SEL0364 - Controle Não Linear Aplicado
Lógica *fuzzy*

Profª Vilma Alves de Oliveira
Monitor: Lucas Jonys

Escola de Engenharia de São Carlos
Universidade de São Paulo

Maio de 2023

Introdução

Conceitos iniciais

- Conceito: É a lógica baseada em análises de informações qualitativas. Isto é feito de forma que a decisão não se resuma entre um ‘sim’ e um ‘não’, mas, também considera abstrações do tipo ‘próximo de’, ‘em torno de’, ‘muito alto’, ‘bem baixo’, etc.

- A ideia da lógica *fuzzy* é refletir como as pessoas pensam, modelando o seu senso de palavra e tomadas de decisão. Assim, a introdução da lógica *fuzzy* tem conduzido as pesquisas para sistemas inteligentes mais humanos e mais adequados à realidade.

Grau de pertinência

- A pertinência de um elemento x em relação a um conjunto $\mathcal{F}$ qualquer, geralmente representada pela letra μ, varia entre 0 e 1.

- O valor 0 indica que x não pertence a $\mathcal{F}$, enquanto que 1 representa completa pertinência de x a $\mathcal{F}$.

- Valores de pertinência entre 0 e 1 indicam pertinências parciais. Isto é, uma afirmação pode ser parcialmente verdadeira e parcialmente falsa.

Grau de pertinência

- Exemplo: estatura de uma pessoa:

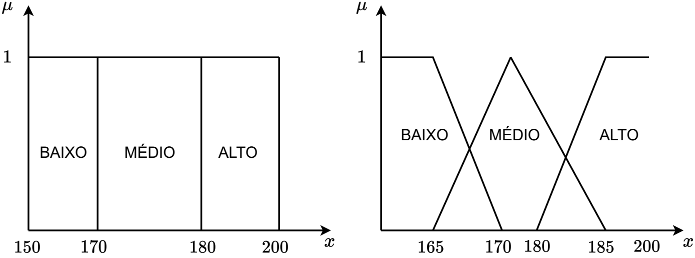

Figura 1: Representação da altura de uma pessoa na forma de conjuntos de acordo com a lógica clássica (à esquerda) e com a lógica *fuzzy* (à direita).

Conjuntos *Fuzzy*

Conjuntos *fuzzy*

- Na lógica clássica, os conjuntos são bem definidos. Assim, um dado elemento pertence ou não a um conjunto. Além disso, ele só pode pertencer a um único conjunto.

- Na lógica *fuzzy*, a definição de conjuntos está relaciona ao que é conhecido como função de pertinência.

Definição 1

Um conjunto *fuzzy* A em um universo de discurso $\mathcal{X}$ é caracterizado por uma função de pertinência μ_A que assume valores no intervalo $[0,\ 1]$:

$$\mu_A(x) \in [0,\ 1]\,, \quad \forall x \in \mathcal{X} \tag{1}$$

Conjuntos *fuzzy*

- As funções de pertinência de um conjunto pode assumir diferentes formas, dependendo do problema a ser modelado. Alguns exemplos são exibidos na Figura 2.

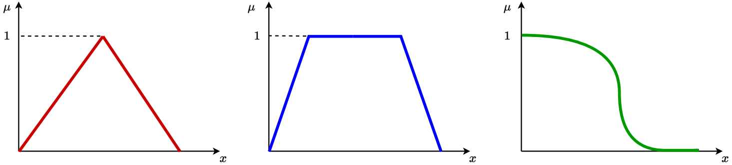

Figura 2: Exemplos de funções de pertinência.

- As funções de pertinência são uma forma contínua de representação de um conjunto *fuzzy*.

Conjuntos *fuzzy*

- Contudo, esta representação também pode ser feita na forma discreta. Para um conjunto *fuzzy* A qualquer composto por elementos x em um universo de discurso $\mathcal{X}$, as notações discretas são:

  - $A = \{x,\ \mu_A(x)\}\,,\ \ x \in \mathcal{X}$

    $$A = \{(0,\ 30), (0.3,\ 35), (1.0,\ 40), (1.0,\ 45), (0.7,\ 50), (0.4,\ 55), (0,\ 60)\}$$

  - $A = \left\{\frac{\mu_A(x)}{x}\right\},\ \ x \in \mathcal{X}$

    $$A = \left\{\left(\frac{30}{0}\right), \left(\frac{35}{0,3}\right), \left(\frac{40}{1,0}\right), \left(\frac{45}{1,0}\right), \left(\frac{50}{0,7}\right), \left(\frac{55}{0,4}\right), \left(\frac{60}{0}\right)\right\}$$

  - $A = \{\mu_A(x)\}\,,\ \ x \in \mathcal{X}$

    $$A = \{0,\ \ 0.3,\ \ 1.0,\ \ 1.0,\ \ 0.7,\ \ 0.4,\ \ 0\}$$

Operações em conjuntos *fuzzy*

- Assim como na lógica clássica, os conjuntos *fuzzy* podem ser operados de diversas maneiras. Sejam os conjuntos *fuzzy* a seguir:

$$A = \{x,\ \mu_A(x)\},\quad x \in \mathcal{X} \tag{2}$$
$$B = \{x,\ \mu_B(x)\},\quad x \in \mathcal{X} \tag{3}$$

  - Interseção (operação mínimo):

$$\mu_A(x) \cap \mu_B(x) = min(\mu_A(x), \mu_B(x)) \tag{4}$$

  - União (operação máximo):

$$\mu_A(x) \cup \mu_B(x) = max(\mu_A(x), \mu_B(x)) \tag{5}$$

  - Complemento:

$$\bar{\mu}_A(x) = 1 - \mu_A(x) \tag{6}$$
$$\bar{\mu}_B(x) = 1 - \mu_B(x) \tag{7}$$

Operações em conjuntos *fuzzy*: exemplo

- Considere os três conjuntos *fuzzy* a seguir:

$$A = \{x,\ \mu_A(x)\} = \{(1,\ 1), (2,\ 0.6), (3,\ 0.2), (4,\ 0), (5,\ 0)\} \tag{8}$$
$$B = \{x,\ \mu_B(x)\} = \{(0,\ 1), (2,\ 0.5), (3,\ 1), (4,\ 0.5), (5,\ 0)\} \tag{9}$$
$$C = \{x,\ \mu_C(x)\} = \{(1,\ 0), (2,\ 0), (3,\ 0.1), (4,\ 0.5), (5,\ 1)\} \tag{10}$$

- Interseção (operação mínimo):

$$A \cap B = min(\mu_A(x), \mu_B(x)) = \{(1,\ 1), (2,\ 0.5), (3,\ 0.2), (4,\ 0), (5,\ 0)\} \tag{11}$$

- União (operação máximo):

$$B \cup C = max(\mu_B(x), \mu_C(x)) = \{(1,\ 1), (2,\ 0.5), (3,\ 1), (4,\ 0.5), (5,\ 1)\} \tag{12}$$

- Complemento:

$$\bar{A} = 1 - \mu_A(x) = \{(1,\ 0), (2,\ 0.4), (3,\ 0.8), (4,\ 1), (5,\ 1)\} \tag{13}$$
$$\bar{B} = 1 - \mu_B(x) = \{(1,\ 0), (2,\ 0.5), (3,\ 0), (4,\ 0.5), (5,\ 1)\} \tag{14}$$

Variáveis linguísticas

Definição 1

Uma variável linguística (ou *fuzzy*) é uma entidade que representa uma variável de um certo problema de forma linguística, ou seja, seus valores são palavras. Uma variável linguística é caracterizada por seu nome e função de pertinência.

- Uma problema *fuzzy* utiliza como entrada uma variável x qualquer, cujas variáveis linguísticas são "Pequeno", "Médio" e "Grande", todas com funções de pertinência triangulares, como exibido na Figura 3.

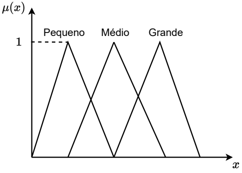

Figura 3: Entrada x de um problema *fuzzy* com três variáveis linguísticas.

Regras *Fuzzy*

Regras *fuzzy*

- As regras *fuzzy* são regras que operam conjuntos *fuzzy*, cada um com suas variáveis linguísticas definidas, de forma a obter consequentes.
- A criação das regras deve ser feita de maneira coerente com o problema trabalhado e com o resultado que se busca obter.
- Funcionamento de uma regra *fuzzy*: primeiro avalia-se o antecedente da regra e então aplica-se o resultado no consequente.

Regras *fuzzy*

- Exemplo: considere um sistema *fuzzy* cuja entrada seja a velocidade média de um carro, enquanto que a saída é o tempo gasto em um trajeto.

- As variáveis linguísticas da entrada e saída são:
  - Velocidade média: Baixa, Média e Alta.
  - Tempo: Baixo, Médio e Alto.

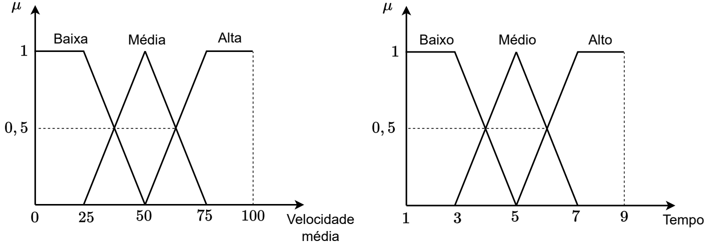

Figura 4: Funções de pertinência.

Regras *fuzzy*

- Exemplos de regras *fuzzy*:
  - Se a velocidade média é Baixa então o tempo é Alto.
  - Se a velocidade média é Média então o tempo é Médio.
  - Se a velocidade média é Alta então o tempo é Baixo.

- Vamos analisar individualmente cada uma das regras do exemplo considerando o valor Velocidade média $= 60\ km/h$.

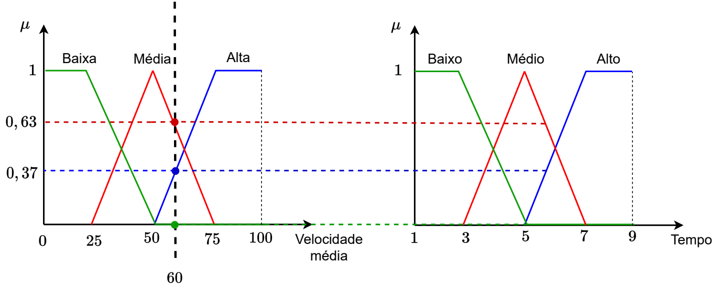

Figura 5: Análise individual de cada uma das regras.

Sistema *Fuzzy*

Estrutura de um sistema *fuzzy*

- O primeiro passo no desenvolvimento de um sistema *fuzzy* é a definição das variáveis de entrada e saída, bem como seu universo de discurso, variáveis linguísticas e funções de pertinência.

- É importante lembrar que essas escolhas variam conforme o problema, e o conhecimento e experiência do projetista refletem em um melhor desempenho do sistema *fuzzy* projetado.

- A estrutura completa de um sistema *fuzzy* pode ser dividida em três partes:
  - Fuzzificação;
  - Processo de inferência *fuzzy*;
    - Avaliação das regras *fuzzy*
    - Agregação das regras *fuzzy*
  - Deffuzificação.

Estrutura de um sistema *fuzzy*

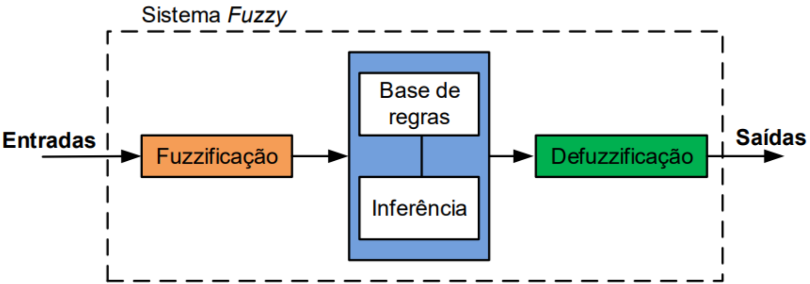

Figura 6: Etapas da construção de um sistema *fuzzy*.

Sistema *fuzzy*

- Para o entendimento de sistema *fuzzy*, as etapas serão explicada considerando um exemplo para um sistema de acionamento de freio.

- As variáveis de entrada são a **velocidade do carro (V)** e a **distância até o obstáculo (D)**, enquanto que a variável de saída é a **força no pedal de freio (F)**.

Sistema *fuzzy*

- As variáveis linguísticas das entradas e saídas são:
  - **Velocidade do carro**: Baixa (B), Média (M) e Alta (A).
  - **Distância até o obstáculo**: Pequena (P) e Grande (G).
  - **Força no pedal de freio**: Força pequena (FP), Força média (FM) e Força grande (FG).

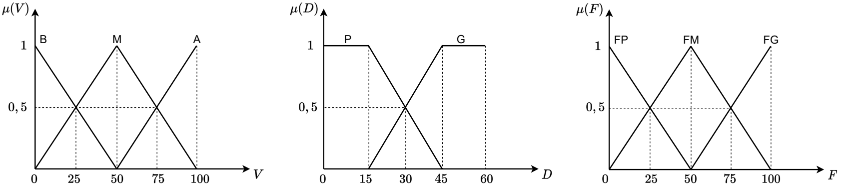

Figura 7: Funções de pertinências das variáveis linguísticas de entrada e saída do sistema *fuzzy* para acionamento de um pedal de freio.

Sistema *fuzzy*

- Um certo especialista fora consultado para criar a base de regras:
  - R1: Se a Velocidade é **B** e a Distância é **P**, então a Força é **FP**.
  - R2: Se a Velocidade é **B** e a Distância é **G**, então a Força é **FP**.
  - R3: Se a Velocidade é **M** e a Distância é **P**, então a Força é **FM**.
  - R4: Se a Velocidade é **M** e a Distância é **G**, então a Força é **FP**.
  - R5: Se a Velocidade é **A** e a Distância é **P**, então a Força é **FG**.
  - R6: Se a Velocidade é **A** e a Distância é **G**, então a Força é **FM**.

Tabela 1: Base de regras *fuzzy* desenvolvida.

		D	
		P	G
V	B	FP	FP
	M	FM	FP
	A	FG	FM

Superfície *fuzzy*

- Para todas as possíveis combinações de entradas o sistema *fuzzy* fornecerá um respectivo valor em sua saída, dependente das funções de pertinência e regras adotadas.
- Para um sistema *fuzzy* com n entradas, estas combinações podem ser vistas em uma superfície de dimensão $n + 1$.
- Neste exemplo, a superfície é um gráfico tridimensional.

Análise da operação de um sistema *fuzzy*

- Vamos analisar o sistema *fuzzy* para o acionamento de freio em uma situação na qual as entradas do sistema são $V = 60$ e $D = 25$.

Fuzzyficação

- O primeiro passo é a fuzzyficação, etapa na qual obtém-se o grau de pertinência de cada entrada em relação às variáveis linguísticas. Para as entradas $V = 60$ e $D = 20$, o grau de pertinência em relação às variáveis linguísticas são:

$$\mu_B(V = 60) = 0,0$$
$$\mu_M(V = 60) = 0,7$$
$$\mu_A(V = 60) = 0,3$$

$$\mu_P(D = 20) = 0,8$$
$$\mu_G(D = 20) = 0,2$$

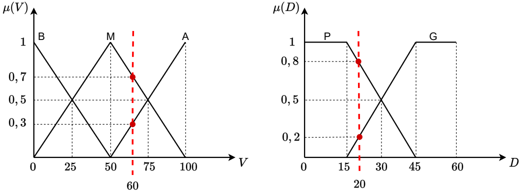

Figura 8: Fuzzyficação das variáveis para o exemplo do freio.

Inferência: avaliação das regras

- Por meio dos valores das entradas é possível verificar quais regras foram ativadas.
- Para a entrada $V = 60$, apenas as variáveis linguísticas M e A possuem pertinências não nulas. Para a entrada $D = 20$, isso ocorre para ambas as variáveis linguísticas, P e G.
- Portanto, 4 regras das 6 regrais totais foram ativadas, sendo elas as regras R3, R4, R5 e R6, destacadas na Tabela 2.

Tabela 2: Conjunto de regras ativadas para $V = 60$ e $D = 20$.

		D	
		P	G
V	B	FP	FP
	M	FM	FP
	A	FG	FM

Inferência: avaliação das regras

- Cada uma das regras ativadas é analisada de forma individual. No caso de antecedentes composto pelo operador *E*, seleciona-se o menor grau de pertinência (interseção), enquanto que para o operador *OU* seleciona-se o maior grau de pertinência (união).

$$R3 \longrightarrow \mu_{MP} = min\,\{\mu_M(V = 60),\; \mu_P(D = 20)\}$$
$$\mu_{MP} = min\,\{0.7,\; 0.8\} = 0.7$$

$$R4 \longrightarrow \mu_{MG} = min\,\{\mu_M(V = 60),\; \mu_G(D = 20)\}$$
$$\mu_{MG} = min\,\{0.7,\; 0.2\} = 0.2$$

$$R5 \longrightarrow \mu_{AP} = min\,\{\mu_A(V = 60),\; \mu_P(D = 20)\}$$
$$\mu_{AP} = min\,\{0.3,\; 0.8\} = 0.3$$

$$R6 \longrightarrow \mu_{AG} = min\,\{\mu_A(V = 60),\; \mu_G(D = 20)\}$$
$$\mu_{AG} = min\,\{0.3,\; 0.2\} = 0.2$$

Inferência: avaliação das regras

- Uma vez avaliado os antecedentes da regra e obtido o grau de pertinência, deve-se verificar o valor do consequente. Para isso, o consequente é "cortado" para o nível de valor verdade do antecedente da regra avaliada.

- Em outras palavras, o grau de pertinência de saída de cada regra é dado pelo operador E entre o grau de pertinência único dos antecedentes e o grau de pertinência do consequente.

$$R3 \longrightarrow \mu_{FM_{R3}} = min\,\{\mu_{MP},\ \mu_{FM}\}$$
$$R4 \longrightarrow \mu_{FP_{R4}} = min\,\{\mu_{MG},\ \mu_{FP}\}$$
$$R5 \longrightarrow \mu_{FG_{R5}} = min\,\{\mu_{AP},\ \mu_{FG}\}$$
$$R6 \longrightarrow \mu_{FM_{R6}} = min\,\{\mu_{AG},\ \mu_{FM}\}$$

Inferência: agregação das regras

- Uma vez avaliadas individualmente cada uma das regras e obtido a pertinência para as saídas, o próximo passo é fazer a agregação delas..

- Para a agregação, utiliza-se a regra OU.

$$R = R1 \cup R2 \cup R3 \cup R4$$

$$R = max\left\{\mu_{FM_{R3}},\ \mu_{FP_{R4}},\ \mu_{FG_{R5}},\ \mu_{FM_{R6}}\right\}$$

Inferência: avaliação e agrupamento das regras

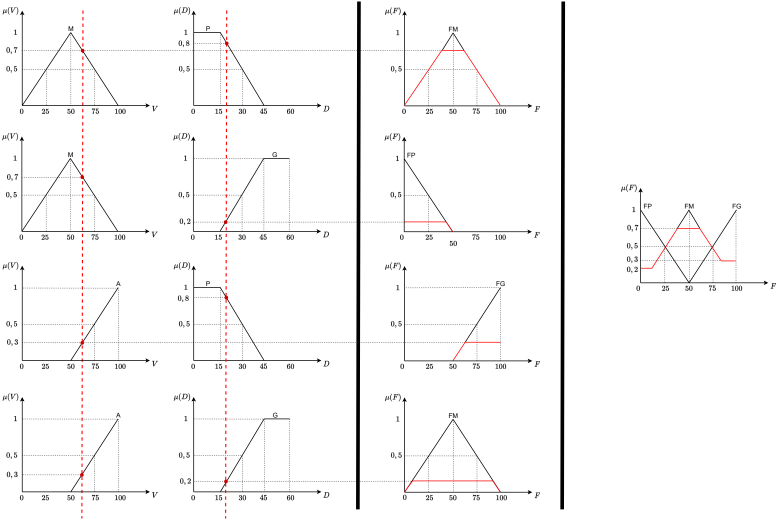

Figura 9: Avaliação e agregação das regras do sistema *fuzzy* de freio.

Defuzzyficação

- A defuzzyficação consiste na obtenção de um valor numérico de saída a partir do conjunto *fuzzy* resultante da agregação das regras.

- Existem diversos métodos: primeiro dos máximos, último dos máximos, centro de área, sendo este último o que mais se destaca e o que é mais utilizado.

$$CDA = \frac{\sum_{k=1}^{N} \mu(x_k) x_k}{\sum_{k=1}^{N} \mu(x_k)}, \tag{15}$$

sendo N o número de discretizações do universo de discurso.

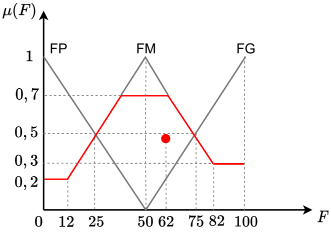

Figura 10: Defuzzyficação pelo método do centro de área.

Sistema *Fuzzy*

Tarefa: construção de um sistema *fuzzy*

A quantidade de sabão em pó aplicada em uma lavagem de roupa em uma máquina é escolhida pelo usuário com base no estado e quantidade de roupas inseridas. O excesso de sabão leva a um desperdício e maior gasto do usuário, enquanto uma falta de sabão prejudica a qualidade do serviço.

Dessa maneira, o objetivo desta tarefa é desenvolver um sistema com lógica *fuzzy* para determinar de maneira automática a quantidade de sabão em pó a ser inserida na máquina.

Como entradas do problema podem ser utilizados o peso total das roupas a serem lavadas e quão sujas elas estão, sendo a saída definida como a quantidade de sabão a ser utilizado. Outras variáveis de entrada podem ser utilizadas conforme desejo do usuário.

Controlador *Fuzzy*

Controlador *fuzzy*

- A estrutura *fuzzy* pode ser utilizada para o controle de diferentes processos. Para isso, deve-se definir as entradas e saídas do processo de controle, variáveis linguísticas, regras e defuzzyficação, assim como descrito anteriormente.

- Geralmente, adota-se como entradas do processo de inferência de um controlador o sinal de erro (e_P) e sua derivada (e_D), sendo o erro a diferença entre o sinal de referência (x_{ref}) e x o sinal realimentado.

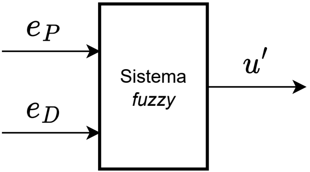

Figura 11: Estrutura de um sistema *fuzzy* para operação no controlador *fuzzy*.

Controlador *fuzzy*

- Geralmente, constrói-se um sistema *fuzzy* linear, de forma a emular um controlador linear clássico, adicionando, em seguida, não linearidades.

- Para construir um sistema *fuzzy* linear, algumas características das variáveis de entrada, funções de pertinência e regras devem ser seguidas.

  - Universo de discurso das entradas e saídas entre $[-1 \;\; 1]$;
  - Base de regras contendo todas as combinações possíveis das entradas;
  - Funções de pertinências triangulares para as entradas, sendo igualmente espaçadas e se cruzando em pontos nos quais o valor da pertinência é 0,5;
  - Uso de multiplicação para o conectivo E;
  - Funções de pertinência para a saída do tipo *singletons* posicionadas nos picos das funções de pertinência da entrada e nos pontos onde a pertinência das entradas corresponde à $\mu = 0, 5$;
  - Acumulação por soma;
  - Método de *defuzzyficação* por centro de área.

Entradas e saídas de um *fuzzy* linear (FL)

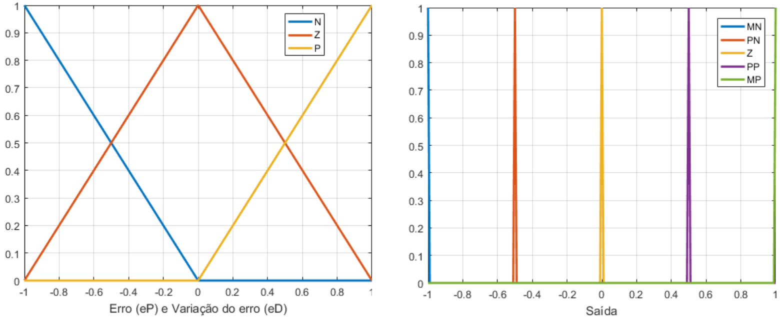

Figura 12: Variáveis linguísticas e funções de pertinência para as entradas e saídas do sistema *fuzzy* linear.

Regras de um *fuzzy* linear (FL)

Tabela 3: Regras do sistema *fuzzy* linear.

		e_D		
		N	Z	P
e_P	N	MN	PN	Z
	Z	PN	Z	PP
	P	Z	PP	MP

Superfície de um *fuzzy* linear (FL)

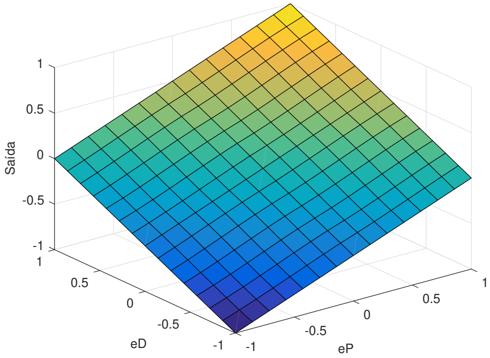

Figura 13: Superfície de um sistema *fuzzy* linear.

Controladores *fuzzy*

- Diversas estruturas diferentes de controladores *fuzzy* podem ser aplicadas a um problema de controle, a depender do objetivo a ser alcançado.

- As diferentes topologias são desenvolvidas de forma análoga aos controladores clássicos:
  - Controlador proporcional *fuzzy* (análogo ao controlador proporcional)
  - Controlador PD *fuzzy* (análogo ao PD)
  - Controlador *fuzzy* Pincremental (análogo ao PI)
  - Controlador *fuzzy* PD+I (análogo ao PID)

Controlador proporcional *fuzzy*

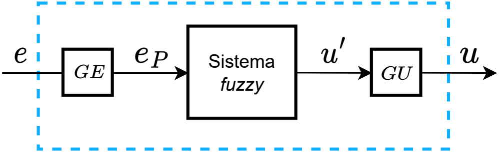

Figura 14: Estrutura do controlador proporcional *fuzzy*.

$$
\begin{aligned}
u &= GUu' \\
u' &= f(e_P) \\
e_P &= GEe \\
u &= GUf(GEe)
\end{aligned}
\tag{16}
$$

Controlador PD *fuzzy*

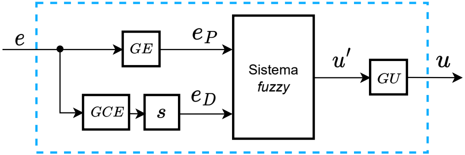

Figura 15: Estrutura do controlador PD *fuzzy*.

$$
\begin{aligned}
u &= GUu' \\
u' &= f(e_P,\ e_D) \\
e_P &= GEe \\
e_D &= sGCEe \\
u &= GUf(GEe,\ \ sGCEe)
\end{aligned}
\tag{17}
$$

Controlador *fuzzy* Pincremental

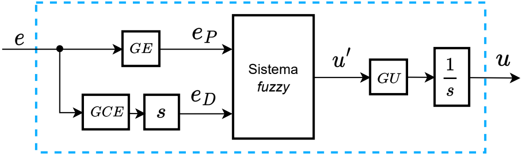

Figura 16: Estrutura do controlador *fuzzy* Pincremental.

$$u = \frac{1}{s} GU u'$$
$$u' = f(e_P,\ e_D)$$
$$e_P = GEe$$
$$e_D = sGCEe$$
$$u = \frac{1}{s} GU f(GEe,\ \ sGCEe) \tag{18}$$

Controlador *fuzzy* PD+I

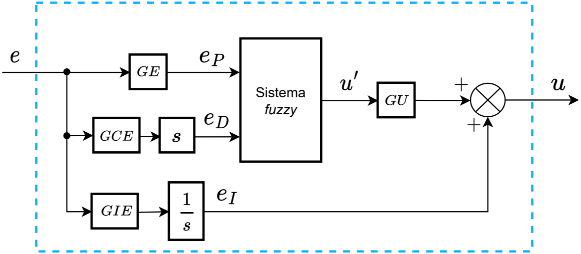

Figura 17: Estrutura do controlador *fuzzy* PD+I.

$$u = GUu' + \frac{1}{s}GIEe$$

$$u' = f(e_P,\ e_D)$$

$$e_P = GEe$$

$$e_D = sGCEe$$

$$u = GUf(GEe,\ sGCEe) + \frac{1}{s}GIEe \tag{19}$$

Analogia entre os controladores *fuzzy* e os clássicos

- Apesar de não ser o objetivo, é possível projetar os controladores *fuzzy* para operarem como seus respectivos controladores clássicos.
- Para isso, a estrutura do sistema *fuzzy* linear deve ser utilizada e os ganhos do controlador *fuzzy* devem ser escolhidos de forma sistemática.
- O sistema *fuzzy* com as entradas e_P e e_D linear apresentado produz uma saída dada pela combinação linear de suas entradas, isto é:

$$u' = f(e_P,\, e_D) = k(e_P + e_D), \tag{20}$$

sendo k um ganho associado ao *fuzzy*.

Analogia entre os controladores *fuzzy* e os clássicos

- Quando o sistema *fuzzy* é linear, conforme as características das funções de pertinência e regras apresentadas, o ganho é $k = 1/2$. Portanto:

$$f(e_P,\ e_D) = \frac{1}{2}(e_P + e_D) \tag{21}$$

$$f(e_P,\ e_D) = \frac{1}{2}(GEe + sGCEe) \tag{22}$$

- A partir disso, pode-se comparar as equações do controlador *fuzzy* com seu respectivo controlador clássico para obter a comparação entre seus ganhos.

- A seguir, a equivalência de ganhos entre o controlador PID e o controlador *fuzzy* PD+I é apresentada.

Analogia entre os controladores *fuzzy* e os clássicos

- Substituindo $f(e_P,\ e_D)$ de (22) em, a saída do *fuzzy* PD+I é dada por:

$$u = GU\frac{1}{2}(GEe + sGCEe) + \frac{1}{s}GIEe$$

$$u = \left(\frac{GU\ GE}{2} + \frac{GU\ GCEs}{2} + \frac{1}{s}GIE\right)e \tag{23}$$

- Sendo a saída do PID clássico $u = K_p + K_i/s + K_d s$, pode-se encontrar a relação entre os ganhos:

$$K_p = \frac{GU\ GE}{2} \tag{24}$$

$$K_i = GIE \tag{25}$$

$$K_d = \frac{GU\ GCE}{2} \tag{26}$$

Analogia entre os controladores *fuzzy* e os clássicos

- Para validação das equações anteriores, considere a planta:

$$G(s) = \frac{3}{s+1}. \tag{27}$$

- Seja um controlador PID com $K_p = 1$, $K_i = 1$ e $K_d = 0.1$, com filtro de ordem $N = 1$ no derivativo, para manter a saída em $y(t) = 10$, com uma redução para $y(t) = 5$ em $t = 10\ s$. Escolhendo $GE = 1/10$ pode-se utilizar as equações de equivalência de ganhos:

$$K_i = GIE = 0.1 \tag{28}$$

$$K_p = \frac{GU\ GE}{2} \implies GU = \frac{2K_p}{GE} = 20 \tag{29}$$

$$K_d = \frac{GU\ GCE}{2} \implies GCE = \frac{2K_d}{GU} = 0.01 \tag{30}$$

Analogia entre os controladores *fuzzy* e os clássicos

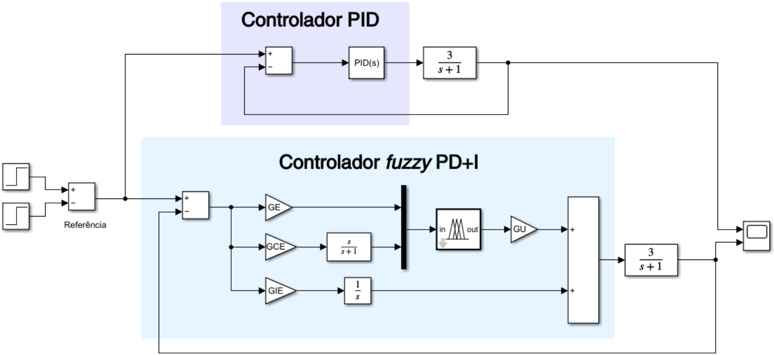

Figura 18: Diagrama de blocos da simulação em Simulink do controlador PID e do *fuzzy* PD+I.

Analogia entre os controladores *fuzzy* e os clássicos

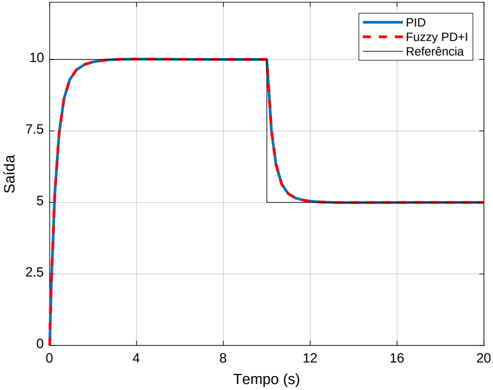

Figura 19: Resposta utilizando o controlador PID e o *fuzzy* PD+I.

Sintonia dos controladores *fuzzy*

- Sintonizar um P/PI/PID (*Ziegler-Nichols*, resposta em frequência, etc)
- Inserir o controlador *fuzzy* análogo
- Transferir os ganhos para o controlador *fuzzy* (se não houver saturação nas entradas e saídas, a resposta deverá ser muito parecida)
- Adicione não linearidades ao controlador *fuzzy* (adaptação de regras, funções de pertinência)
- Ajuste fino
  - GE: melhora a taxa de subida
  - GCE: amortece o sobressinal
  - GIE: remove o erro em regime

Ajuste fino dos controladores *fuzzy*

- Ganho GE:
  - Escolhido com base no erro máximo desejado
  - *Fuzzy* PD: valor tão grande quanto possível reduz problemas com ruídos
  - *Fuzzy* Pincremental: valor grande torna o controle incremental menos estável
  - *Fuzzy* PD+I: valor tão grande quanto possível porém ao custo do aumento do ganho integrativo
- Ganho GCE:
  - *Fuzzy* PD: manter tão pequeno quanto possível reduz problemas com ruídos
  - *Fuzzy* Pincremental: aumento de GCE diminui ação integrativa e incrementa o ganho proporcional, manter GCE tão grande quanto possível para preservar a estabilidade
  - *Fuzzy* PD+I: incremento em GCE incrementa o ganho derivativo, manter tão pequeno quanto possível
- Ganho GCU/GU:
  - afetam o ganho proporcional, deve ser tão grande quanto possível sem criar muito sobressinal. Se pequeno, o sistema pode apresentar-se lento, se muito grande instável.

Referências

Referências

- Slotine, J.J.E. *Applied Nonlinear Control*, Prentice-Hall, 1991.

- Jantzen, Jan. Foundations of fuzzy control. John Wiley Sons, Inc., 2007.