



ESTUDO E APLICAÇÃO DA LÓGICA FUZZY PARA ESTACIONAMENTO AUTOMÁTICO DE VEÍCULOS

Wallace Soares Moreno
wallace.sm@hotmail.com
AEDB

Maíra Ribas Monteiro
ribasmaira@hotmail.com
AEDB e UNIFEI

Juliana Ribas Monteiro
ribasjuliana@hotmail.com
IFRJ e UNIFEI

Resumo: O crescimento do interesse de diversos órgãos da sociedade no ramo da inteligência artificial (IA) é um fato consumado desde o sucesso de grandes empresas de tecnologia com o emprego de técnicas deste tipo. Por isto, é especialmente interessante que os cursos tecnológicos e em especial os de engenharia e computação tenham grandes enfoques neste viés científico, considerado como o futuro de muitas áreas de trabalho hoje realizada por mão de obra humana. Este artigo tem como objetivo o enfoque no funcionamento da lógica fuzzy, em especial, dentre as várias soluções existentes em inteligência artificial; compreensão de suas premissas e expor uma aplicação, estacionamento automático de veículos, onde uma modelagem e controle por esta técnica mostra-se interessante; além de explorar as possibilidades da técnica através de simulações computacionais.

Palavras Chave: IA - Lógica Fuzzy - Modelagem - Simulações -



1. INTRODUÇÃO

O primeiro a propor o modelo matemático que é a base dos conjuntos nebulosos foi o matemático e engenheiro Lotfali Askar-Zadeh nascido no Azerbaijão e naturalizado norte-americano, lecionou por anos na Universidade de Berkeley na Califórnia. É considerado o pai da teoria da possibilidade. Em 1965 publicou um estudo que definia um sistema lógico no qual os valores limites do sistema são 0 e 1, mas diferentemente da álgebra booleana, não se limita a apenas estes dois valores, mas a qualquer número real neste intervalo (ZADEH, 1965). Assim, pode-se definir qualquer função real para uma aplicação *fuzzy* desde que seus valores máximo e mínimo sejam, respectivamente 1 e 0. Mais recentemente, *softwares* de manipulação e trabalho com *fuzzy* permitem a definição de valores diferentes para os limites, conforme o necessário; porém o trabalho com o intervalo entre 0 e 1 ainda é o mais usual.

Todavia, foi apenas a partir das primeiras sugestões de aplicações práticas desta teoria para problemas industriais, através de algoritmos que implementavam uma expressividade de linguagem que tinha como objetivo a política de busca; de controle; de investigação dos dados e suas correlações; e as aplicações que isto implicava; elaboradas e aprofundadas pelo professor (e engenheiro) da Universidade Queen Mary em Londres, Ebrahim H. Mamdani em 1974 que as pesquisas em lógica *fuzzy* se popularizaram (MAMDANI, 1974). Ele foi o primeiro a montar um modelo baseado na lógica *fuzzy* e logo se tornou referência absoluta na área, junto a Zadeh, com quem chegou a trabalhar e juntos participaram de muitas conferências disseminando as possibilidades que a lógica *fuzzy* trazia. Um de seus alunos, Michio Sugeno, com quem trabalhou por pouco mais de uma década, também veio a se tornar referência no assunto (TAKAGI e SUGENO, 1985). Hoje, existem milhares de artigos escritos com análises de problemas sob a ótica da lógica *fuzzy*. A Teoria dos Conjuntos Nebulosos é usualmente classificada como uma expansão da teoria dos conjuntos e a forma como os problemas são abordados e vistos nesta ótica, como uma extensão da lógica clássica; passando assim a ser conhecida como lógica *fuzzy*.

Atualmente, uma das principais ferramentas para se trabalhar com lógica *fuzzy* é o *software* MATLAB. Desenvolvido pela empresa americana MathWorks, especializada em desenvolver soluções computacionais e matemáticas, o MATLAB é utilizado em conjunto com o Simulink, outro *software* da empresa, para realizar os mais variados tipos de trabalhos na área de matemática e engenharia. Desde a plotagem de gráficos até o desenvolvimento de sistemas de redes neurais artificiais, o MATLAB possui vastas aplicações. Tem sido muito empreendido pela facilidade que traz no âmbito educacional e pela forma simples de criar sistemas que também sejam aplicáveis em plantas industriais; sua programação se dá na linguagem própria denominada “.m” pela empresa responsável pelo *software*, mas que é próxima o suficiente da linguagem aberta de programação “C” para que leigos possam entender os princípios fundamentais de seus comandos. Uma de suas principais vantagens é a de ser capaz de simular com precisão programas de controle e comando, por este motivo sendo de grande valia na área de testes. Portanto, o MATLAB oferece uma interface específica, especializada para a construção de muitos modelos, como controladores PIDs, analisadores de sinais e de imagens, entre outros; tornando-o o *software* perfeito para a aplicação dos conhecimentos adquiridos e simulações computacionais pretendidas.

Este artigo tem como objetivo apresentar um estudo sobre inteligência artificial, abordando a lógica *fuzzy* e suas aplicações. O trabalho está dividido da seguinte forma: a Seção 2 apresenta um breve histórico sobre a evolução da inteligência artificial, na Seção 3 é explicada a lógica *fuzzy*, posteriormente na Seção 4 é abordada a metodologia de modelagem por lógica *fuzzy*, depois uma aplicação com lógica *fuzzy* é apresentada na Seção 5 e finalmente a Seção 6 apresenta as conclusões.



2. INTELIGÊNCIA ARTIFICIAL

Na década 1990 o conceito de Inteligência Artificial começou a crescer e foi muito discutido e pesquisado pela área acadêmica, devido às inovações tecnológicas das décadas anteriores, que finalmente permitiam a união de *software* e *hardware* para grande potencial de processamento de dados. Embora tenha sido comprovado em 1971 por Stephen Cook com a Teoria da Complexidade Computacional que a solução de problemas computacionais não depende de *hardwares* mais potentes necessariamente; a criação dos microprocessadores e a disponibilização comercial de acesso a computadores para a sociedade, tal como seu contínuo melhoramento possibilitaram a criação de *hardwares* capazes de executar *softwares* cada vez mais exigentes. Finalmente as ideias de grandes pensadores sobre o assunto como Alan Turing, Joseph Weizenbaum, Marvin Minsky, e tantos outros que escreveram o seu nome na história da computação e programação tinham agora os meios para serem colocadas em prática.

Embora este conceito não fosse, à época, novo ou revolucionário, uma vez que o cunho do termo “Inteligência Artificial” e a as principais características para um sistema ser reconhecido como tal já tivessem sido determinadas em 1956 em uma conferência na Universidade de New Hampshire – e eventualmente atualizados com o tempo, o avanço da tecnologia e a visão de outras mentes brilhantes sobre o assunto – o interesse e trabalhos sobre o tema cresceram exponencialmente nesta década em particular. O próprio Marvin Minsky, que havia há tantos anos estado na importante conferência sobre o assunto, reviu seus conceitos e em 1985 explicou que todo o conceito de “inteligência” não é tão mecanicista como se imaginava e não pode ser tão determinístico e rígido como um tratado matemático. Obviamente, este pensamento foi influenciado pelos estudos de psicólogos e psiquiatras sobre a própria mente humana, padrão comparativo para sistemas de inteligência artificial.

Entretanto, o processo de convencimento de investidores e indústrias sobre o tema não se dava como o esperado. Fora da área dos *experts* em tecnologia e programação, o assunto era visto com muito ceticismo. Era extremamente difícil conseguir financiamento para aprofundar pesquisas na área e os estudiosos se desdobravam com os recursos que conseguiam. Hoje se sabe que uma das primeiras instituições a perceber o potencial da inteligência artificial e investir nisto foi o Governo Americano, que através da *National Security Agency* – NSA, silenciosamente contratou alguns especialistas que julgava mais promissores no ramo e começou a trabalhar em aplicações para a área militar e de vigilância de ameaças – dentro e fora de seu território. Mas, àquela época, os pesquisadores da área que não estavam envolvidos nisto – os que estavam não poderiam, por conseguinte, se pronunciar sobre – se viam sem suporte de qualquer instituição, restringindo bastante o círculo dos que continuaram a debater e desenvolver pesquisas na área – as poucas que aconteciam eram financiadas, geralmente, por universidades ou pelos próprios pesquisadores. Havia jornais exclusivos sobre o assunto, que inclusive circulavam desde os anos 1980, mas apenas dentro da comunidade acadêmica; nunca extrapolando as suposições teóricas da resolução de problemas propostos para a real implantação de um modelo de testes num protótipo, justamente por falta de recursos.

O setor industrial não acreditava que a premissa desta tecnologia poderia lhe ser útil, uma vez que modelagens matemáticas e controladores simples solucionavam todos os problemas encontrados à época, mesmo que em certas vezes com alguma dificuldade ou não de forma satisfatória. E quando não era possível, ou viável do ponto de vista econômico, a aplicação de um operador humano supria a relação custo-benefício do trabalho.

Apenas na virada do milênio este ceticismo começou a mudar. Uma nova empresa de tecnologia apostou na internet e na necessidade – e dificuldade – das pessoas para encontrar conteúdos na rede mundial de computadores. Dois estudantes de doutorado de Stanford, Larry



Page e Sergey Brin, fundaram, em 1998, a Google. Uma empresa com a premissa de encontrar conteúdo na internet para você. Já existiam empresas e soluções similares, mas a forma como a Google programou seu sistema com uma implementação de Inteligência Artificial, a fim de personalizar e otimizar os resultados encontrados de forma específica para cada usuário; seus algoritmos de procura únicos logo a tornaram não apenas a líder no seu nicho de mercado, mas também impulsionou seu valor de mercado (antes mesmo de se tornar pública e disponibilizar suas ações para venda na bolsa de valores em 2004); permitindo que a empresa diversificasse seus negócios e ampliasse seus investimentos nos sistemas já em operação. Hoje, em seus muitos braços na indústria digital, a Google é capaz de captar dados de bilhões de usuários e empresas e alcançar seu objetivo de encontrar resultados rápidos e personalizados, graças à grande capacidade computacional que o seu crescimento financeiro pôde bancar. Com todo este poderio em mãos, há grandes debates sobre até onde uma empresa pode ou deve captar dados de seus clientes para uso próprio.

A exemplo do recente vazamento de que o Facebook, outro gigante da internet criado pela geração *millennial*, disponibilizou dados privados de aproximadamente 50 milhões dos usuários desta rede social para a análise de outra empresa, o que cria além de um paradoxo ético com as diretrizes da própria empresa e os termos de privacidade e de uso aceitos pelo usuário no momento de adesão para criar uma conta, o problema se estendeu por bater de frente com leis americanas a respeito do acontecido: a empresa terceirizada possui sede em outro país e especula-se que os dados obtidos possam ter sido utilizados para até mesmo vir a interferir nas eleições presidenciais americanas de 2016; criando grande alvoroço midiático, abrindo investigações e culminando na convocação do CEO da empresa para uma prestação de contas diante do Congresso Americano. A responsabilidade ética, moral e legal de ser portador de dados da vida de uma pessoa é muito grande e está no centro dos debates pelo mundo no momento. Enquanto isso, estes dados são a maior fonte de plataformas de inteligência artificial para trabalhar. O próprio Facebook vem utilizando soluções do tipo para encontrar modos de otimizar processos de vendas e exprimir *advertisements* personalizados com base nos dados de gostos e desgostos captados a partir do dia-a-dia que os usuários compartilham na rede social.

Foi o sucesso da Google que abriu os olhos do mundo empresarial para as possibilidades que a inteligência artificial poderia trazer para um mundo globalizado e conectado. Atualmente, é trivial que empresas principalmente digitais utilizem soluções de inteligência artificial nos mais variados níveis para otimizar seus negócios. A própria criação do Facebook, redes sociais como se conhece hoje, grandes *sites* de compras, de notícias, em geral, todas as grandes empresas da internet utilizam aplicações desta tecnologia. E seu uso não se limita apenas aí. indústrias de produção em série variadas têm investido nestes tipos de solução para, principalmente, sistemas supervisórios de plantas industriais. Ou outros sistemas de controle variados. Em resumo, a Inteligência Artificial foi muito debatida na década de 1990 pela área acadêmica e hoje a Indústria começa a comprar a ideia e aplicá-la em seus processos.

Dentro do tópico “Inteligência Artificial”, existem várias vertentes de pesquisa que se mostram muito promissoras e inclusive já vêm sendo empreendidas nas mais variadas aplicações como bancos, indústria automobilística, empresas de dados e sistemas, etc. Algumas destas vertentes mais conhecidas são: lógica *fuzzy*, *rough sets*, redes neurais artificiais, algoritmos genéticos e sistemas especialistas; sendo que estas abordagens decaem sobre quatro grupos de forma a abordar inteligência artificial, postulados por Minsky em 1956: conexionista, evolucionária, conjuntos difusos ou aproximados e simbólica. É comum observar situações nas quais duas ou mais destas abordagens são aplicadas a fim de resolver o problema proposto, neste caso fala-se em uma modelagem híbrida do sistema de Inteligência Artificial, é muito usual encontrar este tipo de configuração.



3. LÓGICA FUZZY

A lógica *fuzzy* aprofunda-se no conhecimento humano, como uma simples linguagem de programação. Ela reflete a forma como o ser humano encara o processo de resolução de problemas, de pensamento, de comunicação e de reprodução destas ações. Ao buscar as raízes da forma de encarar os problemas de computação como é enfrentado, decai-se sobre a clássica lógica bivalente de Aristóteles, estabelecida há mais de 2 mil anos. Encara-se um problema, uma variável, uma solução, uma situação, uma condição; como apenas duas possibilidades, conforme bem reformulou Shakespeare em sua famosa peça Hamlet: “ser ou não ser, eis a questão”. Considera-se que para determinado parâmetro ou condição de análise, a resposta pode ser apenas “verdadeiro” ou “falso”. A bola está dentro da caixa, ou não está. O gato de Schrödinger está vivo ou morto, aos olhos de um observador. Não há espaço para talvez, para “meias-verdades”. Porém, no dia-a-dia, encontra-se muitas situações que são impossíveis descrever de forma binária ou que se comportem linearmente. Neste cenário, a lógica *fuzzy* mostra-se uma alternativa promissora e simples.

A lógica *fuzzy* traz uma forma de encarar os problemas na qual não há verdade ou mentira, mas sim uma infinidade de tons de verdade entre os níveis estabelecidos como verdade e mentira.

Observa-se que enquanto na matemática clássica um número absoluto no conjunto dos complexos ocupa um ponto no espaço (ou na reta); um número representado sob a ótica *fuzzy* equivale a uma área em volta daquele número na reta (podendo esta área ser em vários formatos, o mais usual de representar é em forma triangular). Quanto mais abrangente, mais aberta, a área de um número *fuzzy*, mais nebulosa (mais *fuzzy*) considera-se esta área pois a chance do valor dentro dela estar próximo do “verdadeiro” idealizado é menor. Assim, a definição da área de um número *fuzzy* (ou rótulo) é dependente apenas do nível de nebulosidade que o programador deseja acrescentar a ela. Números ou rótulos muito nebulosos, obviamente, tornam o processamento de dados muito mais complicado.

Como pode ser visto na Figura 1, há interpretações diferentes para a definição de número nas lógicas clássicas e na lógica *fuzzy*. Para a lógica clássica, um valor é equivalente a um ponto no espaço, numa reta, ou até mesmo uma reta constante caso esteja-se considerando-a em função do tempo. Pode-se visualizar este conceito em (a) na Figura 1. Em contrapartida, os números *fuzzy* apresentam os conceitos de áreas de pertencimento. Um valor pode ser igual ou pertencer em algum grau ao estipulado ou desejado se estiver dentro daquela área. Por isto utiliza-se também a ferramenta de função de pertinência: em um conjunto nebuloso, o quão pertencente àquele conjunto é um valor? Através da aplicação desta, calcula-se o pertencimento do valor ao rótulo *fuzzy*. Devido a suas peculiaridades, um número *fuzzy* pode conter área maior ou menor; obviamente. Em (b) na Figura 1, tem-se um número *fuzzy* com área de abrangência relativamente pequena, e isto diminui as possibilidades de classificação de valores; enquanto que permite uma maior precisão na identificação de valores que se encontram dentro da área de pertinência. Já em (c), com uma área de abrangência maior, há também uma incerteza maior de que o valor realmente pertencerá ao rótulo estabelecido em algum nível.

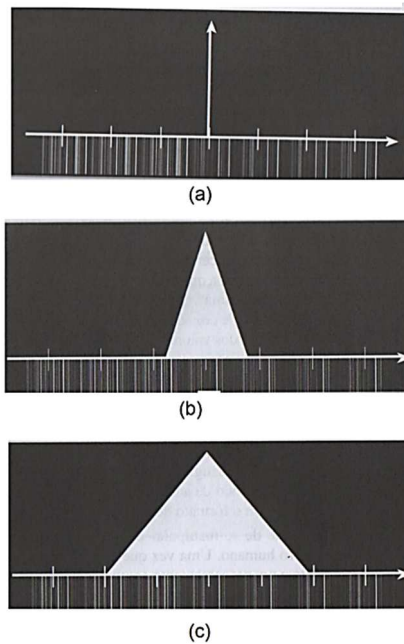


Figura 1: Transcrição visual do comparativo entre o conceito de um número real e um número *fuzzy*.
Fonte: Simões e Shaw (2007).

A lógica *fuzzy*, como qualquer lógica ou linguagem de programação, oferece um conjunto de regras e valores que são os guias para o processador de dados, sobre a forma como estes dados serão manipulados a fim de se obter a saída desejada. A forma mais simples de efetuar uma linha de raciocínio deste tipo é a de causa e consequência. Se determinada circunstância acontece, o sistema toma determinada ação correspondente. Este tipo de condicional está presente em todo tipo de linguagem de programação, de funções no excel a linguagem C. Este tipo de raciocínio acontece a todo tempo mesmo que de forma inconsciente em nosso cérebro. Chama-se implicação lógica, a relação de causa e efeito medida desta forma. Em um programa, ela deve ser feita através de programação e atrelamento entre variáveis e valores de referência. É muito comum em aplicação de regras *fuzzy*.

Todo o processo de inteligência em que se baseia a lógica *fuzzy* tem como objetivo emular a forma humana de pensar. O ser humano é capaz de aproximar dados e informações sensoriais, através de simbolismos vagos ou imprecisos. A lógica *fuzzy* tem, como premissa, realizar o mesmo procedimento: apresentar uma resposta satisfatória mesmo que não haja dados suficientes sobre aquela determinada situação em acontecimentos anteriores. A capacidade de interpolar e extrapolar dados é uma das mais importantes em um sistema de inteligência artificial, pois conclusões são possíveis de serem tomadas, para o ser humano, mesmo que não haja dados suficientes. As informações recebidas pelo cérebro são segmentadas de forma análoga a conjuntos ou números *fuzzy*. Por exemplo, uma pessoa, ao observar o nível de água em um reservatório é capaz de exprimir se este nível se encontra baixo, médio ou alto. Mesmo sem saber a capacidade do reservatório, a taxa de entrada ou de saída de um fluido; etc. Estes rótulos – baixo, médio e alto – são valores fuzzificados, números ou intervalos numéricos transformados em conceitos linguísticos que aproximem as informações do sistema de uma linguagem natural e ao mesmo tempo ofereça uma maneira do computador ser capaz de entender e processar.

Tendo isto em vista, estabelece-se o padrão de que os sinais com os quais a lógica *fuzzy* lida estão sempre no intervalo $[0,1]$ – ou em raríssimas ocasiões, $[-1,1]$. Todos os graus de



verdade (ou mentira) estão dentro deste intervalo, sendo a máxima 0 totalmente falso e 1 totalmente verdadeiro.

A fim de operar análises e empregar a lógica *fuzzy* a problemas, é necessário fazer algumas considerações matemáticas específicas devido a sua condição diferenciada.

Primeiramente, conjuntos ou números *fuzzy*, por necessidade, devem seguir um certo padrão de confiabilidade ao longo de sua área de abrangência. Isto é possível graças à aplicação da ferramenta matemática chamada convexidade. Esta ferramenta estabelece uma fórmula para descrição da área *fuzzy* de interesse de forma que não ocorram “buracos” ou “baías” nestas áreas, o que acarretaria na perda de confiabilidade da análise *fuzzy* por permitir a existência de pontos cegos em sua área de pertinência. Visualmente, isto se transmite a funções *fuzzy* de controle com aspectos geométricos específicos: triângulos, trapézios, funções exponenciais e logarítmicas ou até mesmo a função normal se necessário – as opções são várias, desde que atendam ao princípio da convexidade.

Como números *fuzzy* podem definir um conjunto próprio, é necessário também estabelecer uma forma como valores dentro deste conjunto podem se relacionar. Através de regras de operação conhecidas pela geometria analítica como definições de um espaço vetorial, estas regras são a t-norma e a s-norma. As t-normas incluem: interseção, produto algébrico, produto logarítmico, produto inverso, produto limitado e produto drástico. As s-normas são: união, soma algébrica, soma limitada, soma logarítmica, soma disjunta e soma drástica. Como pode-se perceber, a forma como as t-normas e s-normas são aplicadas a conjuntos *fuzzy* permitem que todas as operações conhecidas sejam realizadas nestes conjuntos, desde que obedecidas suas propriedades – comutativas, associativas, monotônicas e condições de contorno (LIMA et al, 2014).

Os operadores mínimo e produto são utilizados para compor as regras.

$$\text{Interseção: } x \text{ t } y = \text{mín}(x, y);$$

$$\text{União: } x \text{ s } y = \text{máx}(x, y). \quad (1)$$

4. MODELAGEM FUZZY

Conforme todos os esforços da engenharia atual, o objetivo de todos estes estudos e descobertas na área de lógica e controle não são feitas para se manterem no campo teórico. O interesse por trás do desenvolvimento destas teorias é aplicação em sistemas de controle e automação, simplificação e naturalização das interfaces homem-máquina; entre tantas outras possibilidades.

Um modelo *fuzzy* basicamente segue esses passos para processamento de informação: fuzzifica a(s) variável(is) de entrada, faz a checagem da base de conhecimento (podendo haver aprendizado de máquina no processo ou não) de forma a extrair os dados e a lógica de tomada de decisão necessária e defuzzifica a resposta para a saída. Este processo tem como objetivo transformar a variável de entrada real em número *fuzzy*, que então passa pelo processo de inferências *fuzzy* e gera uma tomada de decisão e, ao final, transforma a resposta obtida em *fuzzy* para uma linguagem real, ou eletrônica; se for este o caso.

O mais interessante do processo do modelo *fuzzy* é a sua capacidade de emulação do comportamento humano sob a mesma situação. Tanto que, ao adequar um processo para ser mantido por um controlador *fuzzy* em malha fechada, aconselha-se que seja entrevistado um operador humano especialista naquele processo, pois a partir das informações obtidas, será



possível construir um cenário de todas as situações possíveis em termos linguísticos, capazes de serem compreendidos pelo modelo *fuzzy*, que passará então a cumprir o papel do operador com precisão; inclusive em situações inesperadas, graças ao seu poder de interpolação e extrapolação a partir de dados existentes – novamente, o aprendizado de máquina (WANG e MENDEL, 1992).

Todas as informações sobre o processo, sobre as variáveis, sobre a forma de agir e sobre os comandos de processamento residem na base de conhecimento. É através dela que a variável de entrada é convertida para valores *fuzzy* no intervalo [0,1], associada a funções de pertinência linguísticas e então efetuando a estratégia de controle determinada.

O processo de defuzzificação da variável pode ser um pouco complexo, caso seja necessário que a informação de saída seja utilizada como um valor em um processo ou controle subsequente. Quando a saída desejada é qualitativa, a defuzzificação é desnecessária.

Antes de falar de processamento *fuzzy*, é necessário trazer à luz, conforme citado anteriormente, as funções de pertinência *fuzzy*. Uma função de pertinência tem o objetivo de avaliar o quanto um valor pertence a ela, dentro da escala de trabalho *fuzzy* [0,1]. A partir desta informação, segue-se o curso de tomada de ação. Funções de pertinência são gráficas e tabuladas, transcritas sobre a reta de universo de discurso da variável analisada. Várias funções de pertinência podem ser usadas no mesmo universo de discurso, inclusive sobrepunhando parcialmente uma à outra. Isto, de fato, acontece muito corriqueiramente e é uma das formas mais simples de estabelecer o controle *fuzzy* (SUGENO E KANG, 1988). Várias condições linguísticas de avaliação para a variável, no caso, e seu ponto no universo de discurso ocupará um lugar provável entre duas funções de pertinência. Um cálculo interno é realizado pelo modelo para determinar então se o valor daquela variável naquele momento é mais pertinente a qual função, trazendo consigo a condição linguística daquela variável naquele momento, e ao passar pelas regras no próximo passo permite uma tomada de decisão (CHEN e KUO, 1995).

Os modos mais comuns de funções de pertinência vistos são triangular, trapezoidal e *fuzzy*-tons, como ilustrados pelas Figuras 2 e 3; não impedindo contanto a implementação de outras funções de área conforme a necessidade.

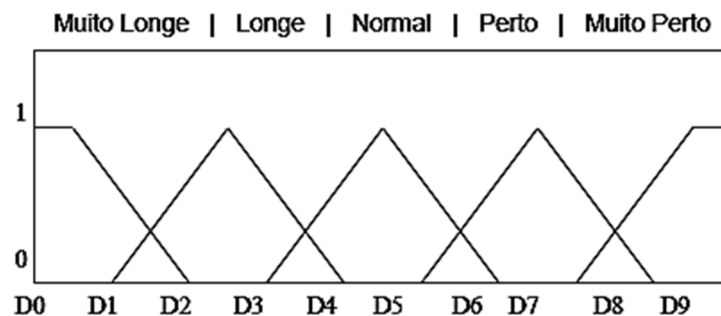


Figura 2: Exemplo de função de pertinência triangular.

Fonte: <http://www3.eletronica.org/tag/fuzzy> (2018).

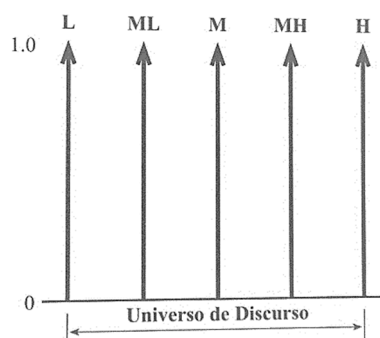


Figura 3: Exemplo de funções de pertinência *fuzzy-tons*.

Fonte: Simões e Shaw (2007).

Observa-se que a Figura 2 mescla funções de pertinência trapezoidais em seus rótulos-limite com funções de pertinência triangulares igualmente espaçadas para os outros rótulos. Supondo-se um valor qualquer “d” dentro do universo de discurso genérico descrito na Figura 2, pode-se estabelecer sua pertinência a qualquer um dos 5 rótulos. Tomando-se como exemplo o rótulo normal, temos que este valor qualquer, dentro do universo de discurso tem uma pertinência equivalente ao rótulo normal de acordo com a seguinte análise:

$$Normal = \begin{cases} 0, & \text{se } d < D_3; \\ \frac{d-D_3}{D_5-D_3}, & \text{se } d \in [D_3, D_5]; \\ 1, & \text{se } d = D_5; \\ \frac{D_7-d}{D_7-D_5}, & \text{se } d \in [D_5, D_7]; \\ 0, & \text{se } d > D_7. \end{cases} \quad (2)$$

Esta análise pode ser repetida para todos os rótulos, havendo ligeira adaptação dos parâmetros quando a função se torna trapezoidal ou assume outra forma. Pelo motivo da interseção das funções de pertinência em vários pontos, é que se diz que o conjunto é nebuloso, pois o valor “d” pode estar num ponto do universo de discurso que seja abrangido por mais de uma função. Para isto, é feito o cálculo a exemplo de (2) para determinar qual é o rótulo mais relevante para aquele valor naquele momento e, mais à frente, qual tipo de medida tomar e com qual intensidade.

Num universo de discurso, a quantidade e o formato das funções de pertinência é determinada pela entrevista com o operador humano especializado, as informações que se possui sobre o processo, sua natureza. Esta tarefa de dimensionamento geralmente se dá pela técnica de tentativa e erro. Felizmente muitos *softwares* atuais permitem a simulação da situação, então é uma questão de tempo para ajustar os parâmetros até obter-se o formato efetivo para a situação. Porém, geralmente trabalha-se com uma quantidade de 2 a 7 funções de pertinência num universo de discurso. Ao executar experiências repetitivas aumentando-se a quantidade de funções de pertinência a partir de 7, não é perceptível na saída melhorias abrangentes o suficiente para serem significativas (SIMÕES e SHAW, 2007).

Observando as Figuras 2 e 3, cada função de pertinência é equivalente a uma condição linguística, conforme discutido anteriormente.

Dentre a linha de processamento de dados por um modelo *fuzzy*, um dos passos se destaca como leque de possibilidades para a situação considerada. Os métodos de



defuzzificação e a escolha deles não pode ser aleatória, pois afeta diretamente a saída obtida e, por conseguinte, os passos seguintes da malha de controle. São eles: defuzzificação por centro-de-área, centro-do-máximo e média-do-máximo.

Lembrando que todo o processamento feito é com base em variáveis linguísticas e funções de pertinência, com determinados graus de pertinência para as variáveis defuzzificadas. Isso cria áreas como resposta (os graus de verdade ditos anteriormente) que devem ser traduzidos para valores reais novamente.

- Centro-de-área (C-o-A)

Calcula o centroide da área composta que representa o termo de saída *fuzzy*. Este termo de saída é a união das contribuições de todas as regras de controle. Assim, divide a área obtida como resposta em partes iguais e obtém o valor médio como saída. Obtém o valor equivalente mais mediano possível, mas que muitas vezes pode não estar de acordo com os graus de pertinência. Este método de defuzzificação possui alguns problemas inerentes: é impossível de ser usado quando as funções de pertinência não se sobrepõem; regras diferentes que geram a mesma saída geram áreas que não são contabilizadas por este método e a necessidade de integração para cálculo de centroide demanda uma capacidade computacional mais alta. É calculado através de (3):

$$v_o = \frac{\sum_{n=1}^N v_n \mu_{r_i}(v_n)}{\sum_{n=1}^N \mu_{r_i}(v_n)} \quad (3)$$

- Centro-do-máximo (C-o-M):

Este método considera apenas os picos das funções de pertinência finais – ou seja, considera a todas como *fuzzy-tons* – e ignora as áreas dessas funções. Regras que contribuem de forma sobreposta ou com várias respostas são consideradas por este método. Assumindo os *fuzzy-tons* como pesos, a defuzzificação de centro do máximo encontra o valor de saída discreto através do ponto de equilíbrio entre estes pesos; ou seja, uma média ponderada dos máximos. Este método permite uma melhor análise de dados paralelos. No caso mais de uma regra acionar a mesma condição como saída, esta condição fica reforçada. É calculado através de (4):

$$v_o = \text{máx} [\mu_{r_i}(v)] \quad (4)$$

- Média-do-máximo (M-o-M):

Parecida com a defuzzificação C-o-M, estabelece uma média aritmética simples entre todos os máximos envolvidos; evitando a possível situação de se ter que escolher entre quais máximos utilizar. Esta abordagem é considerada a solução mais plausível por desconsiderar o formato das variáveis *fuzzy* na saída. É calculado através de (5):

$$v_o = \frac{1}{N} \sum_{n=1}^N \text{máx} [\mu_{r_i}(v_n)] \quad (5)$$

Os métodos de defuzzificação também estão atrelados a uma questão de continuidade: o quanto uma mudança infinitamente pequena na entrada causa de mudança abrupta na saída.



Os métodos C-o-M e C-o-A são contínuos, enquanto que M-o-M é descontínuo. C-o-M e C-o-A oferecem um melhor compromisso entre as variáveis de entrada e saída, ou seja, uma pequena variação não é significativa desde que o compromisso da unidade de grandeza seja mantido. M-o-M, porém, oferece a saída mais plausível e isto quer dizer que esta saída não é única. A variação abrupta pode ocorrer.

Ao falar em modelos *fuzzy*, pode-se citar três tipos principais: baseados em regras, paramétricos, baseados em equações relacionais. Todos eles atuam da mesma forma, mas encaram sua base de conhecimento com abordagens diferentes.

Modelos *fuzzy* baseados em regras têm princípio de funcionamento bem simples: estabelecem-se regras que, quando seus requisitos são satisfeitos, ativam uma saída por consequência. É pura relação condicional. Devido a sua simplicidade, é o mais utilizado e inclusive pode ser embutido em PIDs por exemplo.

Modelos *fuzzy* paramétricos, no entanto, combinam a descrição provida pelas regras com uma situação de aproximação linear local, através de um sistema de regressão linear que represente a relação de entrada e saída localmente. Ou seja, combina o método *fuzzy* com descrições matemáticas dos processos, uma abordagem clássica. Esta relação de regressão linear oferece, então, na saída uma condição de pesos e média ponderada para a saída *fuzzy*.

Por último, existem os modelos *fuzzy* que trabalham sobre equações relacionais. Durante a abordagem inicial do processo para a modelagem através da lógica *fuzzy*, é muito comum que seja feita uma entrevista com um operador especializado para se conhecer exatamente como aquele processo se comporta em termos linguísticos. Assim, é feita a identificação do sistema através da identificação da estrutura e dos parâmetros. O tipo de estrutura escolhido é baseado em modelos a partir do conhecimento que se tem do processo enquanto que a obtenção dos parâmetros inclui a formulação de regras de controle e funções de pertinência. O objetivo da modelagem *fuzzy* é sempre obter um modelo dinâmico que tenha um comportamento de entrada e saída o mais aproximado possível do sistema que está sendo modelado. Porém, há também uma forma automática de obter-se os parâmetros do sistema, através das equações relacionais, que fazem estimativas e simulações entre a entrada e saída com base na estrutura fornecida.

Em vez de fornecer as regras de controle e as funções de pertinência, fornece-se a este modelo a entrada (ou entradas) do sistema e informa-se a saída esperada. O sistema então passa a estimar sozinho as melhores formas de obter as respostas desejadas com as variáveis fornecidas. Este método aplica aprendizado de máquina de forma inerente em seu funcionamento, pois está o tempo todo se adaptando a novas situações com base nas amostragens que realiza. O sistema é muito parecido com o de aprendizagem das redes neurais.

5. APLICAÇÃO DA LÓGICA FUZZY

Devido as suas funcionalidades, o MATLAB foi escolhido como plataforma para conduzir as simulações necessárias ao estudo do tópico.

Dentre os muitos meios de simulação de várias interfaces e situações da engenharia elétrica, eletrônica e de controle e automação; existe a Anfis: *adaptive network-based fuzzy inference system*. Constitui um sistema de inferência *fuzzy* cujos parâmetros são ajustados por colaboração de uma rede neural adaptativa (JANG, 1993). Seu objetivo é permitir a construção de um conjunto de regras difusas e de funções de pertinência necessários a um modelo *fuzzy*. Como no modelo *fuzzy* baseado em equações relacionais, os parâmetros das funções de pertinência e a estrutura das regras *fuzzy* são obtidos através de um processo de aprendizado de máquina supervisionado, no qual são feitas simulações de entradas hipotéticas e, conhecendo-se as saídas hipotéticas, o sistema se ajusta a elas. Aplicações conjuntas de lógica *fuzzy* e redes



neurais artificiais são muito comuns, pois a última facilita muito a implementação e programação da primeira. Sistemas híbridos destas duas técnicas são tão corriqueiros que atualmente possuem sua própria denominação: *neurofuzzy* (SILVA et al, 2010).

Portanto, o MATLAB oferece uma interface específica, especializada para a construção de muitos modelos, como PIDs, analisadores de sinais e de imagens, entre outros. Entre eles está a FIS – Sistema de inferência *fuzzy*. No MATLAB ela é chamada de toolbox *fuzzy* e é ativada ao digitar-se na área de trabalho o comando *fuzzy* e apertar-se <ENTER>. Esta *toolbox* abre uma janela na tela que permite ao usuário ter acesso aos parâmetros e configurações de um sistema *fuzzy*. A partir de então a edição da estrutura e dos parâmetros é livre, permitindo ao usuário realizar as modelagens que desejar.

Foi, então, reproduzida uma aplicação, conforme explicitado a seguir. O problema abordado é um clássico sistema posicionador de veículos (WANG, 1994). Nele, um caminhão está posicionado em um campo de área definida pelas cotas x e y . O caminhão, em qualquer posição que estiver, terá suas cotas (x,y) de posição no plano e os ângulos θ e ϕ referentes respectivamente ao ângulo entre o caminhão e o eixo das cotas y e x . Estas são, então, as variáveis do problema.

No MATLAB, pode-se procurar as soluções de duas formas: escrevendo o algoritmo do processo de controle *fuzzy* ou então inicializando uma FIS, que permita processar os dados.

Aplicando o problema de controle de estacionamento automático de veículos, conhecido como *truck backer-upper control* (WANG,1994), tem-se a área do estacionamento definida pelas cotas y (com determinado tamanho) e x $[0, 20]$. Deseja-se estacionar o veículo em posição perpendicular ($\phi=90^\circ$) na cota $x=10$. A posição inicial (x,y) é qualquer e com um dado ângulo entre seu eixo longitudinal e o eixo da cota x . O ângulo θ é a inclinação das rodas do veículo em relação ao seu eixo longitudinal que é ação de manobra sobre ele com o objetivo de comandar o estacionamento na posição desejada – ilustrado na Figura 4.

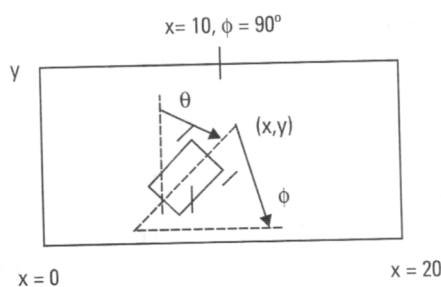


Figura 4: Ilustração da aplicação.

Fonte: Lima et al (2014).

A dinâmica, não-linear, do posicionamento é representada pelas equações (LIMA et al, 2014), onde b é o tamanho do veículo:

$$x(k + 1) = x(k) + \cos[\varphi(k) + \theta(k)] + \sin\varphi(k)\sin\theta(k);$$

$$y(k + 1) = y(k) + \sin[\varphi(k) + \theta(k)] - \cos\varphi(k)\sin\theta(k);$$

$$\varphi(k + 1) = \varphi(k) - \sin\varphi \left[\frac{2\sin\theta(k)}{b} \right]^{-1} \quad (6)$$



A Tabela 1 possui uma amostragem de dados sistema (LIMA et al, 2014).

Utilizando o método proposto por Wang e Mendel (1992) foram obtidas as regras básicas de controle (Tabela 2). As funções de pertinência correspondes aos conjuntos nebulosos são ilustradas pela Figura 5. As etiquetas b1, b2, b3, ce, s1, s2, s3 indicam os conjuntos nebulosos, os índices “-“ e “+” indicam os valores extremos dos parâmetros das funções (LIMA et al, 2014). Os valores modais estão na Tabela 3.

Tabela 1: Amostragem do sistema.

x	φ	θ
1,00	0,00	-19,00
1,95	9,37	-17,95
2,88	18,23	-16,90
3,79	26,59	-15,85
4,65	34,44	-14,80
5,45	41,78	-13,75
6,18	48,60	-12,70
7,48	54,91	-11,65
7,99	60,71	-10,60
8,72	65,99	-9,55
9,01	70,75	-8,50
9,28	74,98	-7,45
9,46	78,70	-6,40
9,59	81,90	-5,34
9,72	84,57	-4,30
9,81	86,72	-3,25
9,88	88,34	-2,20
9,91	89,44	0,00

Fonte: Lima et al (2014).

Tabela 2: Regras *fuzzy* do controle.

θ	x					
		s2	s1	ce	b1	b2
φ	s3	1s2	6s3	-	-	-
	s2	2s2	7s3	12s3	17s3	-
	s1	3b1	8s1	13s2	18s3	23s2
	ce	4b2	9b2	14ce	19s2	24s2
	b1	5b2	10b3	15b2	20b1	25s1
	b2	-	11b3	16b3	21b3	26b2
	b3	-	-	-	22b3	27b2

Fonte: Lima et al (2014).

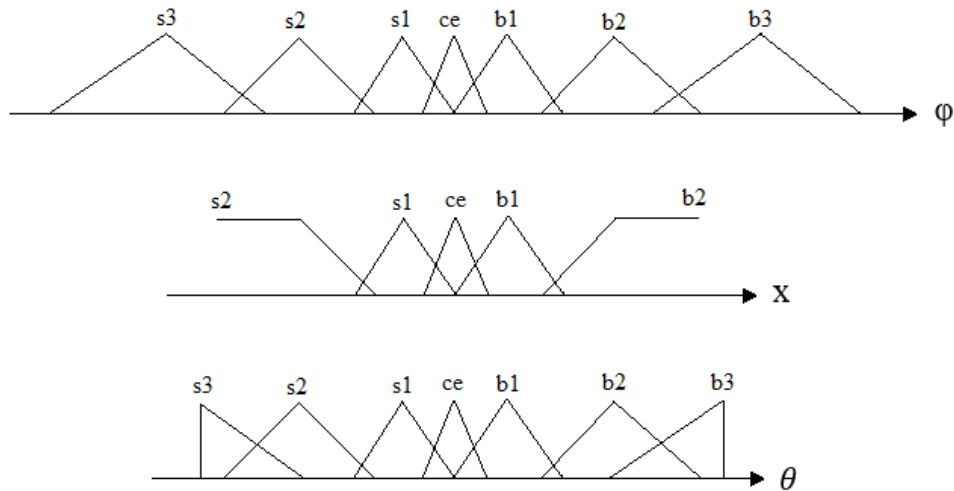


Figura 5: Funções de pertinência.
Fonte: Lima et al (2014).

Tabela 3: Valores modais das funções de pertinência.

	φ^-	φ^+	x^-	x^+	θ^-	θ^+
s3	-115	-15			-40	-20
s2	-45	45	1,5	7	-33	-7
s1	15	90	4	10	-14	0
ce	80	100	9	11	-4	4
b1	90	165	10	16	0	14
b2	135	225	13	18,5	7	33
b3	195	295			20	40

Fonte: Lima et al (2014).

Foram utilizadas regras do tipo Mamdani e funções de pertinência triangulares (LIMA et al, 2014), abaixo tem-se algumas como exemplo:

$$\begin{aligned}
 r_1: & \text{IF } x = s2 \text{ AND } \varphi = s3 \text{ THEN } \theta = s2; \\
 r_6: & \text{IF } x = s1 \text{ AND } \varphi = s3 \text{ THEN } \theta = s3; \\
 r_{27}: & \text{IF } x = b2 \text{ AND } \varphi = b3 \text{ THEN } \theta = b2
 \end{aligned}
 \tag{7}$$

Na fuzzificação a composição foi feita pelo operador produto. A defuzzificação é efetuada pelo método de centro-de-área. Foram necessárias 35 regras para o veículo partir de qualquer posição e estacionar no ponto desejado (LIMA et al, 2014).

Para a simulação computacional, estipulou-se a posição inicial do caminhão em $x = 5$ m, $y = 5$ m e $\varphi = 60^\circ$. Deseja-se estacionar o caminhão na cota $x = 10$ com ângulo $\varphi = 90^\circ$. A partir da atuação inicial, o veículo varia sua posição de forma a se aproximar cada vez mais do *setpoint* estabelecido. Conforme pode-se observar na Figura 6, o sistema de controle é extremamente eficiente, sendo capaz de posicionar o veículo na cota e ângulo desejados.

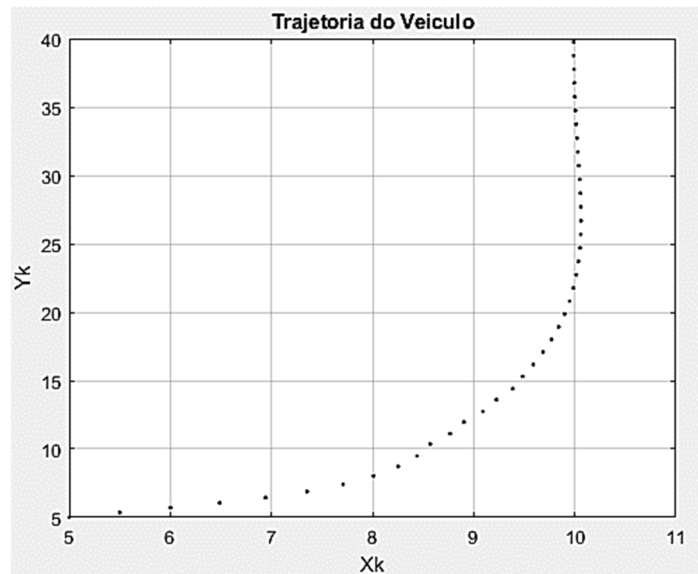


Figura 6: Gráfico de resposta da trajetória do veículo simulada de acordo com os parâmetros estabelecidos.

6. CONCLUSÃO

Como apresentado a inteligência artificial está cada vez mais sendo aplicada e ganhando confiança da área industrial. Modelos baseados em inteligência artificial conseguem obter melhores resultados em sistemas não-lineares, e/ou variantes no tempo e/ou com múltiplas entradas e saídas quando comparados com modelos matemáticos. Em sistemas mais complexos a inteligência artificial consegue modelar de forma mais fácil quando comparada aos modelos matemáticos. Ela está sendo uma alternativa às abordagens tradicionais.

Os modelos *fuzzy* mostram uma boa precisão na modelagem funções não-lineares por causa da sobreposição de conjuntos difusos associados. Os controladores baseados em regras (semelhantes aos controladores *fuzzy*) têm interpretações mais robustas no que diz respeito às incertezas do sistema, e têm uma boa capacidade de mapear características não-lineares de operação. Os controladores *fuzzy* utilizam procedimentos de fuzzificação e defuzzificação de dados que exigem cálculo das regras associadas.

A aplicação *truck backer-upper control* apresentada mostrou o bom desempenho das regras *fuzzy*, foi possível a partir de qualquer posição inicial, estacionar o veículo na posição desejada.

7. AGRADECIMENTOS

Os autores agradecem pela bolsa de Iniciação Científica do Centro de Pesquisa, Pós-Graduação e Extensão (CPGE) mantido pela Associação Educacional Dom Bosco (AEDB).

8. REFERÊNCIAS

- CHEN, C.L.; KUO, F.C.** Design and analysis of a fuzzy logic controller. *Int. J. System Sci.* v. 26, p. 1.223-48, 1995.
- JANG, J.** Anfis: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics*, v. 23 n.3, p. 665-685, 1993.
- LIMA, I.; PINHEIRO, C. A. M.; SANTOS, F. A. O.** *Inteligência Artificial*. Rio de Janeiro: Elsevier, 2014.



- MAMDANI, E. H.** Applications of fuzzy algorithms for control of simple dynamic plant. Proc IEE, v 121(12), p. 1.585-1.588, 1974.
- SILVA et al.** Redes Neurais Artificiais Para Engenharia e Ciências Aplicadas. São Paulo: Artliber, 2010.
- SIMÕES, M. G.; SHAW, I. S.** Controle e modelagem fuzzy. 2ª. ed. São Paulo: Blucher e FAPESP, 2007.
- SUGENO, M.; KANG G. T.** Structure Identification of Fuzzy Model. Fuzzy Sets and Systems, v. 28 n.1, p. 15-33, 1988.
- TAKAGI, T.; SUGENO, M.** Fuzzy Identification of Systems and its Applications to Modeling and Control. IEEE Transactions on Systems, Man and Cybernetics, v. 15 n.1, p. 116-132, 1985.
- WANG, L.X.** Adaptive Fuzzy Systems and Control. New Jersey: Prentice Hall, 1994.
- WANG, L.X.; MENDEL, J.M.** Generating fuzzy rules from examples. IEEE Transactions on Systems, Man and Cybernetics, v. 22(6), p. 1414-1427, 1992.
- ZADEH, L. A.** Fuzzy sets. Information and Control. n. 8, p.338-53, 1965.