Universidade Federal de Santa Catarina

Departamento de Informática e Estatística

INE5430 – Inteligência Artificial

Fizzy Line Tracker

Alunos: Renan Pinho Assi

Vinícius Schwinden Berkenbrock

14/06/19

# Introdução

1. A lógica Fuzzy, também conhecida como lógica difusa ou lógica nebulosa, é uma forma de lógica multivalorada na qual os valores lógicos podem ser qualquer número real entre 0 (FALSO) e 1 (VERDADEIRO). A lógica Fuzzy é utilizada quando, mesmo tendo todas as informações necessárias, não se pode simplesmente responder “sim” ou “não”, deve-se poder responder algo entre eles, como “talvez” ou “quase”.

Um conjunto Fuzzy é definido em um universo de discurso (conjunto base) X, e caracterizado pela sua função de pertinência:

A : X 🡪 [0, 1]

onde A(x) representa o grau com que x pertence a A.

Para o desenvolvimento deste trabalho foi disponibilizado um código fonte base no qual um caminhão deve estacionar na vaga disponibilizada, o caminhão possui sua localização e seu ângulo relativo à linha horizontal -X. Deve-se utilizar a lógica Fuzzy para, dado esses valores, tentar encontrar os ‘melhores’ valores para a manobra (quanto virar) do caminhão.

# Desenvolvimento

Como já mencionado, temos 3 entradas, posição relativa ao eixo X, posição relativa ao eixo Y e ângulo relativo ao eixo -X; e uma saída, o valor de quanto o caminhão deve virar.

## Fuzzificação

Na fuzzificação tivemos que definir a posição que o caminhão se encontra, pois sabemos que a vaga está próxima do meio.

FUZZIFY X

TERM LEFT := TRAPE 0 0.15 0.35 0.5;

TERM RIGHT := TRAPE 0.5 0.65 0.85 1;

TERM LEFT\_CENTER := (0.30, 0) (0.5, 1) (0.5, 0);

TERM RIGHT\_CENTER := (0.50, 0) (0.50, 1) (0.7, 0);

END\_FUZZIFY

Também tivemos que definir o ângulo atual em que o caminhão se encontra pra saber qual a potência da curva.

FUZZIFY ANGLE

TERM NORTH := TRIAN 225 270 315;

TERM WEST := (0,1) (15,1) (45,0) (300,0) (345,1) (360,1);//TRIAN 0 45 315;

TERM SOUTH := TRIAN 45 90 135;

TERM EAST := TRIAN 135 180 225;

END\_FUZZIFY

## Desfuzzificação

Na desfuzzificação definimos quais os valores das curvas a serem realizadas.

Temos Malmsteen como forte, Clapton como média e Chimbinha como fraca.

DEFUZZIFY STEER

TERM TURN\_RIGHT\_MALMSTEEN := TRIAN -1 -0.9 -0.5;

TERM TURN\_RIGHT\_CLAPTON := TRIAN -0.7 -0.3 -0.2;

TERM TURN\_RIGHT\_CHIMBINHA := TRIAN -0.2 -0.1 0;

TERM TURN\_LEFT\_CHIMBINHA := TRIAN 0 0.1 0.2;

TERM TURN\_LEFT\_CLAPTON := TRIAN 0.2 0.3 0.7;

TERM TURN\_LEFT\_MALMSTEEN := TRIAN 0.5 0.9 1;

METHOD : COA;

DEFAULT := 0;

END\_DEFUZZIFY

## Regras

RULEBLOCK

AND : MIN;

ACT : MIN;

ACCU : MAX;

RULE 1 : IF X IS LEFT AND ANGLE IS NORTH THEN STEER IS TURN\_RIGHT\_MALMSTEEN;

RULE 2 : IF X IS LEFT AND ANGLE IS SOUTH THEN STEER IS TURN\_LEFT\_CLAPTON;

RULE 3 : IF X IS LEFT AND ANGLE IS WEST THEN STEER IS TURN\_LEFT\_MALMSTEEN;

RULE 4 : IF X IS LEFT AND ANGLE IS EAST THEN STEER IS TURN\_RIGHT\_MALMSTEEN;

RULE 5 : IF X IS RIGHT AND ANGLE IS NORTH THEN STEER IS TURN\_LEFT\_MALMSTEEN;

RULE 6 : IF X IS RIGHT AND ANGLE IS SOUTH THEN STEER IS TURN\_RIGHT\_CLAPTON;

RULE 7 : IF X IS RIGHT AND ANGLE IS WEST THEN STEER IS TURN\_RIGHT\_MALMSTEEN;

RULE 8 : IF X IS RIGHT AND ANGLE IS EAST THEN STEER IS TURN\_LEFT\_MALMSTEEN;

RULE 9 : IF X IS LEFT\_CENTER AND ANGLE IS NORTH THEN STEER IS TURN\_RIGHT\_MALMSTEEN;

RULE 10 : IF X IS LEFT\_CENTER AND ANGLE IS SOUTH THEN STEER IS TURN\_RIGHT\_CHIMBINHA;

RULE 11 : IF X IS LEFT\_CENTER AND ANGLE IS WEST THEN STEER IS TURN\_RIGHT\_CHIMBINHA;

RULE 12 : IF X IS LEFT\_CENTER AND ANGLE IS EAST THEN STEER IS TURN\_LEFT\_CHIMBINHA;

RULE 13 : IF X IS RIGHT\_CENTER AND ANGLE IS NORTH THEN STEER IS TURN\_LEFT\_MALMSTEEN;

RULE 14 : IF X IS RIGHT\_CENTER AND ANGLE IS SOUTH THEN STEER IS TURN\_LEFT\_CHIMBINHA;

RULE 15 : IF X IS RIGHT\_CENTER AND ANGLE IS WEST THEN STEER IS TURN\_LEFT\_CHIMBINHA;

RULE 16 : IF X IS RIGHT\_CENTER AND ANGLE IS EAST THEN STEER IS TURN\_RIGHT\_CHIMBINHA;

END\_RULEBLOCK

## Algoritmo

No caso do algoritmo, enviamos para lógica fuzzy a posição X e o ângulo recebidos e alteramos o ângulo que será o próximo passo do caminhão com o ângulo recebido como retorno da função.

while ((fromServer = in.readLine()) != null) {

    StringTokenizer st = new StringTokenizer(fromServer);

    x = Double.valueOf(st.nextToken()).doubleValue();

    y = Double.valueOf(st.nextToken()).doubleValue();

    angle = Double.valueOf(st.nextToken()).doubleValue();

    System.out.println("x: " + x + " y: " + y + " angle: " + angle);

    /////////////////////////////////////////////////////////////////////////////////////

    fis.setVariable("X", x);

            fis.setVariable("ANGLE", angle);

            fis.evaluate();

            double offset = fis.getVariable("STEER").defuzzify();

            System.out.println("STEER value: " + offset);

            out.println(offset);

            out.println("r");

}

# 

# Dificuldades Encontradas

Tivemos dificuldade para “tunar” o valor das curvas, de modo que ele alivie o valor quando chega próximo do centro.

# Referências

[https://pt.wikipedia.org/wiki/L%C3%B3gica\_difusa](https://pt.wikipedia.org/wiki/Lógica_difusa)

<http://www.logicafuzzy.com.br/wp-content/uploads/2013/04/uma_introducao_a_logica_fuzzy.pdf>