INE5430 - Inteligência Artificial

Trabalho T4 - Sistemas Fuzzy

Caique Rodrigues Marques
c.r.marques@grad.ufsc.br

Fernando Jorge Mota
contato@fjorgemota.com

Introdução

A base para a execução deste trabalho está no uso de conceitos do sistema *fuzzy* ou sistema nebuloso, composto de lógica *fuzzy* e de conjunto *fuzzy*. O estudo de conjuntos *fuzzy* foi introduzido por Lofti Zadeh, em 1965, e é uma extensão da teoria de conjuntos clássica. A lógica *fuzzy* é uma lógica multivalorada onde a informação, ao contrário da lógica booleana, não é atribuída a um ou zero, mas podendo estar entre esses dois valores.

O cerne do raciocínio *fuzzy* é da transformação de informações linguísticas (podendo ser dúbias e vagas) em valores numéricos para que possam ser trabalhadas matematicamente e computacionalmente, depois, é realizado o inverso para que as soluções sejam avaliadas.

O sistema

O sistema utilizado consiste em fazer um caminhão estacionar na vaga especificada, em marcha ré, usando uma série de regras *fuzzy*. As variáveis que devem ser consideradas são: a posição x (y é considerado como o topo) do caminhão; a direção no qual ele está e o ângulo em que o volante está virado.

Variáveis do sistema

As variáveis entrada do sistema são valores x, apontando a posição inicial do caminhão (y é sempre considerado o topo), e a direção, que consiste onde a frente do caminhão está apontado (este dado é obtido a partir da conversão do angulo atual no qual o caminhão se encontra). A partir desses dados, um ângulo é gerado como saída, mostrando qual o angulo do volante do caminhão deve ser modificado antes de executar o próximo movimento a ré, num intervalo que vai de -30° até 30°. Outro ponto a notar é a delimitação das variáveis: o eixos x está limitado ao intervalo $(0, 1)$; as direções envolvem cima, baixo, esquerda e direita e são definidas a partir de seus valores respectivos em uma escala de 0° a 360°; Já o ângulo é delimitado de -30° até 30°, embora seja enviado para o servidor como um intervalo que vai de -1 a 1. Como esses intervalos fazem parte do funcionamento do sistema, vamos especificar e listar as condições que classificam um determinado valor fornecido pelo servidor dado em um termo que pode ser usado numa regra de forma mais simples. Segue a lista:

- **X**
  - `tooLeft` - Intervalo entre 0 e 0.3;
  - `left` - Intervalo entre 0.1 e 0.5;
  - `half` - Intervalo entre 0.4 e 0.6;
  - `right` - Intervalo entre 0.5 e 0.9;
  - `tooRight` - Intervalo entre 0.7 e 1.
- Direção (`direction`, no código)
  - `left` - Intervalo entre 90 e 270;
  - `right` - Intervalo entre 0 e 90 e intervalo entre 270 e 360;
  - `up` - Intervalo entre 0 e 180;
  - `down` - Intervalo entre 180 e 360.

Além dos termos acima, também definimos um termo que é usado como retorno para cada regra e que respeita intervalos similares ao do termo X:

- Ângulo (`angle`, no código)
  - `tooLeft` - Intervalo entre -1 e -0.4;
  - `left` - Intervalo entre -0.8 e 0;
  - `center` - Intervalo entre -0.3 a 0.3;
  - `right` - Intervalo entre 0 e 0.8;
  - `tooRight` - Intervalo entre 0.4 a 1.

Regras

A partir das definições especificadas na seção anterior, uma série de 40 regras são definidas para determinar qual caminho o caminhão deve percorrer. Todas as regras são condicionais (`IF, THEN`), sendo que dados os x e a direção, então o ângulo do volante deve ser uma das cinco posições especificadas na listagem anterior e são aplicados cálculos de lógica *fuzzy* para encontrar o valor que será retornado. Alguns exemplos dessas regras são encontrados abaixo:

x	direction	angle
tooLeft	left	tooLeft
tooLeft	right	tooRight
...	...	...
tooRight	down	tooRight

Note que os exemplos acima são apenas uma pequena amostra das 20 regras definidas, que não serão todas apresentadas aqui devido à sua extensão. Todas as regras podem ser vistas no arquivo `remoteDriver.fcl`.

Método de defuzzificação

Defuzzificação transforma os valores de entrada numa variável de saída que é interpretada pelo programa para qual a angulação do volante que deve estar, partindo do ponto e da direção iniciais. Essa transformação é através de alguma transformação numérica, o método aqui utilizado é o mais tradicional de defuzzificação: o cálculo do centroide (ou baricentro). O método envolve o cálculo do ponto central da função *fuzzy* de saída baseando-se nas operações feitas sobre os parâmetros de entrada, tal ponto indica a angulação do volante que o caminhão irá fazer.

Problemas encontrados

Dependendo dos parâmetros x e do ângulo com o qual o programa é iniciado, é possível encontrar pontos em que o programa simplesmente não consegue responder com as direções para que o caminhão estacione corretamente. Em alguns pontos, o caminhão consegue estacionar, outras vezes, ele acaba indo mais para a esquerda ou mais para a direita, errando o alvo, mas ainda assim chegando bem próximo ao esperado.

Como são poucos casos em que isso ocorre, acreditamos que se trata apenas de uma questão simples de probabilidade proporcionado pela lógica *fuzzy* e, portanto, acreditamos que é um problema aceitável dado a natureza probabilística deste trabalho. As condições em que o caminhão estaciona erroneamente acontece com mais frequência nas regiões de baixo, visto que o caminhão não consegue espaço o suficiente para conseguir virar em direção ao estacionamento.