



# **ICAC**

# **Lógica Fuzzy**

# **Laboratório**

Prof. Danilo Perico

# Aula de Hoje



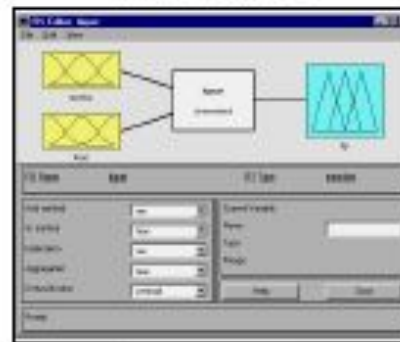
- Construindo um sistema de controle baseado em Lógica Fuzzy.
- Usando o Toolkit Fuzzy do Matlab.

# Lógica Fuzzy no Matlab



- Toolkit Fuzzy têm 5 ferramentas básicas de construção de sistemas Fuzzy:
  - Editor do sistema de inferência (FIS)
  - Editor das Funções de Pertinência
  - Editor de Regras
  - Visualizador da superfície de decisão
  - Visualizador de Regras

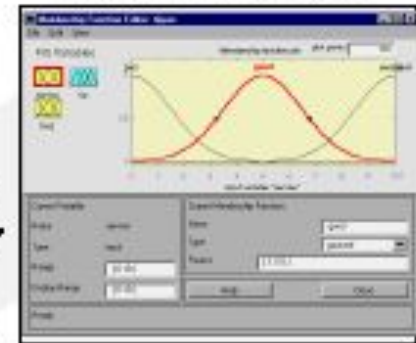
## FIS Editor



## Rule Editor

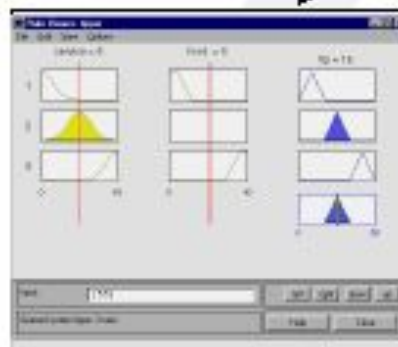


## Membership Function Editor

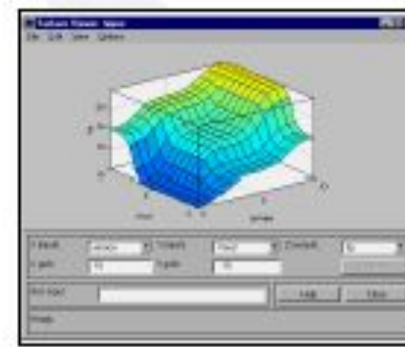


## Fuzzy Inference System

Read-only tools



## Rule Viewer



## Surface Viewer

# ***FIS Editor***



- Fuzzy Inference System (FIS) Editor:
  - Módulo principal do sistema
  - Gerencia assuntos de alto nível:
    - nome das variáveis
    - quantidade de variáveis de entrada e saída
- Junta tudo

# ***Membership Function Editor***



- Editor das Funções de Pertinência
  - Permite definir o formato de todas as funções de pertinência associadas com cada variável

# ***Rule Editor***



- Editor das Regras:
  - Permite editar a lista de regras que define o comportamento do sistema.

# *Surface Viewer*



- Visualizador da superfície de decisão:
  - Permite visualizar graficamente como ficou a superfície de decisão.
  - Ferramenta *read-only*.
  - Usada para diagnóstico.



# *Rule Viewer*



- Visualizador das Regras:
  - Permite visualizar graficamente como ficaram as regras.
  - Ferramenta *read-only*.
  - Usada para diagnóstico.

# Funcionamento das ferramentas



- Iniciadas se digitando "**fuzzy nome**" no prompt do matlab.
- Podem carregar e salvar informações no workspace ou no disco.
- Podem estar abertos simultaneamente.

---

# Exemplo: Controlador de Gorjeta

# Exemplo da Gorjeta

- Seguiremos os passos para construção de um sistema fuzzy:
  - Inicie o sistema digitando “**fuzzy**”
  - Controlador com 2 entradas e 1 saída:
  - Entradas?
    - Qualidade do Serviço
    - Qualidade da Comida
  - Saídas?
    - Gorjeta

# Entradas e saídas no Matlab



- Matlab inicia com 1 entrada e 1 saída:
  - Mudar no *Edit - Add Input*: aparecem 2 entradas
- Mude os nomes das variáveis de entrada e saída:
  - Clique na caixa uma vez (fica vermelha)
  - Mude o nome no campo de texto a esquerda.

# Definindo as funções de pertinência



- Clique: *Edit - add MFs*
  - Adiciona as funções de pertinência de cada variável
  - Define número de funções e tipo
- *Range*: muda os valores possíveis.

# Funções de pertinência



- Adicione as funções de pertinência dos 3 conjuntos:
- Serviço:
  - Range de 0 a 10, 3 funções e tipo *gaussmf*
- Comida:
  - Range de 0 a 10, 2 funções e tipo *trapmf*
- Gorjeta:
  - Range de 0 a 30, 3 funções e tipo *trimf*

# Nomes



- Acerte os nomes dos conjuntos:
  - Serviço:
    - pobre, bom e excelente
  - Comida:
    - Ruim e Deliciosa
  - Gorjeta:
    - pouca, média e generosa
- Mude os triângulos da gorjeta para não se sobreporem e fiquem só de 0 a 30.



# Regras



- Clique: *Edit - Rules*
- Adicione as 3 regras:
  - **Se** “o serviço é pobre” ou “a comida é ruim” **então** “dar pouca gorjeta”.
  - **Se** “o serviço é bom” **então** “dar gorjeta média”.
  - **Se** “o serviço é excelente” ou “a comida é ótima” **então** “dar muita gorjeta”.
- Ver menu de opções!

# Testando o sistema



- Clique: *View - View Rules*
  - Permite testar o sistema
  - Modifique as entradas e ele calcula automaticamente a saída
- Clique: *View - View Surface*
  - Permite verificar a superfície de decisão gerada.
  - Use o mouse para girar figura.

# Finalizando



- Salve para o disco, para poder ser usado novamente.
- Extensão: *.fis*

# Usando o controlador



- Para carregar um controlador pronto:
  - *a = readfis('gorjeta.fis')*
- Para mostrar as características de um controlador:
  - *showfis(a)*
- Para display:
  - *plotfis(a)*
  - *plotmf(a, 'input', 1)*

# Para usar o Controlador



- Parecido com uma rede neural, existe uma função para calcular a saída a partir de uma entrada:
  - *evalfis*(*[1 2]*, *a*)
- Retorna o resultado do controle para as entradas 1 e 2.

---

# Exercício 1: Controle da Velocidade de um Motor

# Controle de motor



- Um determinado motor precisa rotacionar sempre na velocidade correta (RPM).
  - O motor deve manter **7272** RPM
- Controlamos a velocidade do motor alterando a tensão de entrada.

# Controle de motor



- A entrada do problema é a velocidade do motor em RPM
- Em palavras, o motor pode estar:
  - muito lento
  - velocidade correta
  - muito rápido



# Controle de motor



- A saída do Fuzzy é a tensão (de 0 a 12V)
- Em palavras:
  - menos tensão
  - sem alteração
  - mais tensão

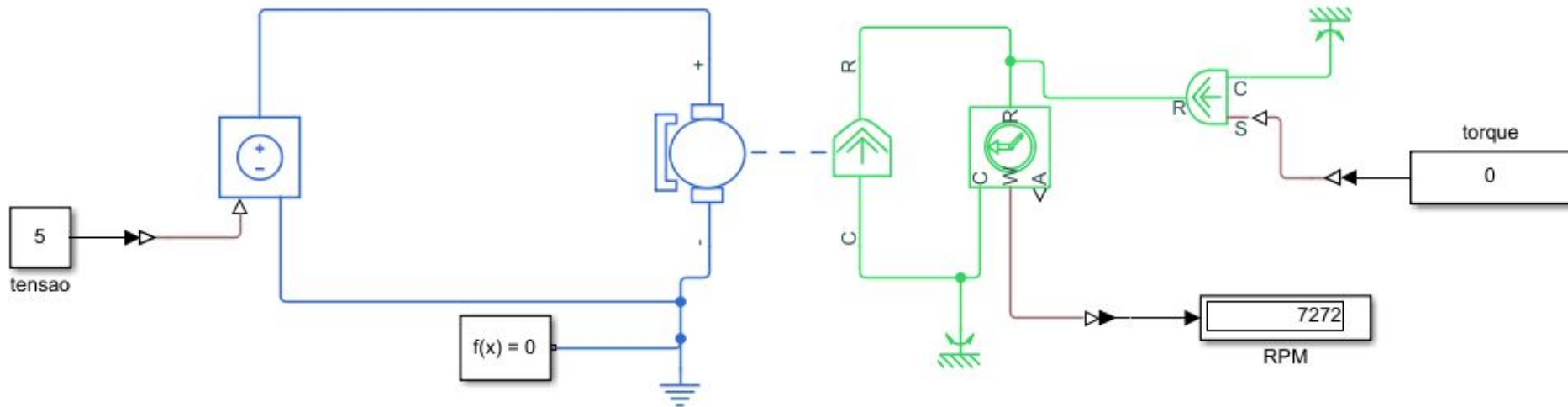
# Controle de motor



- As regras básicas:
  - Se o motor girar de forma muito lenta, então deve-se aumentar a tensão
  - Se o motor estiver girando na velocidade correta, nenhuma alteração deve ser feita
  - Se o motor estiver rápido demais, então é necessário diminuir a tensão

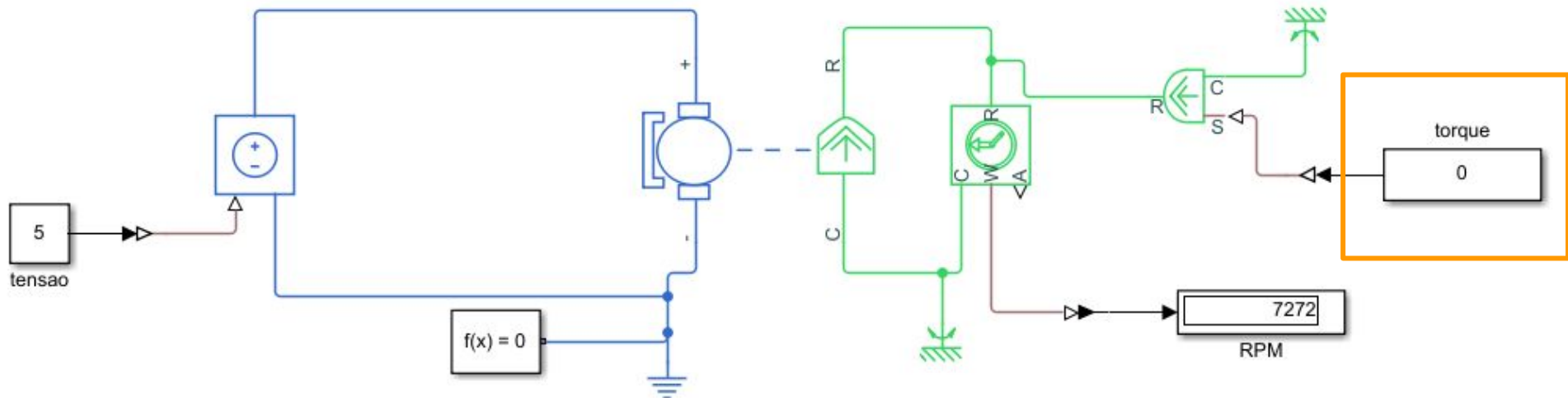
# Controle de motor

- No Simulink:



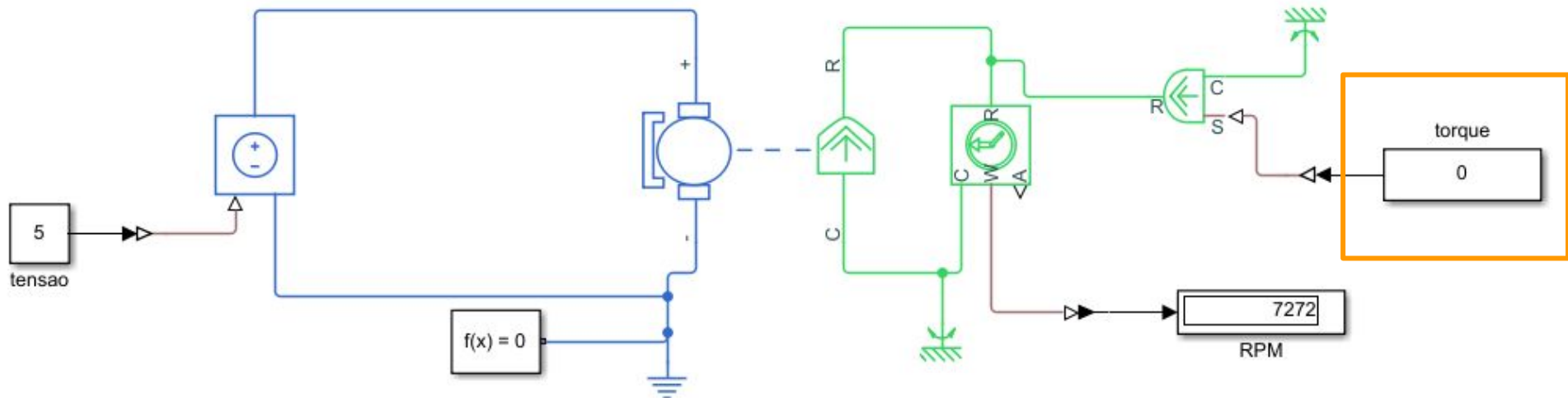
# Controle de motor

- No Simulink:



- 5V mantém o motor na velocidade correta!
- Desde que o motor esteja sem carga (torque 0)

- O valor do torque pode ser alterado de  $-5e-4$  (carga) até  $5e-4$  (rotaciona mais)



- No Simulink, com o controlador Fuzzy:

