

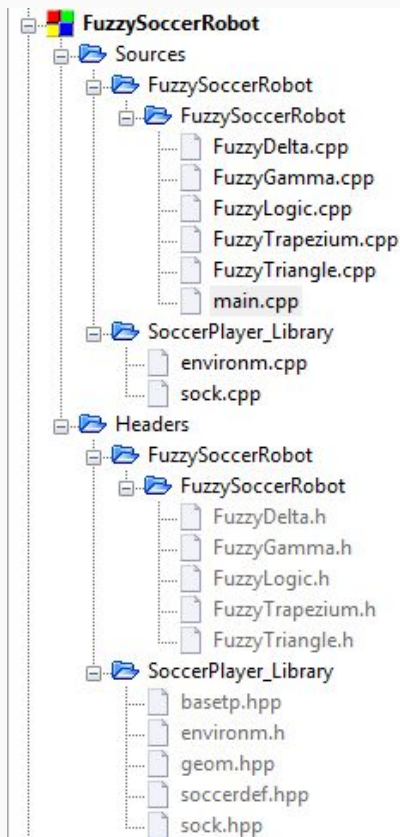
# Futebol de Robôs

Gabriel Osório Alves



Implementação em **C++** utilizando a **IDE Code::Blocks**, com a seguinte divisão de classes:

- **FuzzyTrapezium, FuzzyTriangle, FuzzyDelta e FuzzyGamma:**
  - Modelo matemático das funções fuzzy representadas pelo nome. Todas tem sua função membership e cálculos para encontrar o centro diferenciados.
- **FuzzyLogic:**
  - Cálculo das Regras de Ativação;
  - Método de Mamdani;
  - Cálculo da matriz fuzzy;
  - Defuzificação (Centróide).
- **Main:**
  - Inicia as regras;
  - Controla todas as jogadas e os cálculos;
  - Salva os logs para o [robô neural](#).

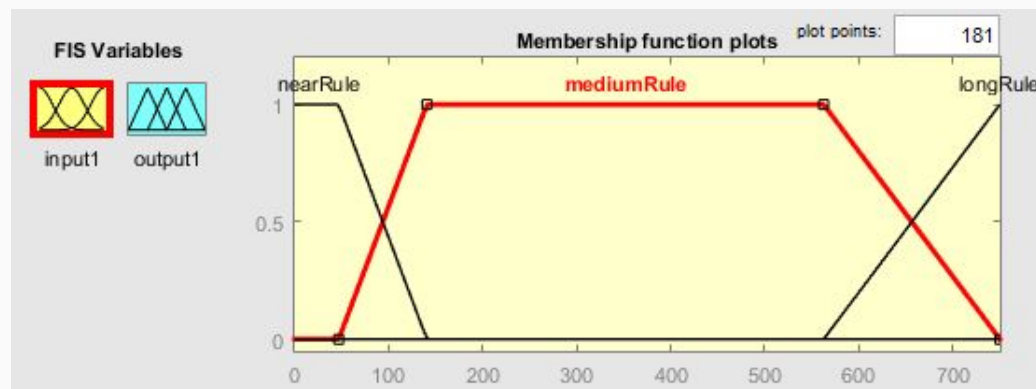
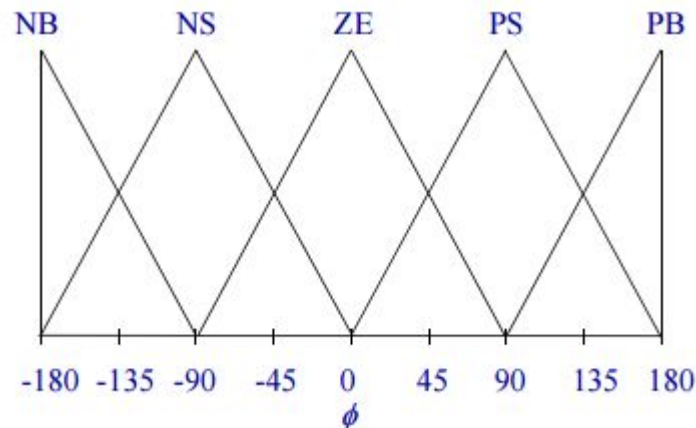


# Regras do Robô Fuzzy:

## Criação e iniciação das regras do Sistema Fuzzy:

```
// Cria as regras NB,NS,ZE,PS,PB
FuzzyLogic::NB = new FuzzyDelta(-M_PI,-M_PI/2);
FuzzyLogic::NS = new FuzzyTriangle(-M_PI, -M_PI/2, 0.0);
FuzzyLogic::ZE = new FuzzyTriangle(-M_PI/2, 0.0, M_PI/2);
FuzzyLogic::PS = new FuzzyTriangle(0.0, M_PI/2, M_PI);
FuzzyLogic::PB = new FuzzyGamma(M_PI/2, M_PI);

// Em adição cria o conjunto de regras para LONG, MEDIUM e NEAR
FuzzyGamma* longRule = new FuzzyGamma(560.0, 750.0);
FuzzyTrapezium* mediumRule = new FuzzyTrapezium(45.0, 140.0, 560.0, 750);
FuzzyDelta* nearRule = new FuzzyDelta(45.0, 140.0);
```



# Matrizes

- **Matriz Near:**

- Faz com que o robô vire na direção do alvo, assim “jogando” a bola na direção do gol (ou oposta a ele caso seja a goleira dele).

- **Matriz Medium:**

- Tentar sempre aproximar a bola pelos lados sempre que a bola esteja na frente do robô e o gol atrás ou dos lados.

- **Matriz Long:**

- Menor caminho possível até a bola.

Matriz Near					
	NB	NS	ZE	PS	PB
NB	NS	NS	PS	PS	NS
NS	NB	NS	NS	PS	NS
ZE	NB	NS	ZE	PS	PB
PS	PS	NS	PS	PS	PB
PB	PS	NS	NS	PS	PB
Matriz Medium					
	NB	NS	ZE	PS	PB
NB	NB	NS	PS	PS	NS
NS	NB	NS	PS	PS	NS
ZE	NB	NS	ZE	PS	PB
PS	PS	NS	NS	PS	PB
PB	PS	NS	NS	PS	PB
Matriz Long					
	NB	NS	ZE	PS	PB
NB	NB	NS	ZE	PS	PB
NS	NB	NS	ZE	PS	PB
ZE	NB	NS	ZE	PS	PB
PS	NB	NS	ZE	PS	PB
PB	NB	NS	ZE	PS	PB

## Inferência:

- Método do Centróide (Mamdani)
- Saída:
  - Ângulo para qual o robô deve virar convertido para a força utilizada nos motores do mesmo pela fórmula disponibilizada:

```
// Como a defuzificação devolve um valor em ângulos, preciso converter para os valores dos motores  
leftMotor  = cos( out ) - sin( out );  
rightMotor = cos( out ) + sin( out );
```

## Resultados:

- Robô funcional, sempre marcava gol quando jogava sozinho;
- Não foi implementada nenhuma tentativa de não fazer ele ficar preso nas paredes (por exemplo dar ré) porém isso só acontecia quando os 2 robôs brigavam por uma bola que estava em algum canto;
- Ganhava aproximadamente 65% dos jogos contra o robô fuzzy disponibilizado.

Como os resultados foram melhores, esse robô fuzzy foi utilizado para o desenvolvimento do robô neural.

### Métodos utilizados:

- Aprendizado utilizando os logs de jogo do robô fuzzy contra ele mesmo;
- Logs só eram salvos quando os mesmos causavam a uma situação de vitória (gol do robô);
- **Neurônios de Entrada:**
  - Inicialmente estava utilizando os 8 neurônios de entrada descritos pelo professor;
  - Problemas em possível tentativa de aprendizado de regras que não existiam (robô seguia a bola ou andava em zigue-zague / círculos);
  - Solução: remoção dos outros dados não utilizados.
- **Neurônios de Saída:**
  - 2 neurônios que representam as forças dos motores de saída.

A seleção de bons parâmetros se mostrou difícil pois boa parte do desenvolvimento foi feito levando em conta os dados que não eram utilizados pelo robô fuzzy (o que fazia com que o método nunca chegasse a bons resultados).

Após várias tentativas com tamanhos de entrada diferentes e com a topologia da camada oculta tendo entre 4 e 16 neurônios, variando o fator de aprendizado entre valores de 0,001 até 0,000001 e com número de épocas indo de 1000 até 50000 descobri que:

- Fator de aprendizado variar dependendo do tamanho da entrada: **boa convergência**;
- Número de épocas acima de 10.000 não influenciava muito
- Número de neurônios na camada oculta demonstrava maior mudança significativa no resultado final.



### **Primeiro robô neural com características funcionais do robô fuzzy:**

- Dados:
  - Tempo de jogo do robô fuzzy: 30 minutos (300.000~ linhas de registros)
  - Neurônios: 6
  - Fator de Aprendizado: 0.0001
  - Épocas: 5000
  - Melhor Época: 1990
  - Tempo de Aprendizado: 1343.04 segundos

Esse robô perseguia a bola, porém não tinha parecia ter aprendido as regras **mais específicas**. (esse robô ainda utilizava dados irrelevantes)

### **Segundo robô neural com características próximas as do robô fuzzy:**

- Dados:
  - Tempo de jogo do robô fuzzy: 10 minutos (140.000~ linhas de registros)
  - Neurônios: 8
  - Fator de Aprendizado: 1/140.000
  - Épocas: 5000
  - Melhor Época: 880
  - Tempo de Aprendizado: 656.87 segundos
  - MSE: 0.0008

Esse robô teve um tempo de aprendizado muito menor com um tamanho de entrada muito menor e mesmo assim obteve resultados melhores que o primeiro. Além do número de épocas, fator de aprendizado e neurônios na camada oculta também foram desconsiderados as outras saídas “inúteis” do robô fuzzy.

**Este robô tinha 3 neurônios na camada de entrada e 2 neurônios na camada de saída.**

### **Terceiro robô neural com características muito próximas as do robô fuzzy:**

- Dados:
  - Tempo de jogo do robô fuzzy: 60 minutos (550.000~ linhas de registros)
  - Neurônios: 10
  - Fator de Aprendizado:  $1/550.000$
  - Épocas: 10000
  - Melhor Época: 9960
  - Tempo de Aprendizado: 7139.04 segundos
  - MSE: 0.006

Este robô foi feito para testar uma entrada maior para a melhoria feita nos casos antigos. O resultado é muito parecido com o do robô fuzzy, apesar de parecer ser um pouco mais “lento” (o que pode ser por causa da procura de uma solução ótima ser muito exaustiva. Isso normalmente acontece quando existe um número muito baixo na camada oculta).

### **Quarto robô neural com características muito próximas as do robô fuzzy:**

- Dados:
  - Tempo de jogo do robô fuzzy: 60 minutos (550.000~ linhas de registros)
  - Neurônios: 12
  - Fator de Aprendizado:  $1/550.000$
  - Épocas: 10000
  - Melhor Época: 9750
  - Tempo de Aprendizado: 3975.24 segundos
  - MSE: 0.006

Este robô foi uma melhoria do terceiro robô, tendo em vista a “demora” para tomada de decisão. Obteve resultados quase iguais aos do robô fuzzy. Mesmo assim nos jogos dele contra o robô fuzzy, o robô fuzzy ganhava com em média 65% dos gols da partida.