado na forma companheira

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\vdots \\ \dot{x}_n &= f_0(\mathbf{x}) + f(\mathbf{x}) + (g_0(\mathbf{x}) + g(\mathbf{x}))u \\ y &= x_1 \end{aligned}$$

com $f_0(\mathbf{x})$ e $g_0(\mathbf{x})$ conhecidos e $f(\mathbf{x})$ e $g(\mathbf{x})$ desconhecidos.

A ideia aqui é fazermos uma lei de controle com aproximação que leve o sistema a seguir a referência $y_c(t)$, mas para elaborarmos tal lei, precisamos de fazer uso de n derivadas temporais do sinal de referência (para servirem como referência para os nossos estados).

Nesse caso, fazemos uso de filtros de geração de trajetória (como mostrado acima) e ao invés disso, fazemos uma lei de controle que siga o sinal de referência filtrada $y_d(t)$ (já que temos as derivadas temporais deste sinal disponíveis).

Note que podemos definir o vetor de erro de estados, $\tilde{\mathbf{x}}$, como

$$\tilde{\mathbf{x}} = \begin{bmatrix} x_1 - y_d(t) \\ x_2 - \dot{y}_d(t) \\ \vdots \\ x_{n-1} - y_d^{(n-2)} \\ x_n - y_d^{(n-1)} \end{bmatrix}$$

e se tivermos disponíveis os sinais $y_d(t), \dots, y_d^{(n)}(t)$, podemos escrever a dinâmica do erro, em malha aberta, como

$$\begin{aligned} \dot{\tilde{x}}_1 &= \tilde{x}_2 \\ \dot{\tilde{x}}_2 &= \tilde{x}_3 \\ &\vdots \\ \dot{\tilde{x}}_x &= f_0(\mathbf{x}) + f(\mathbf{x}) + (g_0(\mathbf{x}) + g(\mathbf{x}))u - y_d^{(n)} \end{aligned}$$

que pode ser escrito de forma matricial como

$$\dot{\tilde{\mathbf{x}}} = A\tilde{\mathbf{x}} + B \left(f_0(\mathbf{x}) + f(\mathbf{x}) + (g_0(\mathbf{x}) + g(\mathbf{x}))u - y_d^{(n)} \right)$$

com

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

O ideal, seria utilizar uma lei de controle da forma

$$u = \frac{y_d^{(n)} - f_0(\mathbf{x}) - f(\mathbf{x}) + K\tilde{\mathbf{x}}}{g_0(\mathbf{x}) + g(\mathbf{x})}$$

em que a parte $K\tilde{\mathbf{x}} = k_1\tilde{x}_1 + k_2\tilde{x}_2 + \dots + k_n\tilde{x}_n$ garante que o sistema linear resultante

$$\dot{\tilde{\mathbf{x}}} = (A + BK)\tilde{\mathbf{x}}$$

seja assintoticamente estável (ou seja, tem todos os autovalores com parte real negativa) em malha fechada.

Entretanto, não podemos utilizar $f(\mathbf{x})$ e $g(\mathbf{x})$ na lei de controle, pois são funções desconhecidas. Portanto, fazemos uso de aproximadores em seus lugares.

Utilizando modelos Takagi-Sugeno para aproximar as funções desconhecidas, podemos escrever

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{h}_f^T(\mathbf{x})\boldsymbol{\theta}_f^* + \varepsilon_f(\mathbf{x}) \\ g(\mathbf{x}) &= \mathbf{h}_g^T(\mathbf{x})\boldsymbol{\theta}_g^* + \varepsilon_g(\mathbf{x}) \end{aligned}$$

e utilizar a lei de controle

$$u = \frac{y_d^{(n)} - f_0(\mathbf{x}) - \mathbf{h}_f^T(\mathbf{x})\widehat{\boldsymbol{\theta}}_f + K\tilde{\mathbf{x}}}{g_0(\mathbf{x}) + \mathbf{h}_g^T(\mathbf{x})\widehat{\boldsymbol{\theta}}_g}$$

Além disso, podemos escrever a dinâmica do erro em malha fechada como

$$\dot{\tilde{\mathbf{x}}} = A\tilde{\mathbf{x}} + B \left(f_0(\mathbf{x}) + \mathbf{h}_f^T(\mathbf{x})\boldsymbol{\theta}_f^* + \varepsilon_f(\mathbf{x}) + \left(g_0(\mathbf{x}) + \mathbf{h}_g^T(\mathbf{x})\boldsymbol{\theta}_g^* + \varepsilon_g(\mathbf{x}) \right) u - y_d^{(n)} \right)$$

$$\dot{\tilde{\mathbf{x}}} = A\tilde{\mathbf{x}} + B \left(f_0(\mathbf{x}) + \mathbf{h}_f^T(\mathbf{x})\boldsymbol{\theta}_f^* + \varepsilon_f(\mathbf{x}) + \left(g_0(\mathbf{x}) + \mathbf{h}_g^T(\mathbf{x})\widehat{\boldsymbol{\theta}}_g \right) u + \left(\mathbf{h}_g^T(\mathbf{x}) \left(\boldsymbol{\theta}_g^* - \widehat{\boldsymbol{\theta}}_g \right) + \varepsilon_g(\mathbf{x}) \right) u - y_d^{(n)} \right)$$

Substituindo a lei de controle, temos

$$\dot{\tilde{\mathbf{x}}} = (A + BK)\tilde{\mathbf{x}} + B \left(\mathbf{h}_f^T(\mathbf{x})\widetilde{\boldsymbol{\theta}}_f + \mathbf{h}_g^T(\mathbf{x})\widetilde{\boldsymbol{\theta}}_g u + \varepsilon_f(\mathbf{x}) + \varepsilon_g(\mathbf{x})u \right)$$

em que $\widetilde{\boldsymbol{\theta}}_f = \boldsymbol{\theta}_f^* - \widehat{\boldsymbol{\theta}}_f$ e $\widetilde{\boldsymbol{\theta}}_g = \boldsymbol{\theta}_g^* - \widehat{\boldsymbol{\theta}}_g$.

Lembrando que o controlador K é escolhido de forma que $A + BK$ seja assintoticamente estável, sabemos que existe uma matriz P definida positiva que resolve a equação de Lyapunov

$$(A + BK)^T P + P(A + BK) = -Q$$

para qualquer Q definida positiva que escolhermos. Sem perda de generalidade, vamos escolher $Q = I$, de modo que $(A + BK)^T P + P(A + BK) = -I$ e

$$\tilde{\mathbf{x}} \left((A + BK)^T P + P(A + BK) \right) \tilde{\mathbf{x}} = -\tilde{\mathbf{x}}^T \tilde{\mathbf{x}}$$

A partir dessa matriz P , definimos a seguinte função de Lyapunov para estudar a estabilidade do sistema

$$V(\tilde{\mathbf{x}}, \widetilde{\boldsymbol{\theta}}_f, \widetilde{\boldsymbol{\theta}}_g) = \tilde{\mathbf{x}}^T P \tilde{\mathbf{x}} + \widetilde{\boldsymbol{\theta}}_f^T \Gamma_f^{-1} \widehat{\boldsymbol{\theta}}_f + \widetilde{\boldsymbol{\theta}}_g^T \Gamma_g^{-1} \widehat{\boldsymbol{\theta}}_g$$

de modo que

$$\dot{V} = 2\tilde{\mathbf{x}}^T P \dot{\tilde{\mathbf{x}}} - 2\widetilde{\boldsymbol{\theta}}_f^T \Gamma_f^{-1} \dot{\widehat{\boldsymbol{\theta}}}_f - 2\widetilde{\boldsymbol{\theta}}_g^T \Gamma_g^{-1} \dot{\widehat{\boldsymbol{\theta}}}_g$$

$$\begin{aligned} \dot{V} &= 2\tilde{\mathbf{x}}^T (A + BK) P \tilde{\mathbf{x}} + 2\widetilde{\boldsymbol{\theta}}_f^T \mathbf{h}_f(\mathbf{x}) B^T P \tilde{\mathbf{x}} + 2\widetilde{\boldsymbol{\theta}}_g^T \mathbf{h}_g(\mathbf{x}) B^T P \tilde{\mathbf{x}} u \\ &\quad + 2\varepsilon_f(\mathbf{x}) B^T P \tilde{\mathbf{x}} + 2\varepsilon_g(\mathbf{x}) u B^T P \tilde{\mathbf{x}} - 2\widetilde{\boldsymbol{\theta}}_f^T \Gamma_f^{-1} \dot{\widehat{\boldsymbol{\theta}}}_f - 2\widetilde{\boldsymbol{\theta}}_g^T \Gamma_g^{-1} \dot{\widehat{\boldsymbol{\theta}}}_g \end{aligned}$$

$$\begin{aligned} \dot{V} &= -\tilde{\mathbf{x}}^T \tilde{\mathbf{x}} + 2\widetilde{\boldsymbol{\theta}}_f^T \left(\mathbf{h}_f(\mathbf{x}) B^T P \tilde{\mathbf{x}} - \Gamma_f^{-1} \dot{\widehat{\boldsymbol{\theta}}}_f \right) + 2\widetilde{\boldsymbol{\theta}}_g^T \left(\mathbf{h}_g(\mathbf{x}) B^T P \tilde{\mathbf{x}} u - \Gamma_g^{-1} \dot{\widehat{\boldsymbol{\theta}}}_g \right) \\ &\quad + 2\varepsilon_f(\mathbf{x}) B^T P \tilde{\mathbf{x}} + 2\varepsilon_g(\mathbf{x}) u B^T P \tilde{\mathbf{x}} \end{aligned}$$

e escolhendo as leis de adaptação

$$\begin{aligned}\dot{\hat{\theta}}_f &= \Gamma_f \mathbf{h}_f(\mathbf{x}) B^T P \tilde{\mathbf{x}} \\ \dot{\hat{\theta}}_g &= \Gamma_g \mathbf{h}_g(\mathbf{x}) B^T P \tilde{\mathbf{x}} u\end{aligned}$$

chegamos em

$$\dot{V} = -\tilde{\mathbf{x}}^T \tilde{\mathbf{x}} + 2\varepsilon_f(\mathbf{x}) B^T P \tilde{\mathbf{x}} + 2\varepsilon_g(\mathbf{x}) u B^T P \tilde{\mathbf{x}}$$

Se não existisse nenhum erro de aproximação, chegaríamos em um sistema estável em malha fechada, e usando o Lema de Barbalat, poderíamos mostrar que o erro de seguimento (representado por $\tilde{\mathbf{x}}$) seria assintoticamente estável (convergiria para zero).

Entretanto, sabemos que tal suposição não é muito real e que a lei de adaptação apresentada não tem nenhuma garantia de que $g_0(\mathbf{x}) + \mathbf{h}_g(\mathbf{x}) \hat{\theta}_g \neq 0$ (para impedir que nossa ação de controle “exploda”. Além disso, como vimos nas últimas aulas, numa situação como essa (levando os erros de aproximação em conta) não existe nada que impeça o fenômeno de deriva dos parâmetros sendo adaptados.

Dessa forma, primeiro consideramos que os parâmetros sendo adaptados estão restritos aos conjuntos convexos S_f e S_g e fazemos as modificações de projeção

$$\begin{aligned}\hat{\theta}_f &= P_{S_f}(\Gamma_f \mathbf{h}_f(\mathbf{x}) B^T P \tilde{\mathbf{x}}) \\ \hat{\theta}_g &= P_{S_g}(\Gamma_g \mathbf{h}_g(\mathbf{x}) B^T P \tilde{\mathbf{x}} u)\end{aligned}$$

e como sabemos, da última aula, tal modificação garante que os parâmetros fiquem limitados em uma região limitada e, se os parâmetros ótimos estiverem contidos dentro da região desejada, não modifica em nada a análise de estabilidade do sistema.

Como na última aula, considerando que

$$|\varepsilon_f(\mathbf{x}) + \varepsilon_g(\mathbf{x}) u| \leq \delta$$

podemos escrever que

$$\begin{aligned}\dot{V} &= -\tilde{\mathbf{x}}^T \tilde{\mathbf{x}} + 2\varepsilon_f(\mathbf{x}) B^T P \tilde{\mathbf{x}} + 2\varepsilon_g(\mathbf{x}) u B^T P \tilde{\mathbf{x}} \\ \dot{V} &\leq -\|\tilde{\mathbf{x}}\|_2^2 + 2\delta \|B^T P\|_2 \|\tilde{\mathbf{x}}\|_2\end{aligned}$$

e temos que $\dot{V} \leq 0$ quando $\|\tilde{\mathbf{x}}\|_2 > 2\delta \|B^T P\|_2$. A ideia é então usar uma zona morta para desativar a lei de adaptação em torno dessa região. Para garantir algumas propriedades de estabilidade interessantes (as mesmas vistas na última aula), usamos uma zona morta da forma

$$d(B^T P \tilde{\mathbf{x}}, \tilde{\mu}) = \begin{cases} B^T P \tilde{\mathbf{x}}, & \text{se } \tilde{\mathbf{x}}^T P \tilde{\mathbf{x}} \geq \bar{\lambda}_P \tilde{\mu} \\ 0, & \text{caso contrário} \end{cases}$$

com $\tilde{\mu} = 2\delta\|B^T P\|_2 + \mu$, $\bar{\lambda}_P$ o maior autovalor de P e $\mu > 0$ uma escolha de projeto.

Aplicando essa zona morta, podemos escrever as leis de adaptação como

$$\dot{\hat{\theta}}_f = P_{S_f}(\Gamma_f \mathbf{h}_f(\mathbf{x})d(B^T P \tilde{\mathbf{x}}, \tilde{\mu}))$$

$$\dot{\hat{\theta}}_g = P_{S_g}(\Gamma_g \mathbf{h}_g(\mathbf{x})d(B^T P \tilde{\mathbf{x}}, \tilde{\mu})u)$$

e, de forma muito semelhante à da última aula, podemos mostrar que a lei de controle utilizada, juntamente com a lei de adaptação modificada acima, garante que

- $\tilde{\mathbf{x}}, \mathbf{x}, \hat{\theta}_f, \hat{\theta}_g$ são limitados
- $\tilde{\mathbf{x}}(t)$ é uniformemente finalmente limitado (UUB) \rightarrow o tempo total que $\tilde{\mathbf{x}}^T P \tilde{\mathbf{x}} \geq \bar{\lambda}_P \tilde{\mu}$ é finito

Exercício 29.1

Considere o modelo de um pêndulo invertido sobre um carrinho (ignorando a dinâmica do carrinho) apresentado no Exercício 19.4:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{g \sin(x_1) - \frac{amlx_2^2 \sin(2x_1)}{2} - a \cos(x_1)u}{\frac{4l}{3} - aml \cos^2(x_1)} \end{aligned}$$

Projete uma lei de controle adaptativa para o sistema, supondo que a aceleração da gravidade é desconhecida. Simule seu comportamento e comente.

Exercício 29.2

Considere agora o modelo de um pêndulo invertido sobre um carrinho (ignorando a dinâmica do carrinho) com atrito, apresentado no item 3 do Exercício 19.4:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{g \sin(x_1) - \frac{amlx_2^2 \sin(2x_1)}{2} - a \cos(x_1)u + v}{\frac{4l}{3} - aml \cos^2(x_1)} \end{aligned}$$

$$v = \begin{cases} 0, & \text{se } x_2 = 0, \\ -v_{\text{seco}} \frac{x_2}{|x_2|} - kx_2, & \text{caso contrário} \end{cases}$$

Projete uma lei de controle adaptativo por aproximação para o caso em que o atrito v é desconhecido. Simule seu comportamento e comente (em especial comparando com o controlador que não estima o atrito).

Aula 30

Backstepping com aproximação adaptativa

Outra técnica de controle de sistemas não-lineares que vimos nesta matéria foi o backstepping, que pode ser utilizada para sistemas na forma de realimentação estrita (strict-feedback form). Nesta aula, veremos como podemos utilizar uma abordagem de aproximação adaptativa com esta técnica.

Abordagem via filtro de comando

Consideramos um sistema da forma

$$\begin{aligned}\dot{x}_1 &= f_1^o(x_1) + f_1(x_1) + (g_1^o(x_1) + g_1(x_1)) x_2 \\ \dot{x}_2 &= f_2^o(x_1, x_2) + f_2(x_1, x_2) + (g_2^o(x_1, x_2) + g_2(x_1, x_2)) x_3 \\ &\vdots \\ \dot{x}_n &= f_n^o(\mathbf{x}) + f_n(\mathbf{x}) + (g_n^o(\mathbf{x}) + g_n(\mathbf{x})) u \\ y &= x_1\end{aligned}$$

em que as funções f_i e g_i são funções desconhecidas.

Assim como fizemos anteriormente, vamos considerar que as funções desconhecidas são substituídas por aproximadores universais

$$\begin{aligned}f_i(x_1, \dots, x_i) &= \mathbf{h}_{f_i}^T(x_1, \dots, x_i) \boldsymbol{\theta}_{f_i}^* + \varepsilon_{f_i}(x_1, \dots, x_i) \\ g_i(x_1, \dots, x_i) &= \mathbf{h}_{g_i}^T(x_1, \dots, x_i) \boldsymbol{\theta}_{g_i}^* + \varepsilon_{g_i}(x_1, \dots, x_i)\end{aligned}$$

Definimos os erros de rastreamento como

$$\begin{aligned} z_1 &= x_1 - y_d \\ z_2 &= x_2 - x_{2d} \\ &\vdots \\ z_n &= x_n - x_{nd} \end{aligned}$$

e podemos utilizar a lei de controle (semelhante à utilizada na aula de backstepping)

$$\alpha_i = \begin{cases} \frac{-f_1^o - \mathbf{h}_{f_1}^T \hat{\boldsymbol{\theta}}_{f_1} - k_1 z_1 + \dot{y}_d}{g_1^o + \mathbf{h}_{g_1}^T \hat{\boldsymbol{\theta}}_{g_1}}, & i = 1 \\ \frac{-f_i^o - \mathbf{h}_{f_i}^T \hat{\boldsymbol{\theta}}_{f_i} - k_i z_i + \dot{x}_{id} - s_{i-1} (g_{i-1}^o + \mathbf{h}_{g_{i-1}}^T \hat{\boldsymbol{\theta}}_{g_{i-1}})}{g_i^o + \mathbf{h}_{g_i}^T \hat{\boldsymbol{\theta}}_{g_i}}, & i = 2, \dots, n \end{cases}$$

$$x_{id}^o = \alpha_{i-1} - \zeta_i, \quad i = 2, \dots, n$$

$$\dot{\zeta}_i = \begin{cases} -k_i \zeta_i + (g_i^o + \mathbf{h}_{g_i}^T \hat{\boldsymbol{\theta}}_{g_i}) (x_{(i+1)d} - x_{(i+1)d}^o), & i = 1, \dots, (n-1) \\ 0, & i = n \end{cases}$$

$$s_i = z_i - \zeta_i, \quad i = 1, \dots, n$$

com $u = \alpha_n$ e x_{id} , $i \neq 1$ calculado por um filtro de comando (como o apresentado na aula de *backstepping* ou como os filtros de trajetória da última aula). Como anteriormente, temos que α_i são as variáveis de controle virtuais, x_{id}^o representam os estados desejados, x_{id} representam os estados desejados filtrados, ζ_i representa o erro filtrado entre o estado desejado e o estado desejado filtrado, e s_i representa o erro de trajetória compensado (que se aproxima do erro de trajetória original desde que o filtro de comando contanto que o estado desejado esteja razoavelmente dentro da banda de passagem do filtro de comando).

A dinâmica do erro de rastreamento z_1 pode então ser escrita como

$$\begin{aligned} \dot{z}_1 &= f_1^o + \mathbf{h}_{f_1}^T \boldsymbol{\theta}_{f_1}^* + \varepsilon_{f_1} + (g_1^o + \mathbf{h}_{g_1}^T \hat{\boldsymbol{\theta}}_{g_1}) x_2 + \mathbf{h}_{g_1}^T (\boldsymbol{\theta}_{g_1}^* - \hat{\boldsymbol{\theta}}_{g_1}) x_2 + \varepsilon_{g_1} x_2 - \dot{y}_d \\ &= f_1^o + \mathbf{h}_{f_1}^T \boldsymbol{\theta}_{f_1}^* + \varepsilon_{f_1} + \mathbf{h}_{g_1}^T \tilde{\boldsymbol{\theta}}_{g_1} x_2 + \varepsilon_{g_1} x_2 - \dot{y}_d + (g_1^o + \mathbf{h}_{g_1}^T \hat{\boldsymbol{\theta}}_{g_1}) x_{2d}^o \\ &\quad + (g_1^o + \mathbf{h}_{g_1}^T \hat{\boldsymbol{\theta}}_{g_1}) (x_{2d} - x_{2d}^o) + (g_1^o + \mathbf{h}_{g_1}^T \hat{\boldsymbol{\theta}}_{g_1}) (x_2 - x_{2d}) \\ &= -k_1 z_1 - (g_1^o + \mathbf{h}_{g_1}^T \hat{\boldsymbol{\theta}}_{g_1}) \zeta_2 + \mathbf{h}_{f_1}^T \tilde{\boldsymbol{\theta}}_{f_1} + \varepsilon_{f_1} + \mathbf{h}_{g_1}^T \tilde{\boldsymbol{\theta}}_{g_1} x_2 + \varepsilon_{g_1} x_2 \\ &\quad + (g_1^o + \mathbf{h}_{g_1}^T \hat{\boldsymbol{\theta}}_{g_1}) (x_{2d} - x_{2d}^o) + (g_1^o + \mathbf{h}_{g_1}^T \hat{\boldsymbol{\theta}}_{g_1}) z_2 \end{aligned}$$

$$\begin{aligned}\dot{z}_1 &= -k_1 z_1 + \left(g_1^o + \mathbf{h}_{g_1}^T \hat{\boldsymbol{\theta}}_{g_1} \right) s_2 + \mathbf{h}_{f_1}^T \tilde{\boldsymbol{\theta}}_{f_1} + \varepsilon_{f_1} + \mathbf{h}_{g_1}^T \tilde{\boldsymbol{\theta}}_{g_1} x_2 + \varepsilon_{g_1} x_2 \\ &\quad + \left(g_1^o + \mathbf{h}_{g_1}^T \hat{\boldsymbol{\theta}}_{g_1} \right) (x_{2d} - x_{2d}^o)\end{aligned}$$

A dinâmica do erro de rastreamento z_i para $i = 2, \dots, (n-1)$ pode ser escrita como

$$\begin{aligned}\dot{z}_i &= f_i^o + \mathbf{h}_{f_i}^T \boldsymbol{\theta}_{f_i}^* + \varepsilon_{f_i} + \left(g_i^o + \mathbf{h}_{g_i}^T \hat{\boldsymbol{\theta}}_{g_i} \right) x_{i+1} + \mathbf{h}_{g_i}^T \left(\boldsymbol{\theta}_{g_i}^* - \hat{\boldsymbol{\theta}}_{g_i} \right) x_{i+1} + \varepsilon_{g_i} x_{i+1} - \dot{x}_{id} \\ &= f_i^o + \mathbf{h}_{f_i}^T \boldsymbol{\theta}_{f_i}^* + \varepsilon_{f_i} + \mathbf{h}_{g_i}^T \tilde{\boldsymbol{\theta}}_{g_i} x_{i+1} + \varepsilon_{g_i} x_{i+1} - \dot{x}_{id} + \left(g_i^o + \mathbf{h}_{g_i}^T \hat{\boldsymbol{\theta}}_{g_i} \right) x_{(i+1)d}^o \\ &\quad + \left(g_i^o + \mathbf{h}_{g_i}^T \hat{\boldsymbol{\theta}}_{g_i} \right) \left(x_{(i+1)d} - x_{(i+1)d}^o \right) + \left(g_i^o + \mathbf{h}_{g_i}^T \hat{\boldsymbol{\theta}}_{g_i} \right) \left(x_{i+1} - x_{(i+1)d} \right) \\ &= -k_i z_i - s_{i-1} \left(g_{i-1}^o + \mathbf{h}_{g_{i-1}}^T \hat{\boldsymbol{\theta}}_{g_{i-1}} \right) - \left(g_i^o + \mathbf{h}_{g_i}^T \hat{\boldsymbol{\theta}}_{g_i} \right) \zeta_{i+1} + \mathbf{h}_{f_i}^T \tilde{\boldsymbol{\theta}}_{f_i} + \varepsilon_{f_i} + \mathbf{h}_{g_i}^T \tilde{\boldsymbol{\theta}}_{g_i} x_{i+1} \\ &\quad + \varepsilon_{g_i} x_{i+1} + \left(g_i^o + \mathbf{h}_{g_i}^T \hat{\boldsymbol{\theta}}_{g_i} \right) \left(x_{(i+1)d} - x_{(i+1)d}^o \right) + \left(g_i^o + \mathbf{h}_{g_i}^T \hat{\boldsymbol{\theta}}_{g_i} \right) z_{i+1} \\ \dot{z}_i &= -k_i z_i - s_{i-1} \left(g_{i-1}^o + \mathbf{h}_{g_{i-1}}^T \hat{\boldsymbol{\theta}}_{g_{i-1}} \right) + \left(g_i^o + \mathbf{h}_{g_i}^T \hat{\boldsymbol{\theta}}_{g_i} \right) s_{i+1} \\ &\quad + \mathbf{h}_{f_i}^T \tilde{\boldsymbol{\theta}}_{f_i} + \varepsilon_{f_i} + \mathbf{h}_{g_i}^T \tilde{\boldsymbol{\theta}}_{g_i} x_{i+1} + \varepsilon_{g_i} x_{i+1} + \left(g_i^o + \mathbf{h}_{g_i}^T \hat{\boldsymbol{\theta}}_{g_i} \right) \left(x_{(i+1)d} - x_{(i+1)d}^o \right)\end{aligned}$$

Por último, a dinâmica do erro de rastreamento de z_n pode ser escrita como (de forma semelhante a z_i)

$$\dot{z}_n = -k_n z_n - s_{n-1} \left(g_{n-1}^o + \mathbf{h}_{g_{n-1}}^T \hat{\boldsymbol{\theta}}_{g_{n-1}} \right) + \mathbf{h}_{f_n}^T \tilde{\boldsymbol{\theta}}_{f_n} + \varepsilon_{f_n} + \mathbf{h}_{g_n}^T \tilde{\boldsymbol{\theta}}_{g_n} u + \varepsilon_{g_n} u$$

A dinâmica dos erros de trajetória compensados são dados por $\dot{s}_i = \dot{z}_i - \dot{\zeta}_i$ levando a

$$\begin{aligned}\dot{s}_1 &= -k_1 s_1 + \left(g_1^o + \mathbf{h}_{g_1}^T \hat{\boldsymbol{\theta}}_{g_1} \right) s_2 + \mathbf{h}_{f_1}^T \tilde{\boldsymbol{\theta}}_{f_1} + \varepsilon_{f_1} + \mathbf{h}_{g_1}^T \tilde{\boldsymbol{\theta}}_{g_1} x_2 + \varepsilon_{g_1} x_2 \\ \dot{s}_i &= -k_i s_i - s_{i-1} \left(g_{i-1}^o + \mathbf{h}_{g_{i-1}}^T \hat{\boldsymbol{\theta}}_{g_{i-1}} \right) + \left(g_i^o + \mathbf{h}_{g_i}^T \hat{\boldsymbol{\theta}}_{g_i} \right) s_{i+1} \\ &\quad + \mathbf{h}_{f_i}^T \tilde{\boldsymbol{\theta}}_{f_i} + \varepsilon_{f_i} + \mathbf{h}_{g_i}^T \tilde{\boldsymbol{\theta}}_{g_i} x_{i+1} + \varepsilon_{g_i} x_{i+1} \\ \dot{s}_n &= -k_n s_n - s_{n-1} \left(g_{n-1}^o + \mathbf{h}_{g_{n-1}}^T \hat{\boldsymbol{\theta}}_{g_{n-1}} \right) + \mathbf{h}_{f_n}^T \tilde{\boldsymbol{\theta}}_{f_n} + \varepsilon_{f_n} + \mathbf{h}_{g_n}^T \tilde{\boldsymbol{\theta}}_{g_n} u + \varepsilon_{g_n} u\end{aligned}$$

Considerando inicialmente uma função de Lyapunov (apenas para o erro de trajetória compensado) da forma

$$V_s = \frac{1}{2} \sum_{i=1}^n s_i^2$$

sua derivada temporal será dada por

$$\begin{aligned}
\dot{V}_s &= \sum_{i=1}^n s_i \dot{s}_i \\
&= s_1 \left(-k_1 s_1 + \left(g_1^o + \mathbf{h}_{g_1}^T \hat{\boldsymbol{\theta}}_{g_1} \right) s_2 \right) + s_1 \left(\mathbf{h}_{f_1}^T \tilde{\boldsymbol{\theta}}_{f_1} + \mathbf{h}_{g_1}^T \tilde{\boldsymbol{\theta}}_{g_1} x_2 \right) + s_1 \left(\varepsilon_{f_1} + \varepsilon_{g_1} x_2 \right) \\
&+ \sum_{i=2}^{n-1} s_i \left(-k_i s_i - s_{i-1} \left(g_{i-1}^o + \mathbf{h}_{g_{i-1}}^T \hat{\boldsymbol{\theta}}_{g_{i-1}} \right) + \left(g_i^o + \mathbf{h}_{g_i}^T \hat{\boldsymbol{\theta}}_{g_i} \right) s_{i+1} \right) \\
&+ \sum_{i=2}^{n-1} s_i \left(\mathbf{h}_{f_i}^T \tilde{\boldsymbol{\theta}}_{f_i} + \mathbf{h}_{g_i}^T \tilde{\boldsymbol{\theta}}_{g_i} x_{i+1} \right) + \sum_{i=2}^{n-1} s_i \left(\varepsilon_{f_i} + \varepsilon_{g_i} x_{i+1} \right) \\
&+ s_n \left(-k_n s_n - s_{n-1} \left(g_{n-1}^o + \mathbf{h}_{g_{n-1}}^T \hat{\boldsymbol{\theta}}_{g_{n-1}} \right) \right) + s_n \left(\mathbf{h}_{f_n}^T \tilde{\boldsymbol{\theta}}_{f_n} + \mathbf{h}_{g_n}^T \tilde{\boldsymbol{\theta}}_{g_n} u \right) + s_n \left(\varepsilon_{f_n} + \varepsilon_{g_n} u \right)
\end{aligned}$$

Considerando que

$$\varepsilon_{f_i} + \varepsilon_{g_i} x_{i+1} = \delta_i, \quad i = 1, \dots, (n-1)$$

$$\varepsilon_{f_n} + \varepsilon_{g_n} u = \delta_n$$

e simplificando, temos

$$\dot{V}_s = - \sum_{i=1}^n k_i s_i^2 + \sum_{i=1}^n \tilde{\boldsymbol{\theta}}_{f_i}^T \mathbf{h}_{f_i} s_i + \sum_{i=1}^{n-1} \tilde{\boldsymbol{\theta}}_{g_i}^T \mathbf{h}_{g_i} s_i x_{i+1} + \tilde{\boldsymbol{\theta}}_{g_n}^T \mathbf{h}_{g_n} s_n u + \sum_{i=1}^n s_i \delta_i$$

Considerando então a função de Lyapunov, que leva em consideração o erro de adaptação,

$$V = \frac{1}{2} \sum_{i=1}^n \left(s_i^2 + \tilde{\boldsymbol{\theta}}_{f_i}^T \Gamma_{f_i}^{-1} \tilde{\boldsymbol{\theta}}_{f_i} + \tilde{\boldsymbol{\theta}}_{g_i}^T \Gamma_{g_i}^{-1} \tilde{\boldsymbol{\theta}}_{g_i} \right)$$

podemos escolher as leis de adaptação de forma a cancelar os termos envolvendo o erro de adaptação da derivada temporal da função de Lyapunov.

Logo, escolhemos

$$\dot{\hat{\boldsymbol{\theta}}}_{f_i} = \Gamma_{f_i} \mathbf{h}_{f_i} s_i, \quad \forall i$$

$$\dot{\hat{\boldsymbol{\theta}}}_{g_i} = \Gamma_{g_i} \mathbf{h}_{g_i} s_i x_{i+1}, \quad i = 1, \dots, (n-1)$$

$$\dot{\hat{\boldsymbol{\theta}}}_{g_n} = \Gamma_{g_n} \mathbf{h}_{g_n} s_n u$$

o que leva a

$$\dot{V} = - \sum_{i=1}^n k_i s_i^2 + \sum_{i=1}^n s_i \delta_i$$

Como nos outros casos que vimos anteriormente, se o erro de aproximação for nulo $\delta_i = 0$, temos que a derivada da função de Lyapunov é semi-definida negativa e o

sistema é estável em malha fechada. Além disso, utilizando o Lema de Barbalat (de forma bem semelhante à que já utilizamos anteriormente), é possível mostrar também que, neste caso, $\lim_{t \rightarrow \infty} \mathbf{s}(t) = 0$.

Para o caso em que o erro de aproximação não é nulo, temos

$$\begin{aligned}\dot{V} &\leq -\mathbf{s}^T \text{diag}(k_1, \dots, k_n) \mathbf{s} + \mathbf{s}^T \boldsymbol{\delta} \\ \dot{V} &\leq -\min(k_1, \dots, k_n) \|\mathbf{s}\|_2^2 + \|\mathbf{s}\|_2 \|\boldsymbol{\delta}\|_2 \\ \dot{V} &\leq -\kappa \|\mathbf{s}\|_2^2 + \|\mathbf{s}\|_2 \|\boldsymbol{\delta}\|_2\end{aligned}$$

com $\kappa = \min(k_1, \dots, k_n)$.

Como antes, temos então que a derivada temporal da função de Lyapunov é negativa para

$$\begin{aligned}-\kappa \|\mathbf{s}\|_2^2 + \|\mathbf{s}\|_2 \|\boldsymbol{\delta}\|_2 &< 0 \\ \kappa \|\mathbf{s}\|_2^2 &> \|\mathbf{s}\|_2 \|\boldsymbol{\delta}\|_2 \\ \|\mathbf{s}\|_2 &> \frac{\|\boldsymbol{\delta}\|_2}{\kappa}\end{aligned}$$

logo, a trajetória do sistema converge para $\|\mathbf{s}\|_2 \leq \frac{\|\boldsymbol{\delta}\|_2}{\kappa}$. Entretanto, como já vimos, usar o controlador dessa forma leva à *deriva dos parâmetros* dentro dessa região.

Uma forma de se lidar com este problema, é utilizando a mesma abordagem que utilizamos nas outras leis de controle adaptativo e empregar uma zona morta (para evitar a *deriva dos parâmetros*) e uma projeção (para garantir que os parâmetros se mantenham na região desejada).

Dessa forma, a lei de adaptação com as modificação robustas passa a ser dada por

$$\begin{aligned}\dot{\boldsymbol{\theta}}_{f_i} &= \begin{cases} P_{s_{f_i}} (\Gamma_{f_i} \mathbf{h}_{f_i} s_i), & \text{se } \|s\|_2 \geq \mu \\ 0 & \text{caso contrário} \end{cases}, \quad \forall i \\ \dot{\boldsymbol{\theta}}_{g_i} &= \begin{cases} P_{s_{g_i}} (\Gamma_{g_i} \mathbf{h}_{g_i} s_i x_{i+1}), & \text{se } \|s\|_2 \geq \mu \\ 0 & \text{caso contrário} \end{cases}, \quad i = 1, \dots, (n-1) \\ \dot{\boldsymbol{\theta}}_{g_n} &= \begin{cases} P_{s_{g_n}} (\Gamma_{g_n} \mathbf{h}_{g_n} s_n u), & \text{se } \|s\|_2 \geq \mu \\ 0 & \text{caso contrário} \end{cases}\end{aligned}$$

com $\mu > \frac{\|\boldsymbol{\delta}\|_2}{\kappa}$. De forma bem semelhante aos outros casos, é possível se mostrar que com as adaptações dessa forma, garantimos que a trajetória do sistema converge para $\|s\|_2 \leq \mu$ (i.e que μ é um limite final para $\|s\|_2$) e que todos os sinais da malha de controle (erros de trajetória, erros de trajetória compensados, erros de adaptação e parâmetros de adaptação) são limitados.

Aula 31

Controle Adaptativo por Aproximação - Caso discreto

Uma vez que a grande maioria das leis de controle são implementadas de forma digital, é interessante que elas já sejam projetadas em tempo discreto. Dessa forma, nesta aula focaremos nossa atenção para o caso em que a dinâmica do sistema a ser controlado é apresentada em tempo discreto.

Introdução

Enquanto existem diversas formas diferentes de controle adaptativo para sistemas a tempo discreto, nessa aula vamos focar nossa atenção para uma estratégia similar à que usamos para o caso contínuo: vamos derivar nossas leis de adaptação por meio dos erros que aparecem e “atrapalham” o nosso sistema quando escolhemos uma função de Lyapunov.

Para os nossos estudos, vamos focar nossa atenção em sistemas que podem ser descritos por

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B((g_0(\mathbf{x}_k) + g(\mathbf{x}_k))\mathbf{u}_k + f_0(\mathbf{x}_k) + f(\mathbf{x}_k))$$

em que as funções f_0 e g_0 são precisamente conhecidas, enquanto que as funções f e g são desconhecidas.

Assim como no caso contínuo, gostaríamos de substituir as funções desconhecidas por aproximadores universais de modo que

$$\begin{aligned}f(\mathbf{x}) &= \boldsymbol{\theta}_f^T \mathbf{h}_f(\mathbf{x}) + \varepsilon_f(\mathbf{x}) \\g(\mathbf{x}) &= \boldsymbol{\theta}_g^T \mathbf{h}_g(\mathbf{x}) + \varepsilon_g(\mathbf{x})\end{aligned}$$

e fazendo uso de uma lei de controle da forma

$$\mathbf{u}_k = \frac{-f_0(\mathbf{x}_k) - \hat{\boldsymbol{\theta}}_f^T \mathbf{h}_f(\mathbf{x}) + K\mathbf{x}_k}{g_0(\mathbf{x}_k) + \hat{\boldsymbol{\theta}}_g^T \mathbf{h}_g(\mathbf{x})}$$

teríamos uma dinâmica de malha fechada dada por

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B \left(\left(g_0(\mathbf{x}_k) + g(\mathbf{x}_k) + \hat{\boldsymbol{\theta}}_g^T \mathbf{h}_g(\mathbf{x}) - \hat{\boldsymbol{\theta}}_g^T \mathbf{h}_g(\mathbf{x}) \right) \mathbf{u}_k + f_0(\mathbf{x}_k) + f(\mathbf{x}_k) \right)$$

$$\mathbf{x}_{k+1} = (A + BK) \mathbf{x}_k + B \left(\tilde{\boldsymbol{\theta}}_f^T \mathbf{h}_f(\mathbf{x}) + \tilde{\boldsymbol{\theta}}_g^T \mathbf{h}_g(\mathbf{x}) \mathbf{u}_k \right) + B (\varepsilon_f(\mathbf{x}) + \varepsilon_g(\mathbf{x}) \mathbf{u}_k)$$

$$\mathbf{x}_{k+1} = (A + BK) \mathbf{x}_k + B \begin{bmatrix} \tilde{\boldsymbol{\theta}}_f^T & \tilde{\boldsymbol{\theta}}_g^T \end{bmatrix} \begin{bmatrix} \mathbf{h}_f(\mathbf{x}) \\ \mathbf{h}_g(\mathbf{x}) \mathbf{u}_k \end{bmatrix} + B (\varepsilon_f(\mathbf{x}) + \varepsilon_g(\mathbf{x}) \mathbf{u}_k)$$

$$\mathbf{x}_{k+1} = (A + BK) \mathbf{x}_k + B \tilde{\Theta}_k^T \boldsymbol{\varphi}(\mathbf{x}_k, \mathbf{u}_k) + B \delta(\mathbf{x}_k, \mathbf{u}_k)$$

em que $\tilde{\boldsymbol{\theta}}_f = \boldsymbol{\theta}_f - \hat{\boldsymbol{\theta}}_f$, $\tilde{\boldsymbol{\theta}}_g = \boldsymbol{\theta}_g - \hat{\boldsymbol{\theta}}_g$, $\tilde{\Theta}_k^T = \begin{bmatrix} \tilde{\boldsymbol{\theta}}_f^T & \tilde{\boldsymbol{\theta}}_g^T \end{bmatrix}$, $\boldsymbol{\varphi}(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} \mathbf{h}_f(\mathbf{x}) \\ \mathbf{h}_g(\mathbf{x}) \mathbf{u}_k \end{bmatrix}$, $\delta(\mathbf{x}_k, \mathbf{u}_k) = \varepsilon_f(\mathbf{x}) + \varepsilon_g(\mathbf{x}) \mathbf{u}_k$.

Tentando fazer como fazíamos no caso contínuo

Se escolhermos uma função de Lyapunov quadrática

$$V(\mathbf{x}_k) = \mathbf{x}_k^T P \mathbf{x}_k$$

sua diferença ao longo de um intervalo será dada por

$$\begin{aligned} \Delta V &= \mathbf{x}_{k+1}^T P \mathbf{x}_{k+1} - \mathbf{x}_k^T P \mathbf{x}_k \\ &= \left((A + BK) \mathbf{x} + B \tilde{\Theta}^T \boldsymbol{\varphi} + B \delta \right)^T P \left((A + BK) \mathbf{x} + B \tilde{\Theta}^T \boldsymbol{\varphi} + B \delta \right) - \mathbf{x}^T P \mathbf{x} \end{aligned}$$

Note que temos aqui duas grandes diferenças em relação ao caso contínuo:

- O termo relativo ao erro de estimação na aparece de forma linear na diferença da função de Lyapunov
- Ao escrevermos uma lei de adaptação para os parâmetros, é difícil escrevê-la de modo que a diferença da sua função de Lyapunov seja linear no erro de estimação

Função de Lyapunov logarítmica

De modo a contornarmos de certa forma este problema, consideraremos uma função de Lyapunov (para os estados do sistema) da forma

$$V(\mathbf{x}_k) = \ln \left(1 + \mathbf{x}_k^T P \mathbf{x}_k \right)$$

que nos permitirá escolher uma estrutura interessante para a lei de adaptação um pouco mais a frente.

Com uma função de Lyapunov dessa forma, a diferença da função de Lyapunov será dada por

$$\begin{aligned}
 \Delta V &= \ln \left(1 + \mathbf{x}_{k+1}^T P \mathbf{x}_{k+1} \right) - \ln \left(1 + \mathbf{x}_k^T P \mathbf{x}_k \right) \\
 &= \ln \left(\frac{1 + \mathbf{x}_{k+1}^T P \mathbf{x}_{k+1}}{1 + \mathbf{x}_k^T P \mathbf{x}_k} \right) \\
 &= \ln \left(\frac{1 + \mathbf{x}_k^T P \mathbf{x}_k - \mathbf{x}_k^T P \mathbf{x}_k + \mathbf{x}_{k+1}^T P \mathbf{x}_{k+1}}{1 + \mathbf{x}_k^T P \mathbf{x}_k} \right) \\
 &= \ln \left(1 + \frac{\mathbf{x}_{k+1}^T P \mathbf{x}_{k+1} - \mathbf{x}_k^T P \mathbf{x}_k}{1 + \mathbf{x}_k^T P \mathbf{x}_k} \right) \\
 &\leq \frac{\mathbf{x}_{k+1}^T P \mathbf{x}_{k+1} - \mathbf{x}_k^T P \mathbf{x}_k}{1 + \mathbf{x}_k^T P \mathbf{x}_k}
 \end{aligned}$$

em que, na última desigualdade, utilizamos a desigualdade $\ln(1+a) \leq a$, $\forall a > -1$.

Substituindo a dinâmica do sistema, passamos a ter

$$\Delta V \leq \frac{((A+BK)\mathbf{x} + B\tilde{\Theta}^T\boldsymbol{\varphi} + B\delta)^T P ((A+BK)\mathbf{x} + B\tilde{\Theta}^T\boldsymbol{\varphi} + B\delta) - \mathbf{x}^T P \mathbf{x}}{1 + \mathbf{x}^T P \mathbf{x}}$$

Supondo que conseguimos encontrar um ganho K tal que

$$\begin{bmatrix} x \\ B\delta \\ B\tilde{\Theta}^T\boldsymbol{\varphi} \end{bmatrix}^T \left(\begin{bmatrix} -(1-\eta)P & 0 & 0 \\ 0 & -\gamma I & 0 \\ 0 & 0 & -\gamma I \end{bmatrix} + \begin{bmatrix} A^T + K^T B^T \\ I \\ I \end{bmatrix} P \begin{bmatrix} A+BK & I & I \end{bmatrix} \right) \begin{bmatrix} x \\ B\delta \\ B\tilde{\Theta}^T\boldsymbol{\varphi} \end{bmatrix} < 0$$

temos que a diferença da função de Lyapunov será dada por

$$\Delta V \leq -\eta \frac{\mathbf{x}_k^T P \mathbf{x}_k}{1 + \mathbf{x}^T P \mathbf{x}} + \gamma \frac{\boldsymbol{\varphi}^T \tilde{\Theta} B^T P B \tilde{\Theta}^T \boldsymbol{\varphi}}{1 + \mathbf{x}^T P \mathbf{x}} + \gamma \frac{\delta^T B^T P B \delta}{1 + \mathbf{x}^T P \mathbf{x}}$$

Tal ganho pode ser encontrado minimizando γ para um valor de η dado sujeito à desigualdade matricial linear

$$\begin{bmatrix} -(1-\eta)Q & 0 & 0 & QA^T + Y^T B^T \\ * & -\gamma I & 0 & I \\ * & * & -\gamma I & I \\ * & * & * & -Q \end{bmatrix} < 0$$

e temos $P = Q^{-1}$, $K = YQ^{-1}$.

Lei de adaptação

Se escolhermos a lei de adaptação

$$\hat{\Theta}_{k+1} = \hat{\Theta}_k + \frac{\boldsymbol{\varphi}_k (\mathbf{x}_{k+1} - (A + BK) \mathbf{x}_k)^T B (B^T B)^{-1}}{1 + \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k},$$

supondo que o posto de colunas de B é completo (logo $\exists (B^T B)^{-1}$), considerando que $\mathbf{x}_{k+1} - (A + BK) \mathbf{x}_k = B \tilde{\Theta}_k^T \boldsymbol{\varphi}_k + B \delta_k$, temos que a dinâmica do erro de adaptação será dada por

$$\tilde{\Theta}_{k+1} = \tilde{\Theta}_k - \frac{\boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \tilde{\Theta}_k}{1 + \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} - \frac{\boldsymbol{\varphi}_k \delta_k^T}{1 + \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k}$$

Escolhendo a função de Lyapunov

$$V_2(\tilde{\Theta}_k) = c\gamma \operatorname{tr} \left(\tilde{\Theta}_k B^T P B \tilde{\Theta}_k^T \right)$$

temos que a sua diferença será dada por

$$\begin{aligned} \Delta V_2 &= c\gamma \left(\operatorname{tr} \left(\tilde{\Theta}_k B^T P B \tilde{\Theta}_k^T \right) - 2 \frac{\operatorname{tr} \left(\tilde{\Theta}_k B^T P B \tilde{\Theta}_k^T \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \right)}{1 + \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} + 2 \frac{\operatorname{tr} \left(\boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \tilde{\Theta}_k B^T P B \delta_k \boldsymbol{\varphi}_k^T \right)}{(1 + \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k)^2} \right. \\ &\quad + \frac{\operatorname{tr} \left(\boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \tilde{\Theta}_k B^T P B \tilde{\Theta}_k^T \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \right)}{(1 + \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k)^2} - \operatorname{tr} \left(\tilde{\Theta}_k B^T P B \tilde{\Theta}_k^T \right) - 2 \frac{\operatorname{tr} \left(\tilde{\Theta}_k B^T P B \delta_k \boldsymbol{\varphi}_k^T \right)}{1 + \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} \\ &\quad \left. + \frac{\operatorname{tr} \left(\boldsymbol{\varphi}_k \delta_k^T B^T P B \delta_k \boldsymbol{\varphi}_k^T \right)}{(1 + \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k)^2} \right) \\ &\leq c\gamma \left(- \frac{\boldsymbol{\varphi}_k^T \tilde{\Theta}_k B^T P B \tilde{\Theta}_k^T \boldsymbol{\varphi}_k}{1 + \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} + \frac{\delta_k^T B^T P B \delta_k}{1 + \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} \right) \end{aligned}$$

Usando o fato que

$$\boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k = \mathbf{h}_f^T \mathbf{h}_f + \mathbf{u}_k^T \mathbf{h}_g^T \mathbf{h}_g \mathbf{u}_k \leq 1 + \mathbf{u}_k^T \mathbf{u}_k$$

temos que

$$\Delta V_2 \leq c\gamma \left(- \frac{\boldsymbol{\varphi}_k^T \tilde{\Theta}_k B^T P B \tilde{\Theta}_k^T \boldsymbol{\varphi}_k}{2 + \mathbf{u}_k^T \mathbf{u}_k + \mathbf{x}_k^T P \mathbf{x}_k} + \frac{\delta_k^T B^T P B \delta_k}{2 + \mathbf{u}_k^T \mathbf{u}_k + \mathbf{x}_k^T P \mathbf{x}_k} \right)$$

Escolhendo c tal que

$$\begin{aligned} \frac{c}{2 + \mathbf{u}_k^T \mathbf{u}_k + \mathbf{x}_k^T P \mathbf{x}_k} &\geq \frac{1}{1 + \mathbf{x}_k^T P \mathbf{x}_k} \\ c &\geq \frac{2 + \mathbf{u}_k^T \mathbf{u}_k + \mathbf{x}_k^T P \mathbf{x}_k}{1 + \mathbf{x}_k^T P \mathbf{x}_k} = 1 + \frac{1 + \mathbf{u}_k^T \mathbf{u}_k}{1 + \mathbf{x}_k^T P \mathbf{x}_k} \end{aligned}$$

temos que

$$\Delta V_2 \leq \gamma \left(- \frac{\boldsymbol{\varphi}_k^T \tilde{\Theta}_k B^T P B \tilde{\Theta}_k^T \boldsymbol{\varphi}_k}{1 + \mathbf{x}_k^T P \mathbf{x}_k} + \frac{\delta_k^T B^T P B \delta_k}{1 + \mathbf{x}_k^T P \mathbf{x}_k} \right)$$

e para isso, basta escolher $c \geq 2 + \max(\mathbf{u}_k^T \mathbf{u}_k)$, com $\max(\mathbf{u}_k^T \mathbf{u}_k)$ sendo o maior valor aceitado para $\mathbf{u}_k^T \mathbf{u}_k$.

Lidando com o erro de aproximação

Se o erro de aproximação for nulo, $\delta_k = 0$, a abordagem apresentada até o momento leva à

$$\Delta V + \Delta V_2 \leq -\eta \frac{\mathbf{x}_k^T P \mathbf{x}_k}{1 + \mathbf{x}_k^T P \mathbf{x}_k} \leq 0$$

que é semidefinida negativa, garantindo apenas a estabilidade em malha fechada do sistema de controle adaptativo. Entretanto, utilizando o Princípio da Invariância de LaSalle, é possível mostrar que os estados do sistema convergem para a origem (enquanto o erro de adaptação se mantém limitado devido à estabilidade do sistema).

No caso em que o erro de aproximação não é nulo, entretanto, a diferença da função de Lyapunov é dada por

$$\Delta V + \Delta V_2 \leq -\eta \frac{\mathbf{x}_k^T P \mathbf{x}_k}{1 + \mathbf{x}_k^T P \mathbf{x}_k} + 2\gamma \frac{\delta^T B^T P B \delta}{1 + \mathbf{x}_k^T P \mathbf{x}_k}$$

e não conseguimos afirmar muito em relação ao erro em malha fechada (note que, diferentemente de um sistema contínuo, é difícil determinar a região para a qual o sistema iria convergir neste caso). O uso das modificações de zona morta e projeção utilizadas para sistemas contínuos é mais complicado para sistemas discreto e por isso, no caso desta aula, vamos fazer uso de outra modificação robusta.

Conhecida como *modificação- σ* , modificamos a lei de adaptação para que ela tenha um comportamento que tende a um conjunto padrão de parâmetros na ausência de sinais de excitação.

Com essa modificação, considerando um conjunto padrão de parâmetros Θ_o , e um escalar que define a velocidade de convergência para os parâmetros padrão, $\sigma \in [0, 1]$, nossa lei de adaptação passa a ser escrita como

$$\hat{\Theta}_{k+1} = \hat{\Theta}_k + \frac{\boldsymbol{\varphi}_k (\mathbf{x}_{k+1} - (A + BK) \mathbf{x}_k)^T B (B^T B)^{-1}}{1 + \frac{1}{\sigma} \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} + (1 - \sigma) (\Theta_o - \hat{\Theta}_k)$$

de modo que a dinâmica do erro de adaptação passa a ser dada por

$$\tilde{\Theta}_{k+1} = \tilde{\Theta}_k - \frac{\boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \tilde{\Theta}_k}{1 + \frac{1}{\sigma} \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} - \frac{\boldsymbol{\varphi}_k \delta_k^T}{1 + \frac{1}{\sigma} \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} - (1 - \sigma) (\Theta_o - \hat{\Theta}_k) + (1 - \sigma) (\Theta - \Theta)$$

$$\tilde{\Theta}_{k+1} = \sigma \tilde{\Theta}_k - \frac{\boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \tilde{\Theta}_k}{1 + \frac{1}{\sigma} \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} - \frac{\boldsymbol{\varphi}_k \delta_k^T}{1 + \frac{1}{\sigma} \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} - (1 - \sigma) \Theta_d$$

com $\Theta_d = \Theta_o - \Theta$. Escolhendo a função de Lyapunov como

$$V_2(\tilde{\Theta}_k) = c\gamma \text{tr} \left(\tilde{\Theta}_k B^T P B \tilde{\Theta}_k^T \right)$$

sua diferença da função de Lyapunov será dada por

$$\begin{aligned} \Delta V_2 \leq & c\gamma \left(- (1 - \sigma^2) \operatorname{tr} \left(\tilde{\Theta}_k B^T P B \tilde{\Theta}_k^T \right) - \sigma \frac{\boldsymbol{\varphi}_k^T \tilde{\Theta}_k B^T P B \tilde{\Theta}_k^T \boldsymbol{\varphi}_k}{1 + \frac{1}{\sigma} \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} + \sigma \frac{\delta_k^T B^T P B \delta_k}{1 + \frac{1}{\sigma} \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} \right. \\ & + (1 - \sigma)^2 \operatorname{tr} \left(\Theta_d B^T P B \Theta_d^T \right) - 2\sigma(1 - \sigma) \operatorname{tr} \left(\tilde{\Theta}_k B^T P B \Theta_d^T \right) \\ & \left. + 2(1 - \sigma) \frac{\operatorname{tr} \left(\boldsymbol{\varphi} \boldsymbol{\varphi}^T \tilde{\Theta}_k B^T P B \Theta_d^T \right)}{1 + \frac{1}{\sigma} \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} + 2(1 - \sigma) \frac{\operatorname{tr} \left(\Theta_d B^T P B \delta_k \boldsymbol{\varphi}_k^T \right)}{1 + \frac{1}{\sigma} \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} \right) \end{aligned}$$

Usando o fato de que, com $\zeta > 0$ e $\beta > 0$,

$$\begin{aligned} 2(1 - \sigma) \frac{\operatorname{tr} \left(\Theta_d B^T P B \delta_k \boldsymbol{\varphi}_k^T \right)}{1 + \frac{1}{\sigma} \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} & \leq (1 - \sigma) \left(\zeta \operatorname{tr} \left(\Theta_d B^T P B \Theta_d^T \right) + \frac{\sigma}{\zeta} \frac{\delta_k^T B^T P B \delta_k}{1 + \frac{1}{\sigma} \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} \right), \\ 2(1 - \sigma) \frac{\operatorname{tr} \left(\boldsymbol{\varphi} \boldsymbol{\varphi}^T \tilde{\Theta}_k B^T P B \Theta_d^T \right)}{1 + \frac{1}{\sigma} \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} & \leq (1 - \sigma) \left(\beta \operatorname{tr} \left(\Theta_d B^T P B \Theta_d^T \right) + \frac{\sigma}{\beta} \frac{\boldsymbol{\varphi}^T \tilde{\Theta}_k B^T P B \tilde{\Theta}_k^T \boldsymbol{\varphi}}{1 + \frac{1}{\sigma} \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} \right) \end{aligned}$$

e que, para $v > 0$,

$$-2\sigma(1 - \sigma) \operatorname{tr} \left(\tilde{\Theta}_k B^T P B \Theta_d^T \right) \leq v(1 - \sigma)^2 \operatorname{tr} \left(\Theta_d B^T P B \Theta_d^T \right) + \frac{1}{v} \sigma^2 \operatorname{tr} \left(\tilde{\Theta}_k B^T P B \tilde{\Theta}_k^T \right)$$

temos que a diferença da função de Lyapunov pode ser limitada por

$$\begin{aligned} \Delta V_2 \leq & - \left(1 - \frac{v+1}{v} \sigma^2 \right) V_{2k} + c\gamma \left(- \left(\frac{(\beta-1)\sigma + \sigma^2}{\beta} \right) \frac{\boldsymbol{\varphi}_k^T \tilde{\Theta}_k B^T P B \tilde{\Theta}_k^T \boldsymbol{\varphi}_k}{1 + \frac{1}{\sigma} \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} \right. \\ & + \left((1+v)(1-\sigma)^2 + (1-\sigma)(\zeta + \beta) \right) \operatorname{tr} \left(\Theta_d B^T P B \Theta_d^T \right) \\ & \left. + \left(\frac{\sigma(\zeta+1) - \sigma^2}{\zeta} \right) \frac{\delta_k^T B^T P B \delta_k}{1 + \frac{1}{\sigma} \boldsymbol{\varphi}_k^T \boldsymbol{\varphi}_k + \mathbf{x}_k^T P \mathbf{x}_k} \right) \end{aligned}$$

Considerando que $\beta > 1$, podemos escolher c tal que

$$\begin{aligned} \frac{c \frac{(\beta-1)\sigma + \sigma^2}{\beta}}{1 + \frac{1 + \mathbf{u}_k^T \mathbf{u}_k}{\sigma} + \mathbf{x}_k^T P \mathbf{x}_k} & \geq \frac{1}{1 + \mathbf{x}_k^T P \mathbf{x}_k} \\ c & \geq \frac{\beta}{(\beta-1)\sigma + \sigma^2} \left(1 + \frac{1 + \mathbf{u}_k^T \mathbf{u}_k}{\sigma (1 + \mathbf{x}_k^T P \mathbf{x}_k)} \right) \end{aligned}$$

e dessa forma, escolhemos $c = \frac{\beta}{(\beta-1)\sigma + \sigma^2} \left(1 + \frac{1 + \max(\mathbf{u}_k^T \mathbf{u}_k)}{\sigma} \right)$, e a diferença das duas funções de Lyapunov juntas pode ser escrita como

$$\Delta V + \Delta V_2 \leq -\eta \frac{\mathbf{x}_k^T P \mathbf{x}_k}{1 + \mathbf{x}_k^T P \mathbf{x}_k} - \left(1 - \frac{v+1}{v} \sigma^2 \right) V_{2k} + d$$

com

$$d \geq \gamma \left(1 + \frac{\sigma(\zeta + 1) - \sigma^2}{\zeta} \right) \delta^T B^T P B \delta + c\gamma \left((1 + \nu)(1 - \sigma)^2 + (1 - \sigma)(\zeta + \beta) \right) \text{tr} \left(\Theta_d B^T P B \Theta_d^T \right)$$

Usando o fato de que $\ln(1 + a) \geq \frac{a}{1+a}$, $a > -1$, podemos escrever

$$\Delta V + \Delta V_2 \leq -\eta V_k - (1 - \sigma^2) V_{2_k} + d$$

o que nos leva a

$$V_{k+1} + V_{2_{k+1}} \leq (1 - \eta) V_k + \sigma^2 V_{2_k} + d$$

$$V_{k+1} + V_{2_{k+1}} \leq \zeta (V_k + V_{2_k}) + d$$

com $\zeta = \max(1 - \eta, \sigma^2)$.

Essa última desigualdade implica em

$$V_k + V_{2_k} \leq \zeta^k (V_0 + V_{2_0}) + \frac{1 - \zeta^k}{1 - \zeta} d$$

e nos permite obter o limite final

$$\ln(1 + \lambda_{\min} \|\mathbf{x}_k\|^2) \leq \frac{d}{1 - \zeta}$$

$$\|\mathbf{x}_k\| \leq \sqrt{\frac{d}{\lambda_{\min} (e^{\frac{d}{1 - \zeta}} - 1)}}$$

Aula 32

Controle Adaptativo por aproximação - Mínimos Quadrados Recursivo

Em muitos casos, é muito comum utilizarmos uma lei de adaptação dada pela utilização do método de mínimos quadrados recursivo. Apesar de ser uma lei de adaptação mais custosa (por demandar mais estados), ela normalmente tem resultados (em termos de adaptação dos parâmetros) melhores.

Vamos focar nossa atenção apenas no uso da lei de adaptação proposta, e em alguns problemas associados à ela.

Lei de adaptação - Mínimos Quadrados Recursivo

Considerando uma lei de adaptação da forma (Kanellakopoulos 1994)

$$P_{k+1} = P_k - \frac{\zeta P_k \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T P_k}{1 + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k}$$
$$\hat{\Theta}_{k+1} = \hat{\Theta}_k + \zeta P_{k+1} \boldsymbol{\varphi}_k (\mathbf{x}_{k+1} - (A + BK) \mathbf{x}_k)^T B (B^T B)^{-1}$$

com $\zeta > 0$ um parâmetro de projeto. Dessa forma, a dinâmica do erro de adaptação será dada da forma

$$\tilde{\Theta}_{k+1} = \tilde{\Theta}_k - \zeta P_{k+1} \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \tilde{\Theta}_k - \zeta P_{k+1} \boldsymbol{\varphi}_k \delta_k^T$$

Usando o Lema da Inversão, podemos escrever que

$$\begin{aligned} P_{k+1} &= P_k - \frac{\zeta P_k \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T P_k}{1 + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k} \\ P_{k+1} &= P_k - \zeta P_k \boldsymbol{\varphi}_k \left(1 + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k\right)^{-1} \boldsymbol{\varphi}_k^T P_k \\ P_{k+1}^{-1} &= P_k^{-1} + \zeta \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \end{aligned}$$

Considerando uma função de Lyapunov para a adaptação da forma

$$\begin{aligned} V_\theta &= \text{tr} \left(\tilde{\Theta}_k^T P_k^{-1} \tilde{\Theta}_k \right) \\ \Delta V_\theta &= \text{tr} \left(\tilde{\Theta}_{k+1}^T P_{k+1}^{-1} \tilde{\Theta}_{k+1} \right) - \text{tr} \left(\tilde{\Theta}_k^T P_k^{-1} \tilde{\Theta}_k \right) \\ \Delta V_\theta &= \text{tr} \left(\tilde{\Theta}_k^T P_{k+1}^{-1} \tilde{\Theta}_k \right) - 2\zeta \text{tr} \left(\tilde{\Theta}_k^T \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \right) - 2\zeta \text{tr} \left(\tilde{\Theta}_k^T \boldsymbol{\varphi}_k \delta_k^T \right) \\ &\quad + \zeta^2 \text{tr} \left(\tilde{\Theta}_k^T \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \right) + 2\zeta^2 \text{tr} \left(\tilde{\Theta}_k^T \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \delta_k^T \right) \\ &\quad + \zeta^2 \text{tr} \left(\delta_k \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \delta_k^T \right) - \text{tr} \left(\tilde{\Theta}_k^T P_k^{-1} \tilde{\Theta}_k \right) \\ \Delta V_\theta &= \left(-2\zeta + \zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \right) \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \tilde{\Theta}_k^T \boldsymbol{\varphi}_k + \left(-2\zeta + 2\zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \right) \delta_k^T \tilde{\Theta}_k^T \boldsymbol{\varphi}_k \\ &\quad + \zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \delta_k^T \delta_k + \text{tr} \left(\tilde{\Theta}_k^T \left(P_{k+1}^{-1} - P_k^{-1} \right) \tilde{\Theta}_k \right) \\ \Delta V_\theta &= \left(-2\zeta + \zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \right) \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \tilde{\Theta}_k^T \boldsymbol{\varphi}_k + \left(-2\zeta + 2\zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \right) \delta_k^T \tilde{\Theta}_k^T \boldsymbol{\varphi}_k \\ &\quad + \zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \delta_k^T \delta_k + \text{tr} \left(\tilde{\Theta}_k^T \left(P_{k+1}^{-1} - P_k^{-1} \right) \tilde{\Theta}_k \right) \\ \Delta V_\theta &= \left(-\zeta + \zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \right) \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \tilde{\Theta}_k^T \boldsymbol{\varphi}_k + \left(-2\zeta + 2\zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \right) \delta_k^T \tilde{\Theta}_k^T \boldsymbol{\varphi}_k \\ &\quad + \zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \delta_k^T \delta_k + \text{tr} \left(\tilde{\Theta}_k^T \left(P_{k+1}^{-1} - P_k^{-1} \right) \tilde{\Theta}_k \right) \end{aligned}$$

Note que

$$\begin{aligned} \zeta \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k &= \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k - \frac{\zeta^2 \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k}{1 + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k} = \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k - \frac{(\zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k)^2}{1 + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k} \\ &= \frac{\zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k + (\zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k)^2 - (\zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k)^2}{1 + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k} \\ &= \frac{\zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k}{1 + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k} \end{aligned}$$

De modo que

$$-1 + \zeta \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k = -1 + \frac{\zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k}{1 + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k} = \frac{-1 - \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k}{1 + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k} = \frac{-1}{1 + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k}$$

e podemos escrever que

$$\Delta V_\theta = \frac{-\zeta \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \tilde{\Theta}_k^T \boldsymbol{\varphi}_k - 2\zeta \delta_k^T \tilde{\Theta}_k^T \boldsymbol{\varphi}_k + \zeta^2 \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k \delta_k^T \delta_k}{1 + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k}$$

Consideramos também uma função de Lyapunov para a matriz de covariância

$$\begin{aligned} V_P &= \text{tr}(P_k) \\ \Delta V_P &= \text{tr}(P_{k+1} - P_k) \\ \Delta V_P &= -\text{tr}\left(\frac{\zeta P_k \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T P_k}{1 + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k}\right) \\ \Delta V_P &= -\zeta \frac{\boldsymbol{\varphi}_k^T P_k P_k \boldsymbol{\varphi}_k}{1 + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k} \end{aligned}$$

Lidando com o erro de aproximação: Fator de esquecimento

Mesmo que o projeto da lei de controle em conjunto com a lei de adaptação apresentada fosse capaz de garantir que os estados do sistema convergem para a origem, como vimos ao longo da disciplina, se não lidarmos com o efeito do erro de aproximação estaremos sujeitos à *deriva dos parâmetros*.

Vamos considerar uma modificação padrão do mínimos quadrados recursivo em que consideramos um fator de esquecimento.

Dessa forma, considerando um fator de esquecimento $\alpha \in [0, 1]$ nossa lei de adaptação agora é dada por

$$\begin{aligned} P_{k+1} &= \frac{1}{\alpha} \left(P_k - \frac{\zeta P_k \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T P_k}{\alpha + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k} \right) \\ \hat{\Theta}_{k+1} &= \hat{\Theta}_k + \zeta P_{k+1} \boldsymbol{\varphi}_k (\mathbf{x}_{k+1} - (A + BK) \mathbf{x}_k)^T B (B^T B)^{-1} \end{aligned}$$

Usando o Lema da Inversão, podemos escrever que

$$\begin{aligned} P_{k+1} &= \frac{1}{\alpha} \left(P_k - \frac{\zeta P_k \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T P_k}{\alpha + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k} \right) \\ P_{k+1} &= \frac{1}{\alpha} P_k - \zeta \frac{1}{\alpha} P_k \boldsymbol{\varphi}_k \left(1 + \zeta \boldsymbol{\varphi}_k^T \frac{1}{\alpha} P_k \boldsymbol{\varphi}_k \right)^{-1} \boldsymbol{\varphi}_k^T P_k \frac{1}{\alpha} \\ P_{k+1}^{-1} &= \alpha P_k^{-1} + \zeta \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \end{aligned}$$

A dinâmica do erro de adaptação é dada por

$$\tilde{\Theta}_{k+1} = \tilde{\Theta}_k - \zeta P_{k+1} \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \tilde{\Theta}_k - \zeta P_{k+1} \boldsymbol{\varphi}_k \delta_k^T$$

E a diferença da função de Lyapunov passa a ser dada por

$$\begin{aligned}
\Delta V_\theta &= \text{tr} \left(\tilde{\Theta}_{k+1}^T P_{k+1}^{-1} \tilde{\Theta}_{k+1} \right) - \text{tr} \left(\tilde{\Theta}_k^T P_k^{-1} \tilde{\Theta}_k \right) \\
\Delta V_\theta &= \text{tr} \left(\tilde{\Theta}_k^T P_{k+1}^{-1} \tilde{\Theta}_k \right) - 2\zeta \text{tr} \left(\tilde{\Theta}_k^T \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \right) - 2\zeta \text{tr} \left(\tilde{\Theta}_k^T \boldsymbol{\varphi}_k \delta_k^T \right) \\
&\quad + \zeta^2 \text{tr} \left(\tilde{\Theta}_k^T \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \right) + 2\zeta^2 \text{tr} \left(\tilde{\Theta}_k^T \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \delta_k^T \right) \\
&\quad + \zeta^2 \text{tr} \left(\delta_k \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \delta_k^T \right) - \text{tr} \left(\tilde{\Theta}_k^T P_k^{-1} \tilde{\Theta}_k \right) \\
\Delta V_\theta &= \left(-2\zeta + \zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \right) \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \tilde{\Theta}_k^T \boldsymbol{\varphi}_k + \left(-2\zeta + 2\zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \right) \delta_k^T \tilde{\Theta}_k^T \boldsymbol{\varphi}_k \\
&\quad + \zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \delta_k^T \delta_k + \text{tr} \left(\tilde{\Theta}_k^T \left(P_{k+1}^{-1} - \alpha P_k^{-1} \right) \tilde{\Theta}_k \right) - (1 - \alpha) \text{tr} \left(\tilde{\Theta}_k^T P_k^{-1} \tilde{\Theta}_k \right) \\
\Delta V_\theta &= \left(-2\zeta + \zeta + \zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \right) \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \tilde{\Theta}_k^T \boldsymbol{\varphi}_k + \left(-2\zeta + 2\zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \right) \delta_k^T \tilde{\Theta}_k^T \boldsymbol{\varphi}_k \\
&\quad + \left(\zeta - \zeta + \zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \right) \delta_k^T \delta_k - (1 - \alpha) V_\theta
\end{aligned}$$

Uma vez que

$$\begin{aligned}
\zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k &= \frac{1}{\alpha} \left(\zeta^2 \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k - \frac{\zeta^3 (\boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k)^2}{\alpha + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k} \right) \\
&= \frac{\zeta^2 \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k}{\alpha + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k}
\end{aligned}$$

temos que

$$-\zeta + \zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k = -\zeta + \frac{\zeta^2 \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k}{\alpha + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k} = \frac{-\alpha \zeta}{\alpha + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k}$$

de modo que

$$\Delta V_\theta = \frac{-\alpha \zeta \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \tilde{\Theta}_k^T \boldsymbol{\varphi}_k - 2\alpha \zeta \delta_k^T \tilde{\Theta}_k^T \boldsymbol{\varphi}_k - \alpha \zeta \delta_k^T \delta_k}{\alpha + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k} + \zeta \delta_k^T \delta_k - (1 - \alpha) V_\theta$$

Considerando que, para $\mu > 0$, temos

$$-2\alpha \delta_k^T \tilde{\Theta}_k^T \boldsymbol{\varphi}_k \leq \mu \alpha^2 \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \tilde{\Theta}_k^T \boldsymbol{\varphi}_k + \frac{\alpha^2}{\mu} \delta_k^T \delta_k$$

e podemos escrever

$$\Delta V_\theta \leq \frac{-\zeta (\alpha - \mu \alpha^2) \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \tilde{\Theta}_k^T \boldsymbol{\varphi}_k}{\alpha + \zeta \boldsymbol{\varphi}_k^T P_k \boldsymbol{\varphi}_k} + \left(\zeta + \frac{\alpha}{\mu} \right) \delta_k^T \delta_k - (1 - \alpha) V_\theta$$

Esta análise separadamente nos levaria a dizer que o uso do fator de esquecimento leva a um efeito robusto na lei de adaptação. Entretanto, note que a função de Lyapunov do erro dos parâmetros é definida de acordo com a inversa da função de Lyapunov. Analisando então a dinâmica da inversa da matriz de covariância, temos

$$P_{k+1}^{-1} = \alpha P_k^{-1} + \zeta \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T$$

e temos um problema. Como a dinâmica agora é estável (ao invés de ser puramente um integrador em relação à $\zeta \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T$), é necessário que o vetor de regressores $\boldsymbol{\varphi}_k$ satisfaça uma condição de *persistência de excitação*, ou seja, que

$$\sum_{s=0}^N \boldsymbol{\varphi}_{k-s} \boldsymbol{\varphi}_{k-s}^T \geq \nu I$$

para algum N e ν para garantir que P_k^{-1} não tenda a zero em algumas direções (e portanto que P_k não tenda a infinito em certas direções).

No caso do controle adaptativo de sistemas lineares, este problema pode ser contornado adicionando-se um pequeno ruído branco à ação de controle de modo a garantir a *persistência de excitação* dos regressores (aceitando o efeito que tal ruído vai ter sobre o sistema controlado). No nosso caso, tal abordagem muitas vezes não garante a persistência de excitação pois seria necessário que o sistema tivesse sempre uma grande excursão sobre o universo de discurso (uma vez que nossas funções de pertinência quase sempre tem suportes limitados) o que é inviável na prática.

Regularização da Covariância

Uma forma de se lidar com este problema é por meio de uma regularização da inversa da matriz de covariância. Com o fator de esquecimento, a dinâmica da inversa da matriz de covariância era dada por

$$P_{k+1}^{-1} = \alpha P_k^{-1} + \zeta \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T$$

com $\alpha \in (0, 1]$. E nos casos em que $\alpha < 1$, se $\zeta \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T$ não variar bastante dentro do nosso universo de discurso, corremos o risco de $P_k^{-1} \rightarrow 0$ em certas direções (o que implica que $P_k \rightarrow \infty$ em certas direções).

A estratégia normalmente utilizada para a regularização da matriz de covariância pode ser um pouco custosa computacionalmente. Dessa forma, vamos considerar uma estratégia um pouco diferente. Consideramos a existência de um pequeno termo variante no tempo adicional na dinâmica da inversa da matriz de covariância

$$P_{k+1}^{-1} = \alpha P_k^{-1} + \lambda \boldsymbol{\eta}_k \boldsymbol{\eta}_k^T + \zeta \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T$$

com $\lambda > 0$, e $\|\boldsymbol{\eta}_k\| = 1$ utilizados para evitar que a dinâmica tenda a zero em certas direções. Os vetores de regularização, $\boldsymbol{\eta}_k$ podem ser escolhidos de várias formas distintas, desde que variem de modo que garantam que a combinação dos $\boldsymbol{\eta}_k \boldsymbol{\eta}_k^T$ gere uma matriz quadrada de posto completo. Considerando que $P_k^{-1} \in \mathbb{R}^{r \times r}$, e os vetores \mathbf{e}_ℓ base do espaço \mathbb{R}^r definidos como

$$\mathbf{e}_{\ell i} = \begin{cases} 1, & \text{se } i = \ell \\ 0, & \text{se } i \neq \ell \end{cases}$$

vamos considerar que η_k é um sinal periódico, de modo a simplificar a implementação (e nossa análise), definido como

$$\eta_k = \mathbf{e}_{\text{mod}(k,r)+1}$$

Ignorando o efeito de φ_k sobre a dinâmica da inversa da matriz de covariância, podemos escrever (considerando um período do sinal η_k)

$$P_k^{-1} = \alpha^r P_{k-r}^{-1} + \lambda \left(\eta_{k-1} \eta_{k-1}^T + \alpha \eta_{k-2} \eta_{k-2}^T + \cdots + \alpha^{r-1} \eta_k \eta_k^T \right)$$

Como

$$\sum_{k=1}^r \eta_k \eta_k^T = I$$

temos que

$$P_k^{-1} \geq (\lambda \alpha^{r-1}) I$$

$$P_k \leq \frac{1}{\lambda \alpha^{r-1}} I$$

Logo, podemos ver que, apesar de não ser a estratégia usualmente utilizada para a regularização da matriz de covariância, a estratégia proposta nesta aula é capaz, da mesma forma, em garantir que a matriz de covariância seja sempre limitada.

O interessante desta proposta, é que ainda conseguimos equações de atualização da matriz de covariância que não necessitam da operação de inversão de matrizes. Utilizando o Lema da Inversão, temos que

$$P_{k+1} = \left(\alpha P_k^{-1} + \lambda \eta_k \eta_k^T \right)^{-1} - \frac{\zeta \left(\alpha P_k^{-1} + \lambda \eta_k \eta_k^T \right)^{-1} \varphi_k \varphi_k^T \left(\alpha P_k^{-1} + \lambda \eta_k \eta_k^T \right)^{-1}}{1 + \zeta \varphi_k^T \left(\alpha P_k^{-1} + \lambda \eta_k \eta_k^T \right)^{-1} \varphi_k}$$

e, uma vez que

$$\begin{aligned} \left(\alpha P_k^{-1} + \lambda \eta_k \eta_k^T \right)^{-1} &= \frac{1}{\alpha} P_k - \frac{1}{\alpha} P_k \eta_k \left(\frac{1}{\lambda} + \frac{1}{\alpha} \eta_k^T P_k \eta_k \right)^{-1} \eta_k^T \frac{1}{\alpha} P_k \\ &= \frac{1}{\alpha} \left(P_k - \frac{\lambda P_k \eta_k \eta_k^T P_k}{\alpha + \lambda \eta_k^T P_k \eta_k} \right) \end{aligned}$$

podemos escrever que

$$P_{k+1} = \frac{1}{\alpha} \left(P_k - \frac{\lambda P_k \eta_k \eta_k^T P_k}{\alpha + \lambda \eta_k^T P_k \eta_k} \right) - \frac{\zeta \left(\left(P_k - \frac{\lambda P_k \eta_k \eta_k^T P_k}{\alpha + \lambda \eta_k^T P_k \eta_k} \right) \varphi_k \varphi_k^T \left(P_k - \frac{\lambda P_k \eta_k \eta_k^T P_k}{\alpha + \lambda \eta_k^T P_k \eta_k} \right) \right)}{\alpha + \zeta \varphi_k^T \left(P_k - \frac{\lambda P_k \eta_k \eta_k^T P_k}{\alpha + \lambda \eta_k^T P_k \eta_k} \right) \varphi_k}$$

Ou definindo

$$\begin{aligned}\bar{P}_k &= P_k - \frac{\lambda P_k \boldsymbol{\eta}_k \boldsymbol{\eta}_k^T P_k}{\alpha + \lambda \boldsymbol{\eta}_k^T P_k \boldsymbol{\eta}_k} \\ P_{k+1} &= \frac{1}{\alpha} \left(\bar{P}_k - \frac{\zeta \bar{P}_k \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \bar{P}_k}{\alpha + \zeta \boldsymbol{\varphi}_k^T \bar{P}_k \boldsymbol{\varphi}_k} \right)\end{aligned}$$

Dessa forma, nossa lei de adaptação agora é dada por

$$\begin{aligned}\bar{P}_k &= P_k - \frac{\lambda P_k \boldsymbol{\eta}_k \boldsymbol{\eta}_k^T P_k}{\alpha + \lambda \boldsymbol{\eta}_k^T P_k \boldsymbol{\eta}_k} \\ P_{k+1} &= \frac{1}{\alpha} \left(\bar{P}_k - \frac{\zeta \bar{P}_k \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \bar{P}_k}{\alpha + \zeta \boldsymbol{\varphi}_k^T \bar{P}_k \boldsymbol{\varphi}_k} \right) \\ \hat{\Theta}_{k+1} &= \hat{\Theta}_k + \zeta P_{k+1} \boldsymbol{\varphi}_k (\mathbf{x}_{k+1} - (A + BK) \mathbf{x}_k)^T B (B^T B)^{-1}\end{aligned}$$

A dinâmica do erro de adaptação continua sendo dada por

$$\tilde{\Theta}_{k+1} = \tilde{\Theta}_k - \zeta P_{k+1} \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \tilde{\Theta}_k - \zeta P_{k+1} \boldsymbol{\varphi}_k \delta_k^T$$

E a diferença da função de Lyapunov passa a ser dada por

$$\begin{aligned}\Delta V_\theta &= \text{tr} \left(\tilde{\Theta}_{k+1}^T P_{k+1}^{-1} \tilde{\Theta}_{k+1} \right) - \text{tr} \left(\tilde{\Theta}_k^T P_k^{-1} \tilde{\Theta}_k \right) \\ \Delta V_\theta &= \text{tr} \left(\tilde{\Theta}_k^T P_{k+1}^{-1} \tilde{\Theta}_k \right) - 2\zeta \text{tr} \left(\tilde{\Theta}_k^T \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \right) - 2\zeta \text{tr} \left(\tilde{\Theta}_k^T \boldsymbol{\varphi}_k \delta_k^T \right) \\ &\quad + \zeta^2 \text{tr} \left(\tilde{\Theta}_k^T \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \right) + 2\zeta^2 \text{tr} \left(\tilde{\Theta}_k^T \boldsymbol{\varphi}_k \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \delta_k^T \right) \\ &\quad + \zeta^2 \text{tr} \left(\delta_k \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \delta_k^T \right) - \text{tr} \left(\tilde{\Theta}_k^T P_k^{-1} \tilde{\Theta}_k \right) \\ \Delta V_\theta &= \left(-2\zeta + \zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \right) \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \tilde{\Theta}_k^T \boldsymbol{\varphi}_k + \left(-2\zeta + 2\zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \right) \delta_k^T \tilde{\Theta}_k^T \boldsymbol{\varphi}_k \\ &\quad + \zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \delta_k^T \delta_k + \text{tr} \left(\tilde{\Theta}_k^T \left(P_{k+1}^{-1} - \alpha P_k^{-1} \right) \tilde{\Theta}_k \right) - (1 - \alpha) \text{tr} \left(\tilde{\Theta}_k^T P_k^{-1} \tilde{\Theta}_k \right) \\ \Delta V_\theta &= \left(-2\zeta + \zeta + \zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \right) \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \tilde{\Theta}_k^T \boldsymbol{\varphi}_k + \left(-2\zeta + 2\zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \right) \delta_k^T \tilde{\Theta}_k^T \boldsymbol{\varphi}_k \\ &\quad + \left(\zeta - \zeta + \zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k \right) \delta_k^T \delta_k - (1 - \alpha) V_\theta + \lambda \text{tr} \left(\tilde{\Theta}_k^T \boldsymbol{\eta}_k \boldsymbol{\eta}_k^T \tilde{\Theta}_k \right)\end{aligned}$$

Uma vez que

$$\begin{aligned}\zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k &= \frac{1}{\alpha} \left(\zeta^2 \boldsymbol{\varphi}_k^T \bar{P}_k \boldsymbol{\varphi}_k - \frac{\zeta^3 (\boldsymbol{\varphi}_k^T \bar{P}_k \boldsymbol{\varphi}_k)^2}{\alpha + \zeta \boldsymbol{\varphi}_k^T \bar{P}_k \boldsymbol{\varphi}_k} \right) \\ &= \frac{\zeta^2 \boldsymbol{\varphi}_k^T \bar{P}_k \boldsymbol{\varphi}_k}{\alpha + \zeta \boldsymbol{\varphi}_k^T \bar{P}_k \boldsymbol{\varphi}_k}\end{aligned}$$

temos que

$$-\zeta + \zeta^2 \boldsymbol{\varphi}_k^T P_{k+1} \boldsymbol{\varphi}_k = -\zeta + \frac{\zeta^2 \boldsymbol{\varphi}_k^T \bar{P}_k \boldsymbol{\varphi}_k}{\alpha + \zeta \boldsymbol{\varphi}_k^T \bar{P}_k \boldsymbol{\varphi}_k} = \frac{-\alpha \zeta}{\alpha + \zeta \boldsymbol{\varphi}_k^T \bar{P}_k \boldsymbol{\varphi}_k}$$

de modo que

$$\Delta V_\theta = \frac{-\alpha \zeta \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \tilde{\Theta}_k^T \boldsymbol{\varphi}_k - 2\alpha \zeta \delta_k^T \tilde{\Theta}_k^T \boldsymbol{\varphi}_k - \alpha \zeta \delta_k^T \delta_k}{\alpha + \zeta \boldsymbol{\varphi}_k^T \bar{P}_k \boldsymbol{\varphi}_k} + \zeta \delta_k^T \delta_k - (1 - \alpha) V_\theta + \lambda \text{tr}(\tilde{\Theta}_k^T \boldsymbol{\eta}_k \boldsymbol{\eta}_k^T \tilde{\Theta}_k)$$

Considerando que, para $\mu > 0$, temos

$$-2\alpha \delta_k^T \tilde{\Theta}_k^T \boldsymbol{\varphi}_k \leq \mu \alpha^2 \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \tilde{\Theta}_k^T \boldsymbol{\varphi}_k + \frac{\alpha^2}{\mu} \delta_k^T \delta_k$$

e podemos escrever

$$\Delta V_\theta \leq \frac{-\zeta (\alpha - \mu \alpha^2) \boldsymbol{\varphi}_k^T \tilde{\Theta}_k \tilde{\Theta}_k^T \boldsymbol{\varphi}_k}{\alpha + \zeta \boldsymbol{\varphi}_k^T \bar{P}_k \boldsymbol{\varphi}_k} + \left(\zeta + \frac{\alpha}{\mu} \right) \delta_k^T \delta_k - (1 - \alpha) V_\theta + \lambda \boldsymbol{\eta}_k^T \tilde{\Theta}_k \tilde{\Theta}_k^T \boldsymbol{\eta}_k$$

Note que, neste caso, o termo $\lambda \boldsymbol{\eta}_k^T \tilde{\Theta}_k \tilde{\Theta}_k^T \boldsymbol{\eta}_k$ pode atrapalhar um pouco o uso desta lei de adaptação. Neste caso, é aconselhável o uso de uma projeção na lei de adaptação, garantindo que o erro de adaptação seja sempre limitado e que este termo possa ser tratado de forma semelhante ao termo $\delta_k^T \delta_k$.

Outras Modificações do Mínimos Quadrados Recursivo

Vale ressaltar que existem diversas outras modificações apresentadas na literatura para o Mínimos Quadrados Recursivo.

Existem propostas em que o fator de esquecimento é variável, e pensado num contexto em que a estimação é perturbada por um sinal limitado (semelhante ao que temos com nosso erro de aproximação). Dentre estas, podemos destacar a modificação para perturbações constantes (de Wit e Carrillo 1990), em que o fator de esquecimento é encontrado por meio da solução de um problema de otimização com restrição, e o método dos elipsóides limitantes ótimos (*OBE - Optimal Bounding Ellipsoids*) (Dasgupta e Huang 1987), em que o fator de esquecimento é encontrado por uma abordagem de elipsóides que garantem que o parâmetro está contido dentro daquela região.

Outras propostas, buscam lidar com a falta de persistência de excitação, com um fator de esquecimento de direção variável (Goel, Bruce e Bernstein 2020), efetivamente reduzindo a inversa da matriz de covariância apenas nas direções em que os regressores estão introduzindo novas informações.

Apesar de não serem apresentadas aqui na disciplina, estas abordagens podem também ser interessantes quando desejamos utilizar um estimador de mínimos quadrados recursivo em nosso contexto.

Bibliografia

- Bělohlávek, R., J. W. Dauben e G. J. Klir (2017). *Fuzzy logic and mathematics: a historical perspective*. Oxford University Press.
- Bezdek, J. C. (1993). «Fuzzy models? What are they, and why? [Editorial]». Em: *IEEE Transactions on Fuzzy Systems* 1.1, pp. 1–6.
- Bonissone, P. P. e E. H. Ruspini (2017). «Lotfi A. Zadeh In Memoriam». Em: *IEEE Transactions on Fuzzy Systems* 25.5, pp. 1021–1022.
- Dasgupta, S. e Y.-F. Huang (1987). «Asymptotically convergent modified recursive least-squares with data-dependent updating and forgetting factor for systems with bounded noise». Em: *IEEE Transactions on Information Theory* 33.3, pp. 383–392. DOI: [10.1109/TIT.1987.1057307](https://doi.org/10.1109/TIT.1987.1057307).
- Dullerud, G. E. e F. Paganini (2000). *A course in Robust Control Theory*. Springer-Verlag New York. ISBN: 978-1-4419-3189-4.
- Farrel, J. A. e M. M. Polycarpou (2006). *Adaptive approximation based control: unifying neural, fuzzy and traditional adaptive approximation approaches*. Wiley-Interscience.
- Goel, A., A. L. Bruce e D. S. Bernstein (2020). «Recursive Least Squares With Variable-Direction Forgetting: Compensating for the Loss of Persistency [Lecture Notes]». Em: *IEEE Control Systems Magazine* 40.4, pp. 80–102. DOI: [10.1109/MCS.2020.2990516](https://doi.org/10.1109/MCS.2020.2990516).
- Guerra, T. M., R. Márquez, A. Kruszewski e M. Bernal (2018). « H_∞ LMI-Based Observer Design for Nonlinear Systems via Takagi-Sugeno Models With Unmeasured Premise Variables». Em: *IEEE Transactions on Fuzzy Systems* 26.3, pp. 1498–1509. ISSN: 1063-6706. DOI: [10.1109/TFUZZ.2017.2728522](https://doi.org/10.1109/TFUZZ.2017.2728522).
- Guerra, T. M. e L. Vermeiren (2004). «LMI-based relaxed nonquadratic stabilization conditions for nonlinear systems in the Takagi-Sugeno's form». Em: *Automatica* 40.5, pp. 823–829. ISSN: 0005-1098. DOI: [10.1016/j.automatica.2003.12.014](https://doi.org/10.1016/j.automatica.2003.12.014).
- Ichalal, D., H. Arioui e S. Mammar (2011). «Observer design for two-wheeled vehicle: A Takagi-Sugeno approach with unmeasurable premise variables». Em: *2011 19th Mediterranean Conference on Control Automation (MED)*, pp. 934–939. DOI: [10.1109/MED.2011.5983232](https://doi.org/10.1109/MED.2011.5983232).
- Kanellakopoulos, I. (1994). «A discrete-time adaptive nonlinear system». Em: *IEEE Transactions on Automatic Control* 39.11, pp. 2362–2365. DOI: [10.1109/9.333794](https://doi.org/10.1109/9.333794).

- Khalil, H. (2002). *Nonlinear Systems*. 3^a ed. Prentice Hall.
- Lee, D. H., J. B. Park e Y. H. Joo (2012). «A fuzzy Lyapunov function approach to estimating the domain of attraction for continuous-time Takagi-Sugeno fuzzy systems». Em: *Information Sciences* 185.1, pp. 230–248. ISSN: 0020-0255. DOI: [10.1016/j.ins.2011.06.008](https://doi.org/10.1016/j.ins.2011.06.008).
- Lendek, Z., T.-M. Guerra e J. Lauber (2015). «Controller Design for TS Models Using Delayed Nonquadratic Lyapunov Functions». Em: *IEEE Transactions on Cybernetics* 45.3, pp. 439–450. DOI: [10.1109/TCYB.2014.2327657](https://doi.org/10.1109/TCYB.2014.2327657).
- Lopes, F. W. B. (2003). «Dessulfurização do gás natural utilizando sistemas microemulsionados». Tese de mestrado. Natal, RN.: UFRN, Programa de pós-graduação em engenharia química.
- Mei, W. e F. Bullo (2017). *LaSalle Invariance Principle for Discrete-time Dynamical Systems: A Concise and Self-contained Tutorial*. arXiv: [1710.03710 \[math.DS\]](https://arxiv.org/abs/1710.03710).
- Mozelli, L., R. Palhares e E. Mendes (2010). «Equivalent techniques, extra comparisons and less conservative control design for Takagi-Sugeno (TS) fuzzy systems». Em: *IET Control Theory & Applications* 4 (12), 2813–2822(9). ISSN: 1751-8644. DOI: [10.1049/iet-cta.2009.0210](https://doi.org/10.1049/iet-cta.2009.0210).
- Mozelli, L., R. Palhares, F. Souza e E. Mendes (2009). «Reducing conservativeness in recent stability conditions of TS fuzzy systems». Em: *Automatica* 45.6, pp. 1580–1583. ISSN: 0005-1098. DOI: [10.1016/j.automatica.2009.02.023](https://doi.org/10.1016/j.automatica.2009.02.023).
- Mozelli, L. A., R. M. Palhares e G. S. Avellar (2009). «A systematic approach to improve multiple Lyapunov function stability and stabilization conditions for fuzzy systems». Em: *Information Sciences* 179.8, pp. 1149–1162. ISSN: 0020-0255. DOI: [10.1016/j.ins.2008.12.002](https://doi.org/10.1016/j.ins.2008.12.002).
- Oliveira, M. C. de e R. E. Skelton (2001). «Stability tests for constrained linear systems». Em: *Perspectives in robust control*. Springer, pp. 241–257.
- Sala, A. e C. Ariño (2007). «Asymptotically necessary and sufficient conditions for stability and performance in fuzzy control: Applications os Polya’s theorem». Em: *Fuzzy Sets and Systems* 158 (24), pp. 2671–2686.
- Shaked, U. (2001). «Improved LMI representations for the analysis and the design of continuous-time systems with polytopic type uncertainty». Em: *IEEE Transactions on Automatic Control* 46.4, pp. 652–656. DOI: [10.1109/9.917671](https://doi.org/10.1109/9.917671).
- Tanaka, K. e H. Wang (2001). *Fuzzy control systems design and analysis: a linear matrix inequality approach*. Wiley-Interscience. ISBN: 0471323241.
- Tuan, H., P. Apkarian, T. Narikiyo e Y. Yamamoto (2001). «Parameterized linear matrix inequality techniques in fuzzy control system design». Em: *IEEE Transactions on Fuzzy Systems* 9.2, pp. 324–332. ISSN: 1063-6706. DOI: [10.1109/91.919253](https://doi.org/10.1109/91.919253).
- Wang, L.-X. (1999). *A course in fuzzy systems*. Prentice-Hall press, USA.

- Willems, J. C. (1972). «Dissipative dynamical systems part I: General theory». Em: *Archive for Rational Mechanics and Analysis* 45.5, pp. 321–351. ISSN: 1432-0673. DOI: [10.1007/BF00276493](https://doi.org/10.1007/BF00276493).
- Xiaodong, L. e Z. Qingling (2003). «New approaches to H_∞ controller designs based on fuzzy observers for T-S fuzzy systems via LMI». Em: *Automatica* 39.9, pp. 1571–1582. ISSN: 0005-1098. DOI: [10.1016/S0005-1098\(03\)00172-9](https://doi.org/10.1016/S0005-1098(03)00172-9).
- Zadeh, L. A. (1965). «Fuzzy Sets». Em: *Information and Control* 8, pp. 338–353.
- de Wit, C. e J. Carrillo (1990). «A modified EW-RLS algorithm for systems with bounded disturbances». Em: *Automatica* 26.3, pp. 599–606. ISSN: 0005-1098. DOI: [10.1016/0005-1098\(90\)90032-D](https://doi.org/10.1016/0005-1098(90)90032-D).

Apêndice A

Lemas interessantes para manipulações de Desigualdades Matriciais Lineares

A ideia desse texto é apresentar alguns lemas que costumam ser utilizados com LMIs (porque muitas vezes ajuda ter um lugar fácil de consultar os lemas).

Lema A.1

Dada uma matriz definida positiva $P = P^T$ e uma matriz quadrada G , temos

$$G^T P^{-1} G \geq \mu G + \mu G^T - \mu^2 P, \quad \forall \mu \in \mathbb{R}$$

Demonstração. Como $P > 0$, temos

$$\begin{aligned} (G^T - \mu P) P^{-1} (G - \mu P) &\geq 0 \\ G^T P^{-1} G - \mu G - \mu G^T + \mu^2 P &\geq 0 \\ G^T P^{-1} G &\geq \mu G + \mu G^T - \mu^2 P \end{aligned}$$

□

Lema A.2

Dada uma matriz definida positiva $P = P^T$ e uma matriz quadrada G , temos

$$G^T P^{-1} G \geq G^T R + R^T G - R^T P R$$

para qualquer R tal que a dimensão de $G^T R$ seja igual à de P .

Demonstração. Como $P > 0$, temos

$$\begin{aligned} (G^T - R^T P) P^{-1} (G - PR) &\geq 0 \\ G^T P^{-1} G - \mu G - \mu G^T + \mu^2 P &\geq 0 \\ G^T P^{-1} G &\geq \mu G + \mu G^T - \mu^2 P \end{aligned}$$

□

Lema A.3

Dadas duas matrizes A e B ,

$$A^T B + B^T A \leq \mu A^T A + \frac{1}{\mu} B^T B, \quad \forall \mu > 0$$

Demonstração.

$$\begin{aligned} \left(A^T - \frac{1}{\mu} B^T \right) \mu \left(A - \frac{1}{\mu} B \right) &\geq 0 \\ \mu A^T A + \frac{1}{\mu} B^T B - A^T B - B^T A &\geq 0 \\ \mu A^T A + \frac{1}{\mu} B^T B &\geq A^T B + B^T A \end{aligned}$$

□

Lema A.4: Complemento de Schur

$$\begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} > 0 \quad (\text{A.1})$$

se, e somente se

$$P_{22} > 0, \quad P_{11} - P_{12} P_{22}^{-1} P_{12}^T > 0$$

ou

$$P_{11} > 0, \quad P_{22} - P_{12}^T P_{11}^{-1} P_{12} > 0$$

Demonstração. Note que as condições

$$P_{11} > 0, \quad P_{22} > 0$$

são necessárias, uma vez a forma quadrática associada à equação (A.1)

$$\begin{bmatrix} \mathbf{u}^T & \mathbf{v}^T \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \mathbf{u}^T P_{11} \mathbf{u} + 2\mathbf{u}^T P_{12} \mathbf{v} + \mathbf{v}^T P_{22} \mathbf{v}$$

se reduz a

$$\begin{aligned}\mathbf{u}^T P_{11} \mathbf{u} &> 0, \quad \text{quando } \mathbf{v} = 0 \\ \mathbf{v}^T P_{22} \mathbf{v} &> 0, \quad \text{quando } \mathbf{u} = 0\end{aligned}$$

Vamos agora estudar o comportamento do mínimo da forma quadrática em relação a \mathbf{u} (assumindo que ela seja convexa, ou seja, que a desigualdade (A.1) é satisfeita). Derivando a forma quadrática e igualando a zero, temos

$$\begin{aligned}2P_{11}\mathbf{u} + 2P_{12}\mathbf{v} &= 0 \\ P_{11}\mathbf{u} &= -P_{12}\mathbf{v} \\ \mathbf{u} &= -P_{11}^{-1}P_{12}\mathbf{v}\end{aligned}$$

substituindo de volta na forma quadrática

$$\begin{aligned}\mathbf{u}^T P_{11} \mathbf{u} + 2\mathbf{u}^T P_{12} \mathbf{v} + \mathbf{v}^T P_{22} \mathbf{v} &= \mathbf{v}^T P_{12}^T P_{11}^{-1} P_{12} \mathbf{v} - 2\mathbf{v}^T P_{12}^T P_{11}^{-1} P_{12} \mathbf{v} + \mathbf{v}^T P_{22} \mathbf{v} \\ &= \mathbf{v}^T \left(P_{22} - P_{12}^T P_{11}^{-1} P_{12} \right) \mathbf{v}\end{aligned}$$

Note que, se $P_{22} - P_{12}^T P_{11}^{-1} P_{12} > 0$, então $\mathbf{v}^T \left(P_{22} - P_{12}^T P_{11}^{-1} P_{12} \right) \mathbf{v} \forall \mathbf{v} \neq 0$. Como este é o valor mínimo da forma quadrática associado a \mathbf{u} , e ele é definido positivo, então a forma quadrática original também é, e a desigualdade (A.1) é satisfeita.

A outra forma associada ao complemento de Schur pode ser encontrada de forma similar, mas encontrando o mínimo relacionada a \mathbf{v} . \square

Lema A.5: Lema Projetivo - adaptado de (Dullerud e Paganini 2000)

A LMI

$$L + P^T J R + R^T J^T P < 0 \tag{A.2}$$

é factível se, e somente se as LMIs

$$\begin{aligned}R^\perp{}^T L R^\perp &< 0, \quad R R^\perp = 0 \\ P^\perp{}^T L P^\perp &< 0, \quad P P^\perp = 0\end{aligned}$$

também o forem.

Demonstração. Note primeiro que R^\perp e P^\perp são bases para o espaço nulo de R e P respectivamente.

Se ambos os espaços nulos tiverem dimensão igual a zero, $P^T J R$ pode representar qualquer matriz, e o fato de que existe solução para (A.2) é trivial.

Defina V_1 tal que V_1 esteja na interseção dos espaços nulos de R e P . Além disso, defina V_2 tal que $[V_1 \ V_2]$ seja uma base para o espaço nulo de R e defina V_3 tal que $[V_1 \ V_3]$ seja uma base para o espaço nulo de P .

Defina V_4 tal que a matriz

$$V = [V_1 \ V_2 \ V_3 \ V_4]$$

seja quadrada e não singular.

Já que a matriz V é quadrada e não-singular, a equação (A.2) é factível se, e somente se

$$V^T L V + V^T P^T J R V + V^T R^T J^T P V < 0 \quad (\text{A.3})$$

Note que, da forma como as matrizes V foram definidas

$$R V = \begin{bmatrix} 0 & 0 & R_1 & R_2 \end{bmatrix}$$

$$P V = \begin{bmatrix} 0 & P_1 & 0 & P_2 \end{bmatrix}$$

Definindo também

$$V^T L V = \begin{bmatrix} L_{11} & L_{12} & L_{13} & L_{14} \\ L_{12}^T & L_{22} & L_{23} & L_{24} \\ L_{13}^T & L_{23}^T & L_{33} & L_{34} \\ L_{14}^T & L_{24}^T & L_{34}^T & L_{44} \end{bmatrix}$$

Defina Y como

$$Y = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} = \begin{bmatrix} R_1^T \\ R_2^T \end{bmatrix} J^T \begin{bmatrix} P_1 & P_2 \end{bmatrix}$$

Como os espaços nulos de $[R_1 \ R_2]$ e $[P_1 \ P_2]$ tem dimensão zero por construção, sabemos que Y pode ser escolhido arbitrariamente pela escolha de um J adequado.

Com essas definições, podemos escrever a condição (A.3) como

$$\begin{bmatrix} L_{11} & L_{12} & L_{13} & L_{14} \\ L_{12}^T & L_{22} & L_{23} + Y_{11}^T & L_{24} + Y_{21}^T \\ L_{13}^T & L_{23}^T + Y_{11} & L_{33} & L_{34} + Y_{12} \\ L_{14}^T & L_{24}^T + Y_{21} & L_{34}^T + Y_{12}^T & L_{44} + Y_{22} + Y_{22}^T \end{bmatrix} < 0$$

Utilizando o complemento do Schur com o bloco superior esquerdo 3×3 , sabemos que a condição acima é verdadeira se, e somente se, as condições

$$\bar{L} = \begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{12}^T & L_{22} & L_{23} + Y_{11}^T \\ L_{13}^T & L_{23}^T + Y_{11} & L_{33} \end{bmatrix} < 0,$$

$$L_{44} + Y_{22} + Y_{22}^T - \begin{bmatrix} L_{14}^T & L_{24}^T + Y_{21} & L_{34}^T + Y_{12}^T \end{bmatrix} \bar{L}^{-1} \begin{bmatrix} L_{14} \\ L_{24} + Y_{21}^T \\ L_{34} + Y_{12} \end{bmatrix} < 0$$

também o forem. Note que, como as matrizes Y podem ser arbitrariamente escolhidas, sempre é possível escolher Y_{22} tal que a segunda condição seja factível. Logo, basta que exista Y_{11} tal que a primeira condição seja factível.

Aplicando o complemento de Schur nesta desigualdade restante, temos

$$L_{11} < 0$$

$$\begin{bmatrix} L_{22} & L_{23} + Y_{11}^T \\ L_{23}^T + Y_{11} & L_{33} \end{bmatrix} - \begin{bmatrix} L_{12}^T \\ L_{13}^T \end{bmatrix} L_{11}^{-1} \begin{bmatrix} L_{12} & L_{13} \end{bmatrix} < 0$$

Note que essa segunda desigualdade pode ser escrita como

$$\begin{bmatrix} L_{22} - L_{12}^T L_{11}^{-1} L_{12} & L_{23} + Y_{11}^T - L_{12}^T L_{11}^{-1} L_{13} \\ L_{23}^T + Y_{11} - L_{13}^T L_{11}^{-1} L_{12} & L_{33} - L_{13}^T L_{11}^{-1} L_{13} \end{bmatrix} < 0$$

Como Y_{11} pode ser escolhido arbitrariamente, podemos escolher $Y_{11} = -L_{23}^T + L_{13}^T L_{11}^{-1} L_{12}$, de modo que passamos a ter as condições

$$\begin{aligned} L_{11} &< 0 \\ L_{22} - L_{12}^T L_{11}^{-1} L_{12} &< 0 \\ L_{33} - L_{13}^T L_{11}^{-1} L_{13} &< 0 \end{aligned}$$

Aplicando o complemento de Schur nessas condições, chegamos em

$$\begin{aligned} \begin{bmatrix} L_{11} & L_{12} \\ L_{12}^T & L_{22} \end{bmatrix} &< 0 \\ \begin{bmatrix} L_{11} & L_{13} \\ L_{13}^T & L_{33} \end{bmatrix} &< 0 \end{aligned}$$

Entretanto, pela definição das matrizes V , tais condições são equivalentes à

$$\begin{aligned} R^\perp{}^T L R^\perp &< 0, & R R^\perp &= 0 \\ P^\perp{}^T L P^\perp &< 0, & P P^\perp &= 0 \end{aligned}$$

□

Lema A.6: Lema de Finsler - adaptado de (Oliveira e Skelton 2001)

Considere $\mathbf{x} \in \mathbb{R}^n$, $Q \in \mathbb{R}^{n \times n}$ e $B \in \mathbb{R}^{m \times n}$ tal que $\text{posto}(B) = \text{rank}(B) < n$. As seguintes afirmações são equivalentes

1. $\mathbf{x}^T Q \mathbf{x} < 0, \quad \forall \mathbf{x} | B \mathbf{x} = 0, \mathbf{x} \neq 0$
2. $B^\perp{}^T Q B^\perp < 0, \quad B B^\perp = 0$
3. $\exists \mu \in \mathbb{R} | Q - \mu B^T B < 0$
4. $\exists X \in \mathbb{R}^{n \times m} | Q + X B + B^T X^T < 0$

Demonstração. 1. \Leftrightarrow 2. Como B^\perp representa uma base para o espaço nulo de B , qualquer \mathbf{x} para o qual $B\mathbf{x} = 0$ pode ser escrito como

$$\mathbf{x} = B^\perp \mathbf{y},$$

logo 1) pode ser reescrita como

$$\mathbf{y}^T B^{\perp T} Q B^\perp \mathbf{y} < 0,$$

que tem como condição necessária e suficiente a condição de que

$$B^{\perp T} Q B^\perp < 0.$$

3., 4. \Rightarrow 2. Multiplique 3. ou 4. a esquerda por $B^{\perp T}$ e a direita por B^\perp e recupera-se 2..

2. \Rightarrow 3. Assuma que 2. seja verdadeira. Decomponha B em dois termos de posto completo $B = B_l B_r$, defina

$$D = B_r^T (B_r B_r^T)^{-1} (B_l^T B_l)^{1/2}$$

e aplique a transformação de congruência

$$\begin{bmatrix} D^T \\ B^{\perp T} \end{bmatrix} (Q - \mu B^T B) \begin{bmatrix} D & B^\perp \end{bmatrix} = \begin{bmatrix} D^T Q D - \mu I & D^T Q B^\perp \\ * & B^{\perp T} Q B^\perp \end{bmatrix} < 0$$

Já que assumimos que 2. é verdadeira, então o bloco na diagonal inferior direita é definido negativo, e, pelo complemento de Schur, existe um μ grande o suficiente tal que a desigualdade apresentada seja satisfeita.

3. \Rightarrow 4. Escolha $X = -(\mu/2)B^T$. □

Apêndice B

Somatórios Fuzzy

Muitas das condições de análise e síntese para sistemas TS podem ser expressas como condições de sinal de matrizes sobre somatórios fuzzy.

Algumas dessas condições, principalmente as relacionadas à análise de sistemas TS, podem ser escritas como

$$\sum_{i=1}^r h_i Q_i > 0,$$

em que Q_i são matrizes formadas pelos parâmetros do sistema e as variáveis de decisão do problema de otimização.

Outras condições, como muitas condições de síntese para sistemas TS disponíveis na literatura, podem ser escritas como

$$\sum_{i=1}^r \sum_{j=1}^r h_i h_j Q_{ij} > 0, \quad (\text{B.1})$$

em que Q_{ij} são matrizes formadas pelos parâmetros do sistema e as variáveis de decisão do problema de otimização.

Resultados Úteis

De modo a garantir que a condição (B.1) seja válida, várias condições suficientes foram propostas na literatura. Alguns exemplos dessas condições são apresentados abaixo:

Lema B.1: (Tanaka e Wang 2001)

Uma condição suficiente para que a expressão (B.1) seja válida, com $i, j \in \{1, \dots, r\}$, é

$$\begin{aligned} Q_{ii} &> 0, \\ Q_{ij} + Q_{ji} &> 0, \quad j > i. \end{aligned}$$

Demonstração. Note que podemos escrever

$$\begin{aligned} & \sum_{i=1}^r \sum_{j=1}^r h_i h_j Q_{ij} \\ &= \sum_{i=1}^r h_i^2 Q_{ii} + \sum_{i=1}^{r-1} \sum_{j=i+1}^r h_i h_j (Q_{ij} + Q_{ji}) \end{aligned}$$

Logo, as condições do lema são suficientes para a expressão (B.1). \square

Lema B.2: Teorema 2.2 - 1 em (Tuan et al. 2001)

Uma condição suficiente para que a expressão (B.1) seja válida, com $i, j \in \{1, \dots, r\}$, é

$$\begin{aligned} & Q_{ii} > 0, \\ & \frac{2}{r-1} Q_{ii} + Q_{ij} + Q_{ji} > 0, \quad j \neq i. \end{aligned}$$

Demonstração. Note que podemos escrever

$$\begin{aligned} & \sum_{i=1}^r \sum_{j=1}^r h_i h_j Q_{ij} \\ &= \sum_{i=1}^r \sum_{j=i+1}^{r-1} \left[\frac{1}{r-1} h_i^2 Q_{ii} + \frac{1}{r-1} h_j^2 Q_{jj} + h_i h_j (Q_{ij} + Q_{ji}) \right] \end{aligned}$$

Logo, as condições do lema são suficientes para a expressão (B.1). \square

Lema B.3: Adaptado do Teorema 2 em (Xiaodong e Qingling 2003)

Uma condição suficiente para que a expressão (B.1) seja válida, com $i, j \in \{1, \dots, r\}$, é que existam matrizes $\Theta_{ij} = \Theta_{ji}^T$ tais que

$$\begin{aligned} & Q_{ii} > \Theta_{ii}, \\ & Q_{ij} + Q_{ji} > \Theta_{ij} + \Theta_{ji}, \quad j > i, \\ & \Theta = \begin{bmatrix} \Theta_{11} & \dots & \Theta_{1r} \\ \vdots & \ddots & \vdots \\ \Theta_{r1} & \dots & \Theta_{rr} \end{bmatrix} > 0. \end{aligned}$$

Demonstração. Note que as duas primeiras condições do lema implicam que

$$\begin{aligned} & \sum_{i=1}^r \sum_{j=1}^r h_i h_j Q_{ij} \\ & \geq \sum_{i=1}^r \sum_{j=1}^r h_i h_j \Theta_{ij} \\ & = \begin{bmatrix} h_1 I \\ \vdots \\ h_r I \end{bmatrix}^T \begin{bmatrix} \Theta_{11} & \dots & \Theta_{1r} \\ \vdots & \ddots & \vdots \\ \Theta_{r1} & \dots & \Theta_{rr} \end{bmatrix} \begin{bmatrix} h_1 I \\ \vdots \\ h_r I \end{bmatrix}^T \end{aligned}$$

e a última condição implica que essa expressão é definida positiva, logo as condições do lema são suficientes para a condição (B.1). \square

Se estivermos trabalhando com múltiplos somatórios fuzzy ao invés de um somatório duplo, o seguinte Lema (que é uma generalização do Lema B.1) pode ser utilizado:

Lema B.4: (Sala e Ariño 2007)

Uma condição suficiente para

$$\sum_{i_1=1}^r \cdots \sum_{i_q=1}^r \left(\prod_{\ell=1}^q h_{i_\ell} \right) L_{i_1 \dots i_q} > 0,$$

é que, para toda combinação de (i_1, i_2, \dots, i_q) , com $i_\ell \in \{1, 2, \dots, r\}$, a soma de suas permutações é definida positiva.

Exemplo B.1. Considere uma condição da forma

$$\sum_{i_1=1}^2 \sum_{i_2=1}^2 \sum_{i_3=1}^2 h_{i_1} h_{i_2} h_{i_3} L_{i_1 i_2 i_3} > 0,$$

as combinações possíveis de (i_1, i_2, i_3) são $(1, 1, 1)$, $(1, 2, 1)$, $(1, 2, 2)$ e $(2, 2, 2)$ e o uso do Lema B.4 implica que condições LMI suficientes para o somatório fuzzy são dadas por

$$L_{111} > 0,$$

$$L_{112} + L_{121} + L_{211} > 0,$$

$$L_{122} + L_{212} + L_{221} > 0,$$

$$L_{222} > 0.$$

Estas condições podem facilmente serem vistas como suficientes para a condição inicial, uma vez que a condição inicial pode ser reescrita como

$$h_1^3 L_{111} + h_1^2 h_2 (L_{112} + L_{121} + L_{211}) + h_1 h_2^2 (L_{122} + L_{212} + L_{221}) + h_2^3 L_{222} > 0.$$

Apêndice C

Transformação do Produto Tensorial

A transformação do produto tensorial, do inglês Tensor Product model transformation, é uma técnica numérica que permite, dada uma representação quasi-Linear a Parâmetros Variantes (qLPV - do inglês quasi-Linear Parameter Varying) de um sistema, encontrar uma representação convexa equivalente. Seu uso pode ser interessante tanto na obtenção numérica de uma representação Takagi-Sugeno de um sistema, quanto na redução do número de regras de um sistema.

Para facilitar a exposição, inicialmente são apresentados alguns conceitos de álgebra multilinear e a generalização da decomposição de valores singulares para tensores, HOSVD. Em seguida, a técnica da transformação do produto tensorial é apresentada. Por fim, é apresentado um exemplo de aplicação da técnica para sistemas dinâmicos.

Conceitos Preliminares

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 (pense em algo similar a encontrar uma base de vetores linearmente independentes para um subespaço). Entretanto, de modo a enunciar o teorema que define esta decomposição, algumas definições são necessárias.

A primeira delas, chamada de *matriz modo- n* ou *matriz de desdobramento* de um tensor, é uma representação matricial para tensores de alta ordem e permite que algumas operações tensoriais sejam representadas como operações matriciais.

Definição C.1: Matriz modo- n de um tensor

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* $\mathcal{A}_{(n)} \in \mathbb{R}^{I_n \times J}$, com $J = \prod_{\substack{k=1 \\ k \neq n}}^N I_k$, 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 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.

Exemplo C.1. Dado um tensor $\mathcal{A} \in \mathbb{R}^{2 \times 2 \times 2}$, seu desdobramento é feito como apresentado acima.

No caso da *matriz modo-1*, o primeiro índice indica a linha do elemento e as colunas são “desdobradas” primeiro pelo terceiro índice e em seguida pelo segundo.

No caso da *matriz modo-2*, o segundo índice indica a linha do elemento e as colunas são “desdobradas” primeiro pelo primeiro índice e em seguida pelo terceiro.

No caso da *matriz modo-3*, o terceiro índice indica a linha do elemento e as colunas são “desdobradas” primeiro pelo segundo índice e em seguida pelo primeiro.

Suas matrizes de desdobramento, então, são dadas por:

$$\mathcal{A}_{(1)} = \left[\begin{array}{cc|cc} a_{111} & a_{112} & a_{121} & a_{122} \\ a_{211} & a_{212} & a_{221} & a_{222} \end{array} \right],$$

$$\mathcal{A}_{(2)} = \left[\begin{array}{cc|cc} a_{111} & a_{211} & a_{112} & a_{212} \\ a_{121} & a_{221} & a_{122} & a_{222} \end{array} \right],$$

$$\mathcal{A}_{(3)} = \left[\begin{array}{cc|cc} a_{111} & a_{121} & a_{211} & a_{221} \\ a_{112} & a_{122} & a_{212} & a_{222} \end{array} \right].$$

Definindo o produto interno entre dois tensores, podemos utilizá-lo para definir uma norma e a ortogonalidade entre dois tensores.

Definição C.2: Produto escalar de dois 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

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1} \sum_{i_2} \dots \sum_{i_N} b_{i_1 i_2 \dots i_N} a_{i_1 i_2 \dots i_N}.$$

Definição C.3: Ortogonalidade de tensores

Dois tensores são ditos ortogonais se seu produto escalar é igual a 0.

Definição C.4: Norma de Frobenius de um tensor

A norma de Frobenius de um tensor \mathcal{A} é dada por

$$\|\mathcal{A}\| = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}.$$

Po último, é necessário definir o produto modo- n entre um tensor e uma matriz. Tal produto pode ser visto como uma generalização da multiplicação à esquerda e à direita entre duas matrizes (equivalentes ao produto modo-1 e produto modo-2 nessa notação).

Definição C.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 $U \in \mathbb{R}^{J_n \times I_n}$, representado por $\mathcal{A} \times_n U$, 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

$$(\mathcal{A} \times_n U)_{i_1 i_2 \dots i_{n-1} j_n i_{n+1} \dots i_N} = \sum_{i_n} a_{i_1 i_2 \dots i_{n-1} i_n i_{n+1} \dots i_N} U_{j_n i_n}.$$

Tal operação pode também ser descrita em termos das matrizes de modo- n do tensor \mathcal{A} e do tensor resultado da operação como

$$(\mathcal{A} \times_n U)_{(n)} = U \mathcal{A}_{(n)}.$$

Propriedade C.1

Dado um tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ e as matrizes $F \in \mathbb{R}^{J_n \times I_n}$ e $G \in \mathbb{R}^{J_m \times I_m}$, com $n \neq m$, tem-se que

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

Propriedade C.2

Dado um tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ e as matrizes $F \in \mathbb{R}^{J_n \times I_n}$ e $G \in \mathbb{R}^{K_n \times J_n}$, tem-se que

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

A partir dessas definições, pode-se enunciar a HOSVD, que será essencial para a técnica da transformação do produto tensorial.

Teorema C.1: HOSVD

Todo tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ pode ser escrito como o produto

$$\mathcal{A} = \mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \dots \times_N U^{(N)},$$

representado de maneira curta como

$$\mathcal{A} = \mathcal{S} \times_{n=1}^N U^{(n)},$$

em que

1. $U^{(n)} = \begin{bmatrix} \mathbf{u}_1^{(n)} & \mathbf{u}_2^{(n)} & \dots & \mathbf{u}_{I_n}^{(n)} \end{bmatrix}$ é uma matriz unitária de dimensões $I_n \times I_n$ chamada de matriz singular de modo- n ;
2. $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, 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 $\alpha \neq \beta$:

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

b) ordenação:

$$\|\mathcal{S}_{i_n=1}\| \geq \|\mathcal{S}_{i_n=2}\| \geq \dots \geq \|\mathcal{S}_{i_n=I_n}\| \geq 0$$

para todos os valores possíveis de n .

As normas de Frobenius $\|\mathcal{S}_{i_n=i}\|$, simbolizadas como $\sigma_i^{(n)}$, são os valores singulares de modo- n de \mathcal{A} e o vetor $\mathbf{u}_i^{(n)}$ é o vetor singular de modo- n correspondente.

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.

As matrizes singulares de modo- n são as matrizes de vetores singulares esquerdos das matrizes de desdobramento do tensor. Sendo assim, cada matriz (e os respectivos

valores singulares de modo- n) é encontrada a partir de uma decomposição de valores singulares de cada matriz de desdobramento do tensor.

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

$$\mathcal{S} = \mathcal{A} \times_1 U^{(1)T} \times_2 U^{(2)T} \cdots \times_N U^{(N)T}.$$

Transformação do Produto Tensorial - Passos

A transformação do produto tensorial é uma técnica numérica que fornece aproximações convexas para funções, válidas dentro de um subespaço limitado do domínio da função a ser aproximada. Tais aproximações convexas podem ser representadas através de um polítopo de valores (não necessariamente pertencentes ao espaço imagem da função) cuja combinação convexa é capaz de representar o comportamento da função. Tal representação é equivalente a um modelo fuzzy Takagi-Sugeno da função.

Quando aplicada sobre um modelo qLPV de um sistema dinâmico, tal técnica é então capaz de encontrar uma representação fuzzy Takagi-Sugeno do sistema. Neste caso, temos um polítopo de sistemas lineares, pois fixando os valores dos parâmetros do modelo qLPV, este se torna um modelo linear. Tal representação permite o uso de LMIs para a síntese de controladores, ou para análise do sistema.

A seguir, apresentam-se os passos da técnica da transformação do produto tensorial e seu uso de uma forma geral. Em seguida, é apresentado um exemplo de seu uso específico para a obtenção de um modelo Takagi-Sugeno de um sistema.

Amostragem e Representação Tensorial

O primeiro passo para a aplicação da transformação do produto tensorial é, dada a função a ser aproximada, amostrá-la dentro de um espaço hiperretangular (que corresponde ao domínio de validade da aproximação) e armazenar tais amostras em um tensor.

Dada uma função $f : [\underline{x}_1, \bar{x}_1] \times \cdots \times [\underline{x}_N, \bar{x}_N] \rightarrow \mathbb{R}^{M_1 \times \cdots \times M_m}$, em que $[\underline{x}_1, \bar{x}_1] \times \cdots \times [\underline{x}_N, \bar{x}_N]$ é um subespaço hiperretangular pertencente ao \mathbb{R}^N , define-se um grid de amostragem de tamanho $I_1 \times \cdots \times I_N$ sobre o domínio da função.

Define-se um tensor, $\mathcal{S}_d \in \mathbb{R}^{I_1 \times \cdots \times I_N \times M_1 \times \cdots \times M_m}$, que guarda os valores amostrados de f para cada ponto do grid.

Consideração C.1

A premissa básica, usual, para o uso da técnica de transformação do produto tenso-rial 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.

Dessa forma, pode-se escrever a interpolação como

$$f(\mathbf{x}) \approx \mathcal{S}_d \times_{n=1}^N \lambda^{(n)}(x_n). \quad (\text{C.1})$$

em que as funções $\lambda^{(n)}(x_n)$ são funções de peso que, quando multiplicadas pelas amostras, geram a interpolação polinomial desejada (polinômios de Lagrange por partes).

Exemplo C.2. Considere a função $f(x, y) = xy(1 + \sin(x) \cos(y))$, com $x \in [-10, 10]$ e $y \in [-10, 10]$, apresentada na Figura C.1. Tem-se que $f : [-10, 10] \times [-10, 10] \rightarrow \mathbb{R}$.

Definindo um passo de amostragem do grid de 0,1 em x e em y , tem-se um grid de amostragem de tamanho 201×201 . Avaliando a função em cada ponto do grid e guardando no tensor \mathcal{S}_d , obtêm-se na verdade uma matriz de tamanho 201×201 .

Decomposição de Valores Singulares de Alta Ordem - HOSVD

Tendo definido o tensor $\mathcal{S}_d \in \mathbb{R}^{I_1 \times \dots \times I_N \times M_1 \times \dots \times M_m}$ que armazena as amostras da função a ser aproximada, utiliza-se a HOSVD apresentada na seção anterior para decompor o tensor.

A ideia aqui é apresentar o tensor de amostras como uma soma ponderada, onde os pesos variam de acordo com as variáveis da função. Sendo assim, percebe-se que para esse fim não é interessante decompor o tensor nas $N + m$ dimensões diferentes. Neste caso, ele é decomposto apenas nas N primeiras dimensões que representam as variáveis da função a ser aproximada.

Pode-se então escrevê-lo como:

$$\mathcal{S}_d = \mathcal{S} \times_{n=1}^N U^{(n)}.$$

Nessa notação podemos entender que o tensor de amostras pode ser representado como uma soma ponderada, em que cada matriz $U^{(n)}$ representa os pesos relativos a variável n do grid de amostras. Cada vetor coluna $\mathbf{u}_i^{(n)}$ representa uma função de peso diferente para essa variável.

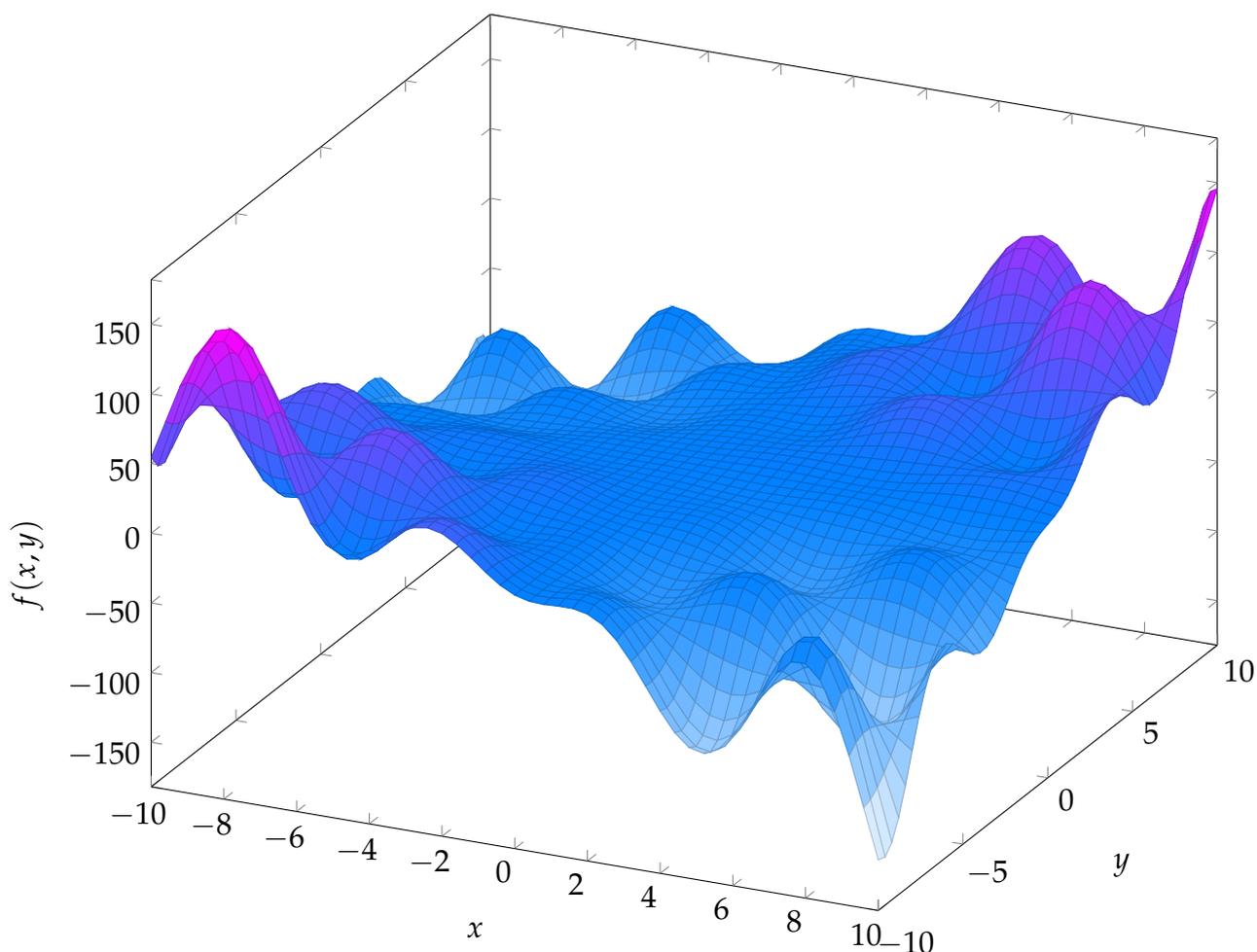


Figura C.1: $f(x, y) = xy(1 + \text{sen}(x) \cos(y))$, utilizada no exemplo C.2.

Conforme apresentado anteriormente, o cálculo da HOSVD ocorre em dois passos. Durante o primeiro passo é realizada uma decomposição de valores singulares de cada matriz de desdobramento do tensor de modo a encontrar as matrizes $U^{(n)}$ e os valores singulares de modo- n . Nesta etapa, cada coluna $\mathbf{u}_i^{(n)}$ encontrada corresponde a um valor 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.

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 fun-

ções de peso para cada variável, que é equivalente ao número de colunas das matrizes $U^{(n)}$ e a precisão da aproximação.

Exemplo C.3. Considerando ainda a função utilizada no exemplo C.2 e o tensor (matriz) \mathcal{S}_d . Mantendo-se apenas as colunas $\mathbf{u}_i^{(n)}$ que possuem valores singulares maiores do que 1×10^{-5} o maior valor singular daquele modo, chega-se em uma representação com $U^{(1)} \in \mathbb{R}^{201 \times 2}$, $U^{(2)} \in \mathbb{R}^{201 \times 2}$ e $\mathcal{S} \in \mathbb{R}^{2 \times 2}$.

Pensando nas matrizes $U^{(n)}$, apresentadas na Figura C.2, como representando funções de peso da variável n do grid, temos que, dentro do intervalo de aproximação, a função $f(x, y) = xy(1 + \sin(x) \cos(y))$ pode ser representada pela soma dos termos do tensor de núcleo \mathcal{S} ponderada pelas funções de peso.

No caso deste exemplo, podemos escrever:

$$f(x, y) = \sum_{i=1}^2 \sum_{j=1}^2 \mathbf{u}_i^{(1)} \mathbf{u}_j^{(2)} \mathcal{S}_{ij}.$$

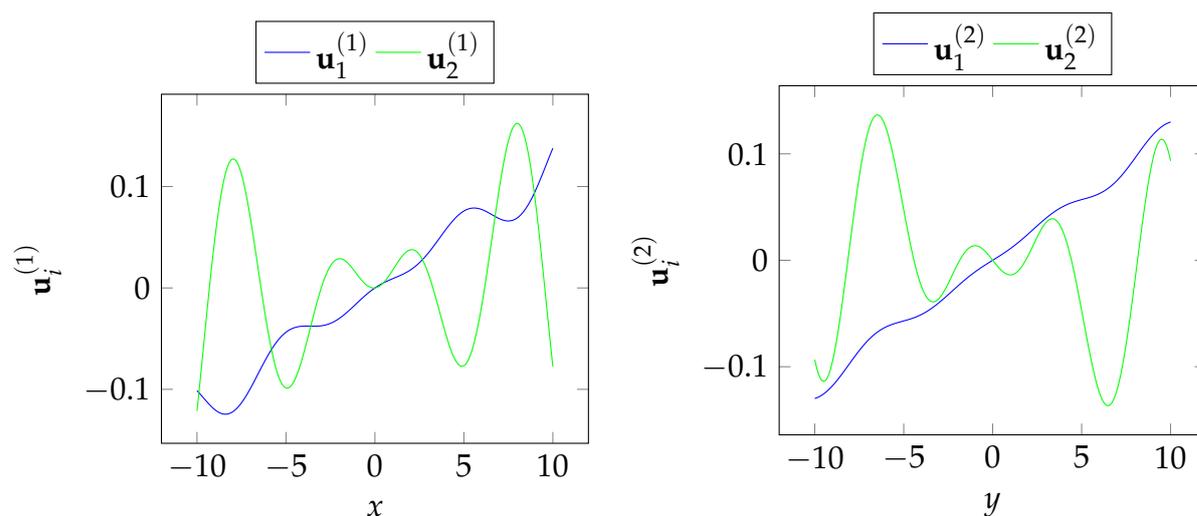


Figura C.2: funções de peso obtidas no exemplo C.3.

Representações convexas

O uso da HOSVD permitiu a decomposição da função como uma soma ponderada de um produto de termos dependentes de apenas uma variável. Entretanto, em alguns casos, busca-se uma representação com características adicionais.

Para que a representação obtida seja semelhante a um modelo Takagi-Sugeno da função é necessário que as funções de peso tenham características semelhantes às de funções de pertinência de um conjunto fuzzy. 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 $U^{(n)}$, que por sua vez podem ser vistas como características do tensor de núcleo \mathcal{S} .

Para se 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 que $\tilde{U}^{(n)} = U^{(n)}T_n$, tem-se que $U^{(n)} = \tilde{U}^{(n)}T_n^{-1}$. Utilizando as propriedades C.1 e C.2 é possível mostrar que

$$\mathcal{S} \times_{n=1}^N U^{(n)} = \tilde{\mathcal{S}} \times_{n=1}^N \tilde{U}^{(n)},$$

com $\tilde{\mathcal{S}}$ dado por

$$\tilde{\mathcal{S}} = \mathcal{S} \times_{n=1}^N T_n^{-1}.$$

Na literatura, as características normalmente são definidas em relação às funções de peso (obtidas após a interpolação das colunas das matrizes de peso). 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 das matrizes de peso.

A seguir são apresentadas as características normalmente utilizadas na literatura e o porquê elas são desejáveis.

Soma Normalizada (SN) e Não Negatividade (NN)

Definição C.6: Soma Normalizada (SN)

Soma Normalizada (SN) - Uma matriz $U^{(n)}$ é dita normalizada em soma se a soma de suas colunas, $\mathbf{u}_i^{(n)}$, resulta em um vetor cujos componentes são todos iguais a um (representado por $\mathbf{1}$). Escrevendo de outra maneira, uma matriz $U^{(n)}$ com r colunas é dita normalizada em soma se

$$\sum_{i=1}^r \mathbf{u}_i^{(n)} = \mathbf{1}.$$

Definição C.7: Não Negatividade (NN)

Não Negatividade (NN) - Uma matriz $U^{(n)}$ é dita não negativa se nenhum de seus elementos é negativo. Escrevendo de outra maneira, uma matriz $U^{(n)}$ é dita não negativa se

$$u_{ij}^{(n)} \geq 0, \quad \forall i, j.$$

Quando todas as matrizes de peso são SN e NN, o modelo obtido equivale a um modelo Takagi-Sugeno e as funções de peso (correspondentes às colunas das matrizes de peso) são equivalentes a funções de pertinência de um modelo Takagi-Sugeno. Sendo

assim, sempre que se é desejável encontrar uma aproximação fuzzy Takagi-Sugeno deseja-se que todas as matrizes de peso sejam SN e NN.

Exemplo C.4. Considerando as matrizes de peso obtidas no exemplo C.3 e aplicando uma transformação de modo a torná-las SN e NN obtêm-se as funções de peso apresentadas na Figura C.3. É interessante notar que em ambos os casos não foi possível obter as transformações mantendo-se o número de colunas das matrizes de peso (como pode-se perceber que uma nova função foi acrescentada a cada gráfico, comparando as Figuras C.2 e C.3).

Nota C.1

Para que a transformação 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

$$\mathbf{1} - \sum_{i=1}^r \mathbf{u}_i^{(n)}.$$

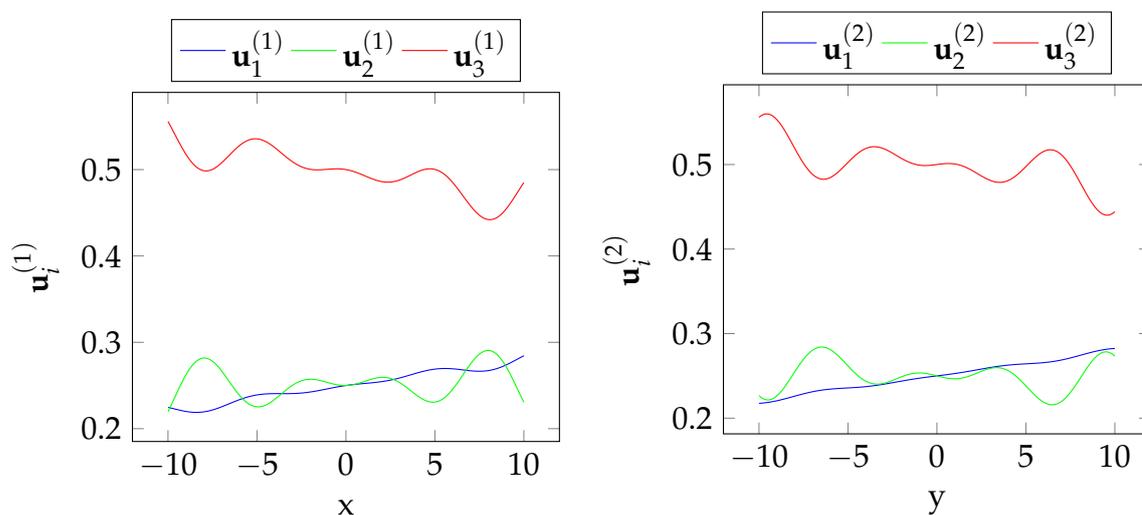


Figura C.3: funções de peso SN-NN obtidas no exemplo C.4.

Normalizada (NO) e Próxima a Normalizada (CNO)

Definição C.8: Normalizada (NO)

Normalizada (NO) - Uma matriz $U^{(n)}$ é dita normalizada se ela é SN e NN e além disso os valores máximos de cada coluna, $\mathbf{u}_i^{(n)}$, são os mesmos e iguais a um. Escrevendo de outra maneira, uma matriz $U^{(n)}$ é dita normalizada se ela é SN, NN e

$$\max(\mathbf{u}_i^{(n)}) = 1, \quad \forall i.$$

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 C.9: Próxima a Normalizada (CNO) - definição usual

Próxima a Normalizada (CNO) - Uma matriz $U^{(n)}$ é dita próxima a normalizada se ela é SN e NN e além disso os valores máximos de cada coluna são próximos de um.

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 C.10: Próxima a Normalizada (CNO) - definição alternativa

Próxima a Normalizada (CNO) - Uma matriz $U^{(n)}$ é dita próxima a normalizada se ela é SN e NN e o fecho convexo dos seus vetores linha tem o maior volume possível dentro do simplex unitário padrão (*i.e.* não é possível alargar o fecho convexo sem que as condições de SN e NN deixem de ser atendidas).

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

Transformação CNO

De modo a reduzir o custo do problema de otimização, pode-se calcular o fecho convexo dado pelos vetores linha e manter apenas essas linhas para o cálculo da transformação (uma vez que o simplex de menor volume a conter o fecho convexo também será o simplex de menor volume a conter todos os pontos). Note que tal passo não é obrigatório, e a transformação pode ser calculada utilizando-se todos os pontos.

O alvo dessa transformação é encontrar uma matriz K tal que

$$U = \tilde{U}K,$$

tal que a matriz \tilde{U} seja SN e NN.

Isso é o mesmo que impor que as linha de \tilde{U} possuam a *propriedade de soma convexa* e impor que

$$\text{sum}(UT) = \mathbf{1},$$

$$UT \succeq 0,$$

em que $T = K^{-1}$, e a primeira restrição significa que a soma das linhas de UT deve ser igual a um vetor de uns.

Por outro lado, o volume v de um simplex $r - 1$ dimensional descrito por r vértices é dado por

$$v = \frac{1}{(r-1)!} \det \left(\begin{bmatrix} k_{11} & k_{12} & \dots & k_{1(r-1)} & 1 \\ k_{21} & k_{22} & \dots & k_{2(r-1)} & 1 \\ \vdots & \vdots & \vdots & \vdots & 1 \\ k_{r1} & k_{r2} & \dots & k_{r(r-1)} & 1 \end{bmatrix} \right).$$

Mas, como as linhas de K devem satisfazer a *propriedade de soma convexa*, pode-se escrever

$$v = \frac{1}{(r-1)!} \det \left(K \begin{bmatrix} I_{r-1} & \mathbf{1}_{r-1} \\ \mathbf{0}_{r-1}^T & 1 \end{bmatrix} \right) = \frac{1}{(r-1)!} \det(K).$$

Logo, de modo a encontrar o menor simplex que cobre os pontos nas linhas de U , queremos minimizar

$$J = \det(K) = \det(T)^{-1}.$$

Podemos então enunciar o resultado abaixo.

Teorema C.2: Transformação CNO

A matriz de transformação T que dá a uma matriz de peso U a característica CNO pode ser encontrada resolvendo-se o seguinte problema de otimização

$$\begin{aligned} \min_T \quad & -\ln(\det(T)) \\ \text{s.a.} \quad & \text{sum}(UT) = \mathbf{1}, \\ & UT \succeq 0. \end{aligned}$$

Exemplo C.5. Considerando ainda as matrizes de peso obtidas no exemplo C.3 e aplicando a transformação explicada acima para o cálculo da matriz de transformação CNO utilizando otimização, obtêm-se as funções de peso apresentadas na Figura C.4.

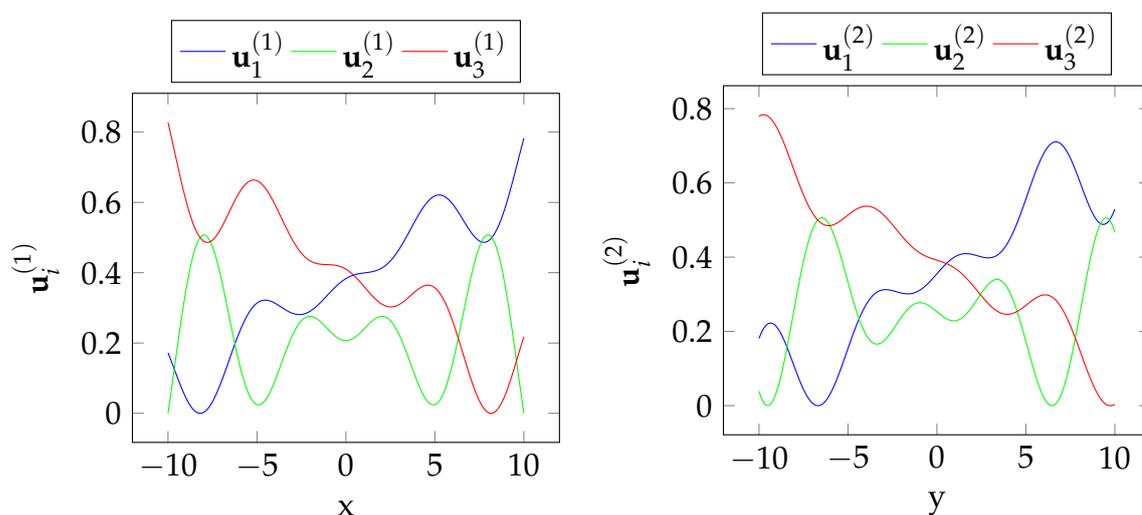


Figura C.4: funções de peso CNO obtidas no exemplo C.5.

Normalizada Relaxada (RNO) e Normalizada Inversa (INO)

Definição C.11: Normalizada Relaxada (RNO)

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, $\mathbf{u}_i^{(n)}$, são os mesmos. Escrevendo de outra maneira, uma matriz $U^{(n)}$ é dita normalizada relaxada se ela é SN, NN e

$$\max(\mathbf{u}_i^{(n)}) = \max(\mathbf{u}_j^{(n)}), \quad \forall i, j.$$

Definição C.12: Normalizada Inversa (INO)

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, $\mathbf{u}_i^{(n)}$, são os mesmos e iguais a zero. Escrevendo de outra maneira, uma matriz $U^{(n)}$ é dita normalizada se ela é SN, NN e

$$\min(\mathbf{u}_i^{(n)}) = 0, \quad \forall i.$$

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 a 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.

Exemplo C.6. Considerando agora as matrizes de peso obtidas no exemplo C.4, pois a transformação utilizada nesta etapa exige que as matrizes de entrada sejam SN e NN, e aplicando a transformação RNO-INO obtêm-se as funções de peso apresentadas na Figura C.5.

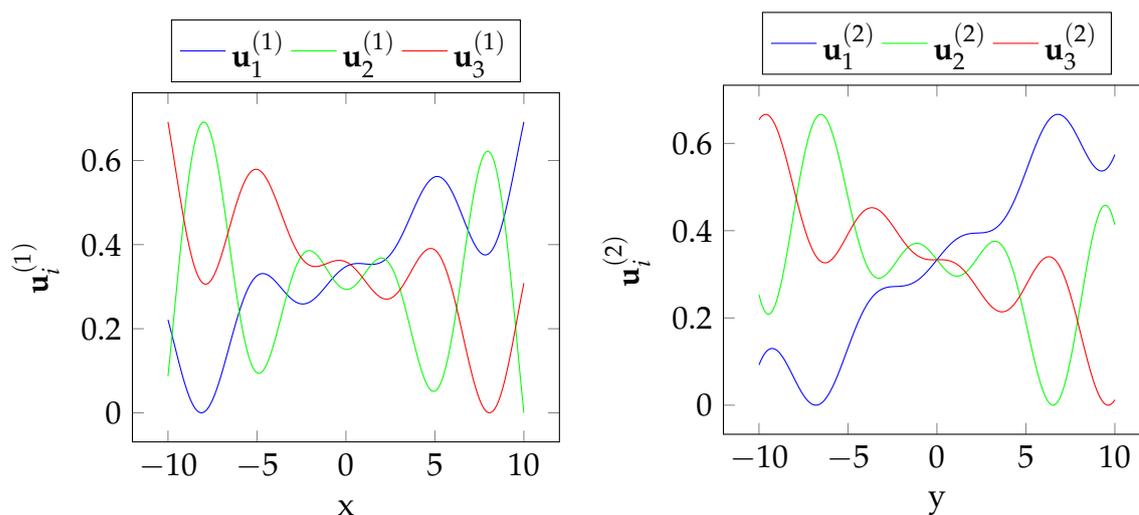


Figura C.5: funções de peso RNO-INO obtidas no exemplo C.6.

É interessante notar que, conforme comentado acima, encontraram-se funções que parecem levar a uma representatividade similar para os vértices correspondentes a todas as colunas. Um pouco diferente do que ocorreu com a transformação CNO (em

que aparentemente alguns vértices têm menor representatividade do que outros) e justificando o uso da transformação RNO-INO para certos casos.

É interessante notar que todas essas representações são equivalentes, e portanto geram a mesma aproximação para a função $f(x, y)$. Para fins de informação a Figura C.6 apresenta o erro quadrático de aproximação obtido ao se utilizar as matrizes de peso RNO-INO (que é o mesmo de se utilizando as outras representações).

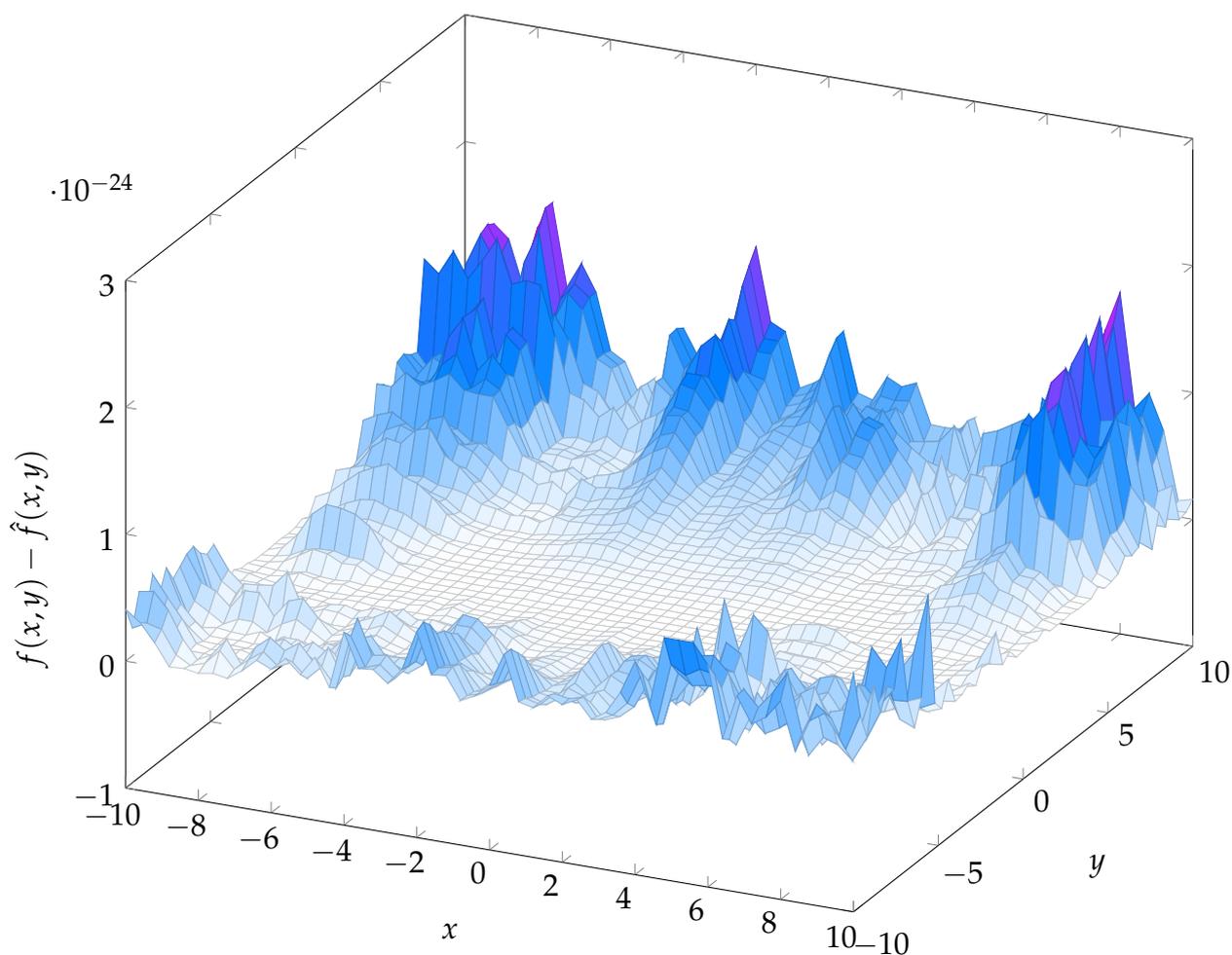


Figura C.6: Erro quadrático de aproximação ao se utilizar as matrizes de peso do exemplo C.6.

Aproximações para as matrizes de peso

Tendo as matrizes de peso com as características desejadas é necessário que se encontre funções que substituam as colunas de cada matriz de peso, de modo que a aproximação encontrada possa ser utilizada para pontos que não pertençam ao grid.

Da equação (C.1):

$$\begin{aligned}
 f(\mathbf{x}) &\approx \mathcal{S}_d \times_{n=1}^N \boldsymbol{\lambda}^{(n)}(x_n), \\
 f(\mathbf{x}) &\approx \left(\mathcal{S} \times_{n=1}^N U^{(n)} \right) \times_{n=1}^N \boldsymbol{\lambda}^{(n)}(x_n), \\
 f(\mathbf{x}) &\approx \mathcal{S} \times_{n=1}^N (\boldsymbol{\lambda}^{(n)}(x_n) U^{(n)}), \\
 f(\mathbf{x}) &\approx \mathcal{S} \times_{n=1}^N \mathbf{u}^{(n)}(x_n), \tag{C.2}
 \end{aligned}$$

em que $\mathbf{u}^{(n)}(x_n)$ é um vetor linha cujos elementos são dados pela interpolação das colunas de $U^{(n)}$.

Tal resultado mostra que as mesmas funções de peso que foram consideradas para gerar uma interpolação pelas amostras originais são utilizadas para gerar a interpolação das colunas de cada matriz de peso e gerar as funções de pertinência. Logo, inicialmente, qualquer interpolação polinomial por partes pode ser utilizada nas funções de pertinência. Entretanto, note que apenas as interpolações constante por partes e linear por partes garantem que as funções de pertinência obtidas após a interpolação terão as características desejadas.

É possível reescrever (C.2) como:

$$f(\mathbf{x}) \approx \sum_{i_1=1}^{I_1} \dots \sum_{i_N=1}^{I_N} \mathbf{u}_{i_1}^{(1)}(x_1) \dots \mathbf{u}_{i_N}^{(N)}(x_N) \mathcal{S}_{i_1 \dots i_N},$$

em que $\mathcal{S}_{i_1 \dots i_N}$, assim como $f(\mathbf{x})$ pertence a $\mathbb{R}^{M_1 \times \dots \times M_m}$. Se as matrizes de peso utilizadas para a interpolação possuírem as características SN e NN, essa expressão mostra que $f(\mathbf{x})$ pode ser aproximada como uma combinação convexa dos pontos $\mathcal{S}_{i_1 \dots i_N}$. Nesse caso, tais pontos são chamados vértices da representação convexa.

Exemplo de Aplicação a Sistemas Dinâmicos

No caso de aplicação da técnica da transformação do produto tensorial à sistemas dinâmicos, normalmente tem-se a seguinte situação: dado um modelo qLPV de um sistema dinâmico não-linear, encontrar uma representação fuzzy Takagi-Sugeno do sistema.

Considere o modelo qLPV de um sistema dado por:

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

em que $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, $\mathbf{y} \in \mathbb{R}^k$, $A(\mathbf{x}) \in \mathbb{R}^{n \times n}$, $B(\mathbf{x}) \in \mathbb{R}^{n \times m}$, $C(\mathbf{x}) \in \mathbb{R}^{k \times n}$ e $D(\mathbf{x}) \in \mathbb{R}^{k \times m}$.

Uma abordagem possível para a aplicação da técnica seria criar uma função

$$f(\mathbf{x}) = \begin{bmatrix} A(\mathbf{x}) & B(\mathbf{x}) \\ C(\mathbf{x}) & D(\mathbf{x}) \end{bmatrix}$$

e dentro de um espaço hiperretangular definido com limites nos valores de \mathbf{x} , encontrar uma aproximação para ela assim como se fez na seção anterior. Dessa maneira encontra-se uma representação fuzzy Takagi-Sugeno para o sistema.

Nesse caso, $\mathcal{S}_{i_1 \dots i_N}$ representam as matrizes dos sistemas lineares vértices que compõem o modelo fuzzy Takagi-Sugeno do sistema.

Entretanto, na maioria das vezes, não são todos os estados em \mathbf{x} que afetam os valores das matrizes A , B , C e D e em alguns casos algumas das matrizes podem ser constantes.

Sendo assim, na maioria dos casos, é útil encontrar o subconjunto de estados $\check{\mathbf{x}}$ que influenciam as matrizes do sistema e encontrar uma aproximação para uma função $f(\check{\mathbf{x}})$ reduzindo o tamanho final do grid de amostragem. Além disso, para fins de redução da computação, formar tal função contendo apenas as matrizes que não são constantes.

Quando não se incluem todas as matrizes para formar a função $f(\check{\mathbf{x}})$, $\mathcal{S}_{i_1 \dots i_N}$ contém apenas as matrizes que foram incluídas. Porém, como as outras matrizes do sistema são constantes, elas são as mesmas para todos os sistemas vértice.

Exemplo C.7. Considere as equações cinemáticas de um robô móvel com movimento restringido ao plano xy :

$$\begin{aligned} \dot{x} &= v \cos(\theta), \\ \dot{y} &= v \sin(\theta), \\ \dot{\theta} &= w. \end{aligned}$$

Em que x é a posição do robô na direção x , y é a posição do robô na direção y , θ é o ângulo de guinada do robô, v é a velocidade de translação e w é a velocidade de rotação. Considerando que o vetor de estados do sistema é dado por $\mathbf{z} = [x \ y \ \theta]^T$, que o vetor de entradas é dado por $\mathbf{u} = [v \ w]^T$, e que o vetor de saídas é dado por $\mathbf{p} = [x \ y]^T$, pode-se escrever:

$$\begin{aligned} \dot{\mathbf{z}} &= B(\mathbf{z})\mathbf{u} \\ \mathbf{p} &= C\mathbf{z} \\ B(\mathbf{z}) &= \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \\ C &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \end{aligned}$$

No caso deste exemplo pode-se ver o que foi comentado acima. Das três variáveis que compõem o vetor de estados, apenas uma delas, θ , influencia as matrizes do sistema. Além disso, temos também que apenas a matriz B varia com este estado. As matrizes A e D são matrizes de zeros e por isso não aparecem nas equações acima, e a matriz C é uma matriz constante.

Sendo assim, a função que devemos aproximar nesse exemplo é $B(\theta)$ e sabemos que as outras matrizes serão as mesmas para os todos os sistemas vértices que forem encontrados no modelo final.

Como θ é o ângulo de guinada do robô móvel, sabe-se que $\theta \in [-\pi, \pi]$. Decidiu-se então por se utilizar um grid de 101 pontos para $\theta \in [-\pi, \pi]$. No caso deste exemplo, temos que $B(\theta) \in \mathbb{R}^{3 \times 2}$ e, portanto, com 101 amostras teremos um tensor $\mathcal{S}_d \in \mathbb{R}^{101 \times 3 \times 2}$.

Executando os mesmos passos da técnica explicados na seção anterior, é possível chegar em várias representação para o modelo. As funções de peso encontradas em cada representação estão apresentadas na Figura C.7.

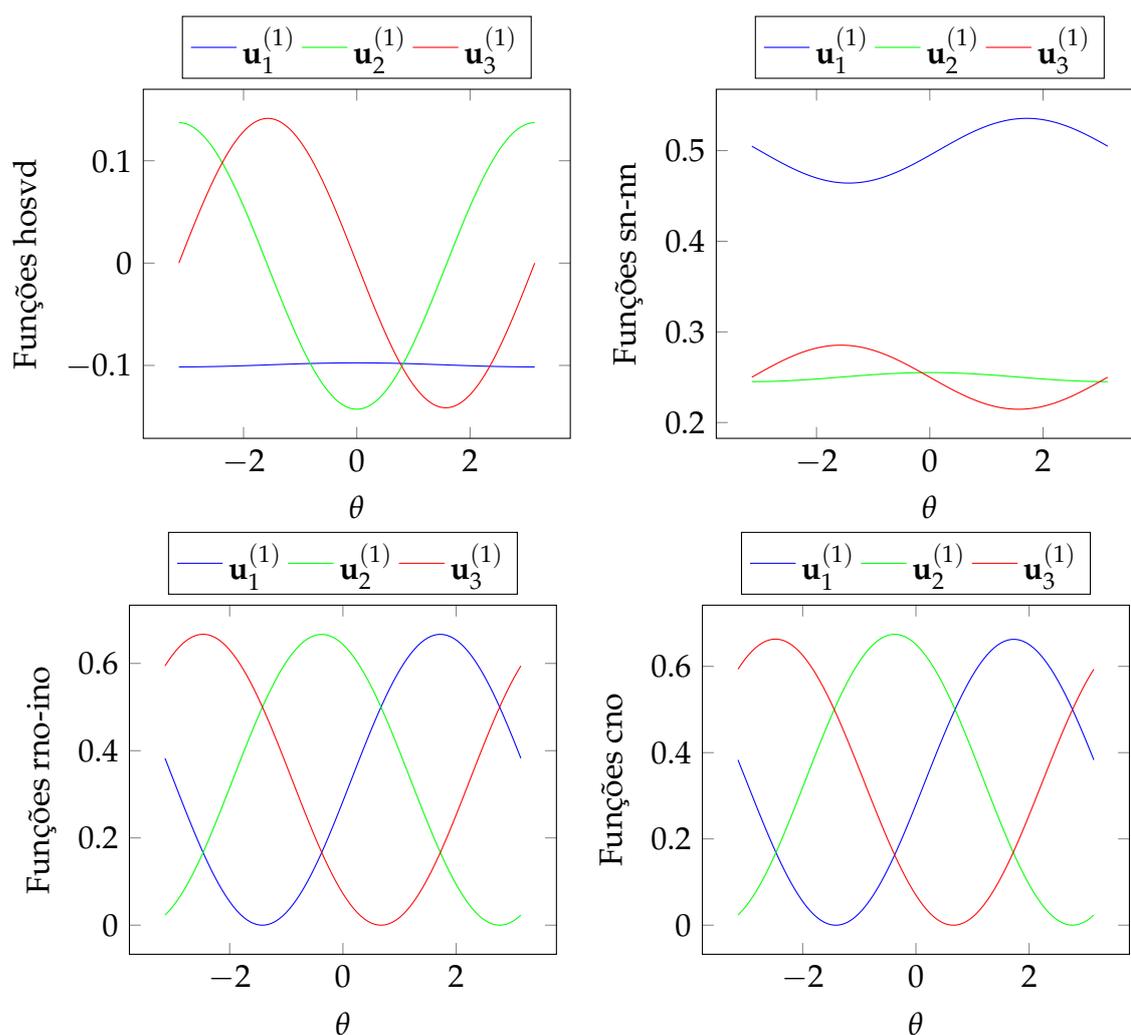


Figura C.7: diferentes funções de peso obtidas no exemplo C.7.

Como todas as representações obtidas são equivalentes, elas possuem o mesmo erro de representação para os pontos pertencentes ao grid. Nesse exemplo, tal erro é de $1,2321 \times 10^{-25}$.