

---

**Lógica Fuzzy Aplicada no Controle das  
Polias de uma Transmissão Continuamente  
Variável**

*Evandro Gonçalves Barroso Medina*

---

PROJETO DE CONCLUSÃO DE CURSO

Data de Depósito: 20/11/2022

Assinatura: \_\_\_\_\_

# **Lógica Fuzzy Aplicada no Controle das Polias de uma Transmissão Continuamente Variável**

*Evandro Gonçalves Barroso Medina*

*Valter Ribeiro Lima Júnior*

Monografia apresentada ao Instituto Superior de Educação do UNIFOR/MG, como requisito parcial para obtenção do título de bacharel em Ciência da Computação, sob a orientação do Prof<sup>o</sup>. Me. Valter Ribeiro Lima Júnior

UNIFOR-MG – Formiga  
20 de novembro de 2022

*Dedico este trabalho aos  
meus pais e a minha esposa,  
todos fonte de amor, auste-  
ridade e incentivo.*

# AGRADECIMENTOS

---

Ao Prof. Me. Valter Ribeiro Lima Júnior pela companhia, ensinamentos, orientação e oportunidade de desenvolver este projeto.

A todos os professores do Centro Universitário de Formiga (UNIFOR-MG) que direta ou indiretamente contribuíram com o conhecimento nas diversas disciplinas do curso de Ciência da Computação, alicerce para o desenvolvimento deste trabalho.

*“Há uma força motriz mais poderosa que o vapor, a eletricidade e a energia atômica: a vontade”*

*Albert Einstein*

# Sumário

---

---

<b>Lista de Figuras</b>	<b>ii</b>
<b>Lista de Tabelas</b>	<b>iii</b>
<b>1 INTRODUÇÃO</b>	<b>1</b>
1.1 OBJETIVOS . . . . .	2
1.1.1 Objetivo Geral . . . . .	2
1.1.2 Objetivos Específicos . . . . .	2
1.2 JUSTIFICATIVA . . . . .	2
1.3 ESTRUTURA DA MONOGRAFIA . . . . .	3
<b>2 REFERENCIAL TEÓRICO</b>	<b>4</b>
2.1 O VEÍCULO E A TRANSMISSÃO . . . . .	4
2.1.1 Transmissão Automática . . . . .	4
2.1.2 Diferença entre Transmissão Automática e Manual . . . . .	4
2.1.3 Diferença entre Transmissão Automática e Transmissão CVT . . . . .	5
2.1.4 Transmissão CVT . . . . .	5
2.1.5 Tipos de CVT . . . . .	6
2.2 A Lógica Fuzzy . . . . .	6
2.3 Controladores Programáveis . . . . .	7
2.3.1 Controlador Fuzzy X Controlador PID . . . . .	8
2.3.2 Linguagem de Controle Fuzzy (FCL) . . . . .	9
2.3.3 jFuzzyLogic . . . . .	10
2.3.4 Sistema de Inferência Fuzzy (FIS) . . . . .	10
<b>3 ESTADO DA ARTE</b>	<b>11</b>
3.1 Proposta de Controlador <i>Fuzzy</i> para Transmissão Continuamente Variável . . .	11
3.2 Controlador <i>Fuzzy</i> Aplicado a Veículos Elétricos com Transmissão Variável Con- tínua . . . . .	12

3.3	Método e Sistema Utilizando Lógica <i>Fuzzy</i> para o Controle de uma Transmissão CVT . . . . .	12
<b>4</b>	<b>METODOLOGIA</b>	<b>14</b>
4.1	Análise dos Métodos de Controle . . . . .	14
4.2	Modelagem do Sistema . . . . .	15
4.2.1	Linguagem de Programação Utilizada . . . . .	18
4.2.2	Estratégia de Controle . . . . .	18
4.2.3	Parâmetros de Entrada para o Controlador <i>Fuzzy</i> . . . . .	18
4.2.4	Método de Inferência . . . . .	19
4.3	Simulador . . . . .	22
4.3.1	Implementação do simulador . . . . .	23
4.3.2	Arquivo FCL . . . . .	24
4.3.3	Processo de Fuzzificação . . . . .	28
4.3.4	Processo de Inferência . . . . .	28
4.3.5	Processo de Defuzzificação . . . . .	30
<b>5</b>	<b>Testes e Resultados</b>	<b>31</b>
5.1	Avaliação dos Resultados . . . . .	31
<b>6</b>	<b>Conclusões</b>	<b>39</b>
6.1	Considerações Finais . . . . .	39
6.2	Contribuições . . . . .	39
6.3	Trabalhos Futuros . . . . .	40

# Lista de Figuras

---

---

2.1	Diagrama de uma CVT . . . . .	5
2.2	Estrutura e elementos funcionais do controle <i>fuzzy</i> . . . . .	7
2.3	Estrutura genérica da FCL . . . . .	9
4.1	Diagrama de Blocos . . . . .	17
4.2	Diagrama de blocos da estratégia de controle . . . . .	18
4.3	Exemplo em que o centro da área não é aplicável . . . . .	21
4.4	Relação entre os resultados para o método COA e COG . . . . .	21
4.5	Exemplos de funções de pertinência triangular e trapezoidal . . . . .	22
4.6	Exemplo da interface de declaração do Bloco de Função em linguagem estruturada de texto e em diagrama . . . . .	24
4.7	Gráfico função de pertinência da variável de entrada rotação do motor . . . . .	26
4.8	Gráfico função de pertinência da variável de entrada abertura da borboleta . . . . .	26
4.9	Gráfico função de pertinência da variável de saída controle da polia primária . . . . .	27
4.10	Demonstra os valores <i>singletons</i> para a função de pertinência da variável de saída . . . . .	30
5.1	Resultado para a saída no trajeto 3 . . . . .	38



# Lista de Tabelas

---

5.1	Resultados das saídas para o controle em cada trajeto . . . . .	38
-----	---	----

# RESUMO

---

---

MEDINA, E. G. B. *Lógica Fuzzy Aplicada no Controle das Polias de uma Transmissão Continuamente Variável*. Monografia (Graduação) — Centro Universitário de Formiga – UNIFOR-MG – Formiga, 2022.

O mundo atual passa por diversas dificuldades ambientais. Aquecimento global, alterações climáticas e poluição são exemplos de adversidades enfrentadas. Um dos causadores dos problemas descritos é a emissão de gases poluentes por parte dos veículos que queimam combustíveis fósseis. Em países em desenvolvimento, como o Brasil, torna-se inviável à maior parte da população adquirir um veículo elétrico ou híbrido, seja por motivo de valor elevado ou de poucos postos elétricos de abastecimento. Diante desses fatos, surge a oportunidade de realizar estudos direcionados à diminuição da emissão de poluentes por parte de automóveis a combustão. Este trabalho tem o intuito de demonstrar benefícios da utilização da lógica *fuzzy* aplicada no controle das polias de uma transmissão continuamente variável (CVT). O projeto explora a categoria de veículos com câmbio automático do tipo CVT, atualmente em ascensão no país. Foi desenvolvido um algoritmo que implementa a técnica da inteligência artificial (IA), chamada lógica *fuzzy*. Esse algoritmo é executado por um simulador que gera resultados de saída para controlar a transmissão com o objetivo de otimizar a transferência de energia do motor até as rodas. Como consequência, pode-se obter melhora no consumo de combustível. Diferentemente das propostas similares encontradas na literatura, este projeto apresentou os benefícios em se utilizar apenas um processo de controle *fuzzy*. Desta forma, elimina-se a utilização de cálculos complexos utilizados em sistemas de controle tradicionais. Após a realização de testes com o simulador, foi possível observar o desempenho do sistema de controle. Foi também factível manipular dados e prever resultados de inferência esperados, que abrangem as relações entre as variáveis de entrada e de saída. Toda a estrutura e características do processo, bem como os testes e resultados obtidos a partir das simulações em distintas situações de trajetos do veículo, são apresentadas neste trabalho.

Palavras-chave: Lógica *fuzzy*; Transmissão CVT; Inteligência artificial; Controlador *fuzzy*.

# ABSTRACT

---

---

MEDINA, E. G. B. *Fuzzy Logic Applied to Control the Pulleys of a Continuously Variable Transmission*. Article (Graduation) — Centro Universitário de Formiga – UNIFOR-MG – Formiga-MG, 2022.

The world is experiencing nowadays several environmental difficulties. Global warming, climate change and pollution are examples of faced adversities. One of the causes of the problems described is the emission of polluting gases by vehicles that burn fossil fuels. In developing countries such as Brazil, it becomes unfeasible for most of the population to purchase an electric or hybrid vehicle, either because of the high price or few electric vehicle charging stations. In view of these facts, the opportunity arises to carry out studies aimed at reducing the emission of pollutants by combustion vehicles. This work aims to demonstrate the benefits of using fuzzy logic applied to control the pulleys of a continuously variable transmission (CVT). This project explores the category of vehicles with automatic transmission of the CVT type, currently with ascension in the country. An algorithm was developed that implements the technique of artificial intelligence (AI), called fuzzy logic. This algorithm is executed by a simulator that generates output results to control the transmission in order to optimize the transfer of energy from the engine to the wheels. As a consequence of this, an improvement in fuel consumption can be achieved. Unlike similar proposals found in the literature, this project presented the benefits of using only one fuzzy control process. In this way, the use of complex calculations used, until then, in traditional control systems is eliminated. After carrying out tests with the simulator, it was possible to observe the performance of the control system. It was also feasible to manipulate data and predict expected inference results, which encompass the relationships between input and output variables. All the structure and characteristics of the process, as well as the tests and results obtained from the simulations in different situations of vehicle paths, are presented in this work.

Keywords: Fuzzy logic; CVT transmission; Artificial intelligence; Fuzzy controller.

---

# INTRODUÇÃO

---

No Brasil, o consumidor tem preferido, no momento da compra, veículos com transmissão automática em detrimento da transmissão manual. Em 2020, quase a metade dos carros emplacados é equipada com o câmbio automático, e este tipo de transmissão está amplamente difundida nos veículos acima do seguimento de entrada (OLIVEIRA, 2021). Isso confirma uma tendência de aumento na adesão dessa tecnologia, que passou de 12% de participação de mercado no ano de 2009 para 49% dos carros licenciados no país em 2019 (FELIX, 2021).

As transmissões automáticas encontradas nos carros atualmente podem ser divididas em 3 tecnologias diferentes: conversores de torque, automatizadas e as continuamente variáveis. Esta última é o foco desse projeto. O termo *Continuously Variable Transmission* (CVT), traduz-se para o português como transmissão continuamente variável e representa uma transmissão que possui relação de marchas infinitas (RAGASSI EDSON; LALLI, 2021). A grande vantagem de um carro equipado com transmissão CVT é poder trabalhar em uma faixa de funcionamento que aproveita melhor a força e a potência do motor, que pode beneficiar na economia de combustível e, conseqüentemente, reduzir a emissão de poluentes (SETLUR et al., 2003). A CVT trabalha com duas polias cônicas e uma correia metálica que as relaciona. A polia primária se conecta ao motor e recebe a entrada de força na transmissão; essa força é transmitida para a polia secundária através da correia. Posteriormente, a polia secundária transmite a força para o sistema de tração (DANCZKAY; ALVES, 2015).

Um sistema que implementa a lógica *fuzzy* pode ser dividido em três fases. A primeira delas, chamada “fuzzificação”, é responsável por transformar valores numéricos de entrada em valores linguísticos, por exemplo: velocidade = 200 km/h é convertido para velocidade = alta. A segunda fase, chamada inferência, estipula regras que combinam valores linguísticos de entrada e produz uma saída com um valor também linguístico. Como exemplo desta segunda fase, considere a proposição: SE veículo = potente E velocidade = alta ENTÃO consumo = alto.

A terceira e última fase é a “defuzzificação”, nela, ocorre a tradução de um valor linguístico de saída da fase de inferência para, novamente, um valor numérico, como, por exemplo, inferir que o valor do consumo em 0.9 é alto. Em vista dos fatos apresentados sobre o crescimento da utilização da CVT e a usabilidade da lógica *fuzzy* aplicada em um sistema de controle que otimize o funcionamento deste tipo de câmbio, é notória a possibilidade de um estudo que aplique essa lógica no comando do mecanismo que altera o diâmetro das polias da transmissão CVT. Diante dos fatos contextualizados, esse projeto se justifica pelos benefícios que trará para a sociedade e para o meio ambiente no caso da aplicação da possível solução criada. Trata-se de uma reação em cadeia, pois um veículo que consome menos combustível, automaticamente, emite menos gases poluentes e por consequência agride menos o meio ambiente.

## **1.1 OBJETIVOS**

### **1.1.1 Objetivo Geral**

Este projeto visa implementar a técnica de Inteligência Artificial (IA) chamada lógica *fuzzy*, a ser aplicada ao software de gerenciamento de uma transmissão automática do tipo continuamente variável (CVT), a fim de otimizar a relação de marchas visando à economia de combustível.

### **1.1.2 Objetivos Específicos**

- a) identificar as características do veículo que afetam diretamente o funcionamento da transmissão, que servirão de valores de entrada para o sistema (o processamento destes dados de entrada é o que influencia na precisão do resultado no processo de inferência do sistema *fuzzy*);
- b) identificar o método de inferência mais adequado;
- c) implementar o algoritmo com base nos fatores identificados;
- d) implementar um simulador que gere aleatoriamente valores dentro da faixa operacional de cada parâmetro para testar o algoritmo, interpretar e avaliar os resultados do comportamento dinâmico da abertura e fechamento das polias da transmissão.

## **1.2 JUSTIFICATIVA**

Justifica-se a elaboração deste projeto devido ao fator redução do consumo de combustível. Motivo esse que gera reações consequentes, como a diminuição da poluição e também dos danos ao meio ambiente. Outra vantagem da aplicação desse projeto é a economia. Um veículo que consome menos combustível conseguirá percorrer trajetos maiores com a mesma quantidade de combustível. Isso reduz a necessidade de abastecimento. Algo bastante benéfico visto que, no Brasil, carros híbridos e totalmente elétricos, com baixo ou nenhum nível de emissão de poluentes, são por vezes inviáveis devido a fatores como o alto preço ou poucos postos de abastecimento elétrico.

Realizou-se a pesquisa bibliográfica sobre o tema e também sobre trabalhos realizados a respeito do problema observado, há dificuldade em encontrar algoritmos já produzidos para serem analisados como referência e que abordam o objetivo desse projeto. Alguns trabalhos sobre o mesmo tema foram encontrados e utilizados como fonte de consulta, de resultados e metodologia. No entanto, toda a implementação do algoritmo teve que partir do zero, ou seja, sem alusão a algum código já implementado.

Salvo em casos de códigos abertos (*open source*), que possam existir, implementações correlatas são difíceis de ser encontradas.

Na bibliografia utilizada, considera-se a aplicação do controle *fuzzy* em componentes mecânicos de automóveis comercializados por grandes empresas multinacionais. O desenvolvimento das pesquisas relacionadas envolve áreas das engenharias internas das empresas. O conteúdo e resultados destas pesquisas não se tornam públicos. Tais dificuldades também justificam a elaboração desse projeto, que poderá servir de consulta para futuros trabalhos relacionados.

### **1.3 ESTRUTURA DA MONOGRAFIA**

Este trabalho foi estruturado de forma a apresentar os tópicos importantes relacionados à pesquisa bibliográfica, enfatizando informações sobre todos os pontos que envolvem a proposta, bem como os resultados obtidos por meio de um simulador para avaliação da solução.

O capítulo 2 apresenta a revisão bibliográfica que aborda os principais conceitos da área de pesquisa, embasa de forma teórica e fornece sustentação à parte experimental. O capítulo 3 trata do estado da arte, o ponto de partida de todo o desenvolvimento do trabalho, que leva em consideração o que há de mais recente em termos tangíveis de pesquisa, artigos, projetos ou invenções foram realizados dentro da hipótese abordada. O capítulo 4 apresenta os procedimentos para concretização do sistema proposto. No capítulo 5 são descritos de que forma os testes são realizados, e os resultados referentes a estes testes também são apresentados. Tópicos como: considerações finais, contribuições e trabalhos futuros são descritos no capítulo 6, que apresenta as conclusões.

---

# REFERENCIAL TEÓRICO

---

Este capítulo apresenta, de forma teórica, um conjunto de referências necessárias para se entender o funcionamento da parte mecânica que deve ser controlada pelo software proposto. Portanto, trata-se de uma análise generalizada de como a máquina e o software se relacionam.

## 2.1 O VEÍCULO E A TRANSMISSÃO

A transmissão de um veículo, de uma forma geral, é a responsável por fazer com que a força gerada pelo motor seja transmitida para as rodas. Essa força, relativa ao movimento de rotação de um corpo após a aplicação de determinada força sobre ele, ou seja, o torque do motor, é que faz com que as rodas girem e o carro ande (KIA, 2022).

### 2.1.1 Transmissão Automática

As transmissões automáticas encontradas nos carros atualmente podem ser divididas em 3 tecnologias diferentes: os conversores de torque, as automatizadas e as continuamente variáveis. Esta última é foco do projeto (RAGASSI EDSON; LALLI, 2021).

### 2.1.2 Diferença entre Transmissão Automática e Manual

A caixa de uma transmissão automática tradicional possui um conversor de torque, já a caixa manual possui a embreagem. Esses dois tipos de transmissões têm a mesma função, porém funcionam de forma diferente. A caixa manual precisa ser manipulada pelo condutor por meio da alavanca de câmbio para realizar trocas de marchas e também para acionar o pedal de embreagem. No câmbio automático, a interferência do condutor para este processo não se faz necessária, o trabalho é realizado pela própria máquina e de forma mais precisa (PORTAL, 2022).

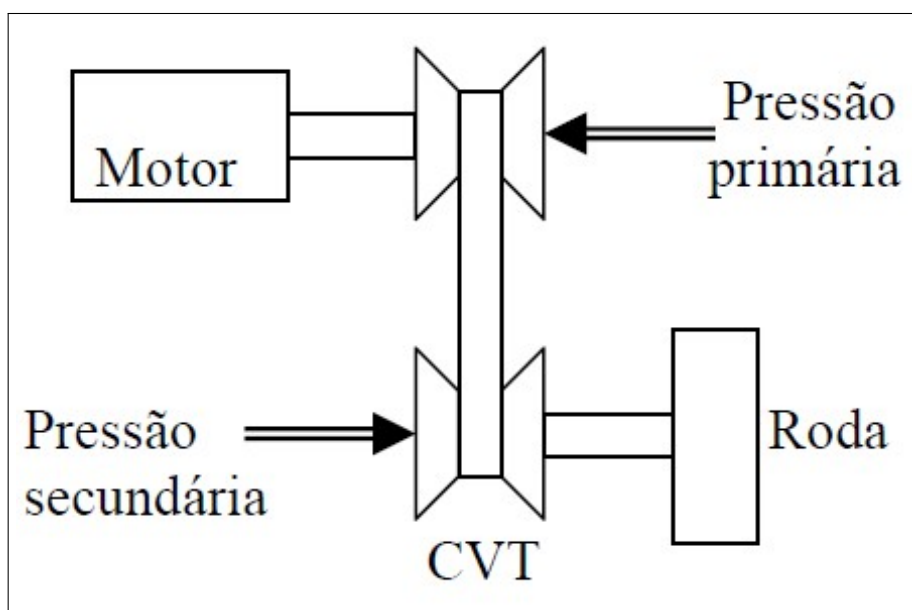
### 2.1.3 Diferença entre Transmissão Automática e Transmissão CVT

A construção e o princípio de funcionamento do câmbio CVT são bem mais simples que o sistema de transmissão convencional. A caixa de transmissão do tipo CVT não possui relação de marchas predefinidas, portanto, não possui um sistema de engrenagens que se interligam. O CVT, por meio de um par de polias com diâmetros variáveis, unidas por uma correia metálica de alta resistência, permite infinitas relações de marchas. O câmbio CVT possibilita o motor se manter em uma rotação de menor consumo de combustível devido a essas infinitas relações de marchas, cujas trocas são realizadas de forma quase imperceptível e que são feitas conforme inferência atribuída pelo sistema (PORTAL, 2022).

### 2.1.4 Transmissão CVT

O termo *Continuously Variable Transmission* (CVT), traduz-se para o português como transmissão continuamente variável e representa uma transmissão que possui relação de marchas infinita (RAGASSI EDSON; LALLI, 2021). A primeira transmissão CVT foi patenteada em 1886 e, desde então, sua tecnologia vem sendo aprimorada e, cada vez mais, desperta o interesse dos fabricantes mundiais de veículos automotores (PORTAL, 2022). A grande vantagem de um carro equipado com transmissão CVT é poder trabalhar numa faixa que aproveita melhor a força e a potência do motor, o que pode maximizar a economia de combustível e, conseqüentemente, emitir menos poluentes (SETLUR et al., 2003). Como apresentado na FIG. 2.1, a transmissão CVT trabalha com duas polias cônicas e uma correia metálica que as conecta. A polia primária se conecta ao motor e recebe a entrada de força na transmissão. Essa força é transmitida para a polia secundária através da correia. Posteriormente, a polia secundária transmite a força para o sistema de tração (DANCZKAY; ÁLVES, 2015).

Figura 2.1: Diagrama de uma CVT



Fonte: Danczkay e Alves (2015)



A abertura e o fechamento dessas polias determinam a relação de rotação que vem do motor e é transmitida para as rodas. A abertura total da polia primária e o fechamento total da polia secundária transferem maior força para as rodas. Esta posição é comum ao arrancar com o veículo em estado de repouso. Continuamente, ao ganhar velocidade, a abertura das polias se invertem: a polia primária se fecha, aumentando seu diâmetro em acordo com a abertura da polia secundária, que o diminui (RAGASSI EDSON; LALLI, 2021). Assim, conforme o motor do veículo aumenta sua velocidade de rotação, um sistema hidráulico comanda simultaneamente a largura adequada das duas polias, ajustando instantaneamente a relação de transmissão para a solicitação do momento. O mecanismo hidráulico que altera o diâmetro das polias, abrindo e fechando-as, é controlado por software, o qual se pode utilizar de lógica *fuzzy*.

### 2.1.5 Tipos de CVT

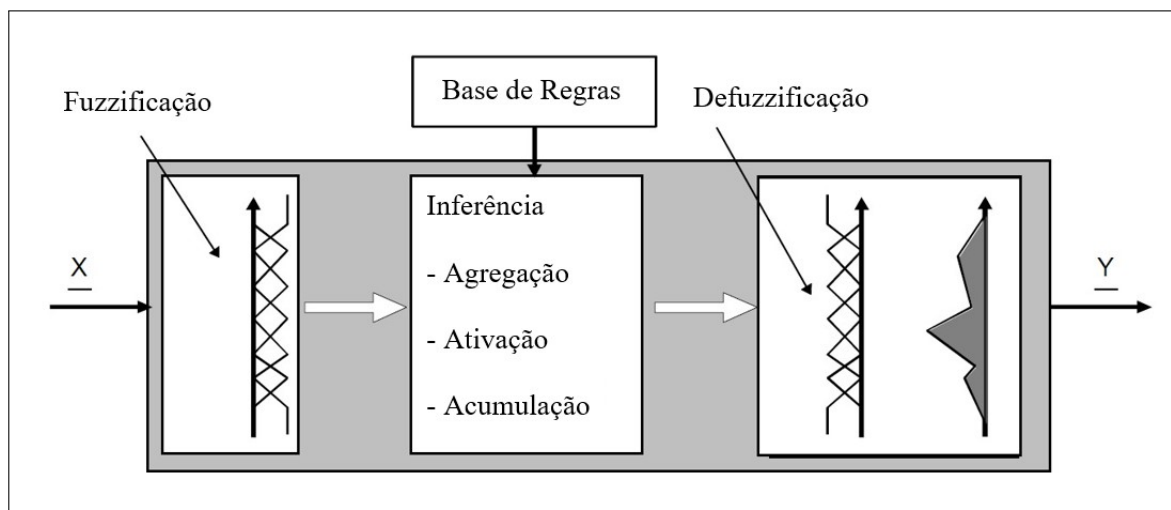
Existem diferentes tipos de transmissão continuamente variável, como CVT de Fricção, Toroidal, Hidrostática e de correia em V. As mais utilizadas são as de fricção e as hidrostáticas. A de fricção é o tipo mais comum de CVT e, portanto, é o utilizado para a proposta deste trabalho. Neste modelo de fricção, duas polias rotativas são colocadas em contato em pontos de distância variável de seus eixos de rotação. Desta forma, a fricção entre elas permite a transferência de giro de uma para outra por meio de uma correia (PORTAL, 2022).

## 2.2 A Lógica Fuzzy

O estudo sobre lógica *fuzzy* foi introduzido em 1965 na publicação *Fuzzy Sets*, do matemático Lofti Asker Zadeh. Este tipo de lógica, também conhecido como lógica difusa, teve grande aceitação devido à facilidade de aplicação em controle de processos (RIGNEL, 2021). O advento da lógica *fuzzy* foi causado pela necessidade de um método capaz de expressar de uma maneira sistemática quantidades imprecisas, vagas e mal definidas (SIMÕES; SHAW, 2007). A lógica *fuzzy* tem possibilidade de tratar incertezas. Logo, valores como muito, pouco, quase, talvez e ligeiramente, são comuns. Diferente da lógica booleana, que admite apenas os valores verdadeiro (1) e falso (0) (RAGASSI EDSON; LALLI, 2021), na lógica difusa, é possível definir graus de pertinência, por exemplo, dizer que, uma proposição é 0.5 verdadeira, isso torna o resultado quase verdadeiro e também quase falso. Outro exemplo análogo é dizer que dada uma proposição, seu resultado é 0.9 verdadeiro, isso significa muito verdadeiro ou até mesmo bem pouco falso (SILVA, 2021). Um sistema que implementa a lógica *fuzzy* é apresentado na FIG. 2.2 e seu funcionamento pode ser dividido em três fases.

A primeira delas, chamada “fuzzificação”, é responsável por transformar valores numéricos de entrada em valores linguísticos, por exemplo, velocidade = 200km/h é convertido para velocidade = alta. A segunda fase, chamada inferência, estipula regras que combinam valores linguísticos de entrada e produz uma saída com um valor também linguístico. Como exemplo desta segunda fase, considere a proposição: SE veículo = potente E velocidade = alta ENTÃO consumo = alto. A terceira e última fase é a “defuzzificação”; nela ocorre a tradução de um valor linguístico de saída da fase de inferência para um valor numérico novamente, como, por exemplo, dizer que consumo é 0.9 alto. Diante das possibilidades de tratamento de processos

Figura 2.2: Estrutura e elementos funcionais do controle *fuzzy*



Fonte: Adaptado de IEC611131-7 (2000)

com a aplicação da lógica *fuzzy*, observa-se a possibilidade de um estudo que aplique este tipo de controle no mecanismo que altera o diâmetro das polias da transmissão CVT. A partir do momento em que a lógica *fuzzy* permitiu facilidades no controle de processos em comparação com controladores programáveis clássicos do tipo Proporcional Integral Derivativo (PID), ampliou-se a utilização desta lógica para o controle de processos. Há referências de alguns estudos comparativos entre PID e *Fuzzy*. As diferenças entre esses dois tipos de controles são descritos com mais detalhes na subseção 2.3.1.

## 2.3 Controladores Programáveis

Na língua inglesa, conhecido como PLC (*Programmable Logic Controller*), o controlador programável (CP) é basicamente um computador industrial, cujo objetivo é o de controlar processos industriais específicos (MORAES; CASTRUCCI, 2006).

Conforme a Associação Brasileira de Normas Técnicas (ABNT), o CLP é um equipamento eletrônico digital, com hardware e software compatíveis com as aplicações industriais a que se destina (ZANCAN, 2011).

No entanto, levam-se em consideração para este trabalho apenas as características lógicas que fazem parte deste controlador que, utilizado até então para o controle de processos automatizados, opera por meio da utilização de métodos matemáticos complexos. Tal complexidade é uma das razões analisadas para que a implementação de um método baseado na utilização da lógica *fuzzy* seja utilizada a fim de mitigar as dificuldades que esse método clássico apresenta.

A utilização de um CP é indicada para sistemas que possuem processos em que as variáveis assumem apenas valores da lógica booleana: verdadeiro (1) e falso (0). Portanto, abrange apenas variáveis digitais que assumem valores dentro de um conjunto finito (LEGASPE, 2012 apud MICHEL, 2007). Esta também pode ser considerada uma das limitações que a utilização do sistema *fuzzy* pode romper, pois, um sistema que dispõe de lógica difusa possibilita, conforme

Simões e Shaw (2007), a capacidade de expressar de uma maneira sistemática quantidades imprecisas, vagas e mal definidas.

### 2.3.1 Controlador Fuzzy X Controlador PID

Ambiguidades que não podem ser processadas pela lógica booleana são o grande empecilho para automatização de atividades humanas. No intuito de solucionar este tipo de problema, em 1965 os conceitos dos conjuntos difusos foram criados por Lofti A. Zadeh, da universidade da Califórnia. Em 1974, o Prof. Ebrahim Mandani, da universidade de Londres, conseguiu controlar uma máquina a vapor por meio do raciocínio difuso após várias tentativas frustradas utilizando diversos tipos de controladores, incluindo PID. O primeiro exemplo de uma aplicação industrial baseada em controle *fuzzy* foi realizado em 1980. A partir de então diversas aplicações surgiram (CALDAS; LEONARDI, 1999).

Otimizar custos de implementação é um dos benefícios da estratégia de controle da modelagem *fuzzy*. Além da possibilidade de resolver problemas de otimização controlando sistemas de equações e desigualdades através de seus parâmetros, beneficia-se também da abordagem de técnicas de programação não linear. Portanto, evita a necessidade de uma grande formação teórica e cálculos complexos, como é feito em práticas de controles clássicos. As técnicas de lógica *fuzzy* criaram facilidades capazes de incluir pessoas, propriamente, como especialistas, que de forma empírica, podem colaborar com informações e dados específicos, pontuais e precisos na prática (IEC611131-7, 2000) (SIMÕES; SHAW, 2007).

#### Conceito do Controlador Fuzzy

Baseados em regras *fuzzy*, os controladores lógicos *fuzzy* (CLFs) são modelos específicos adequados às aplicações nas engenharias para as quais as estratégias clássicas de controle não alcançam bons resultados ou quando há muita dificuldade de se obter um modelo matemático. Encontra-se na parte 7 da norma IEC 61131, publicada pela Comissão Eletrotécnica Internacional, um padrão de programação de controle *fuzzy* que define características básicas de entendimento para integrar aplicações de controle *fuzzy* a sistemas de controle (CINGOLANI; ALCALÁ-FDEZ, 2013) (IEC611131-7, 2000). A linguagem estratégica, baseada em experiência e conhecimento, fornecida por um especialista é convertida por um processo do controlador *fuzzy* em estratégia de controle automático. Portanto, a necessidade de este controlador dispor de um conjunto de regras criadas por um humano e a forte abordagem não linear dos dados são suas principais características. Os CLFs são poderosos em resolver uma vasta gama de problemas, mas necessitam de grande especialização para que sejam programados. Muitas ferramentas de lógica *fuzzy* já foram desenvolvidas para facilitar a programação desses controladores. O MATLAB Fuzzy Logic Toolbox é um exemplo de ferramenta disponibilizada comercialmente (CINGOLANI; ALCALÁ-FDEZ, 2013). A jFuzzyLogic, disponível como *software* de código aberto, é também um exemplo de ferramenta e foi utilizada para o desenvolvimento do simulador.

### 2.3.2 Linguagem de Controle Fuzzy (FCL)

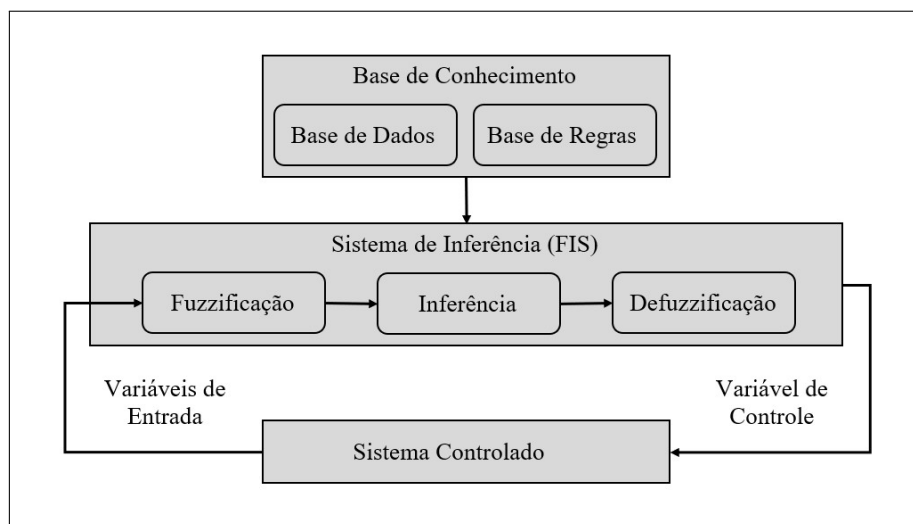
O padrão de controle *fuzzy* publicado na norma IEC 61131-7, que define as características básicas de entendimento para integrar aplicações de controle *fuzzy* a sistemas de controle, como já dito, trata-se da Linguagem de Controle Fuzzy ou *Fuzzy Control Language* (FCL), no inglês.

A FCL oferece uma linguagem comum de controle com alcance global e independente, além de possuir portabilidade, que possibilita ser utilizada na implementação de sistemas em diferentes plataformas ou utilizando diferentes linguagens de programação. Possui sintaxe semelhante a uma linguagem estrutural de texto, utiliza um módulo básico programável, que é o FIS, e também um tipo especial de bloco de função, sem a necessidade de execução ordenada (CINGOLANI; ALCALÁ-FDEZ, 2013). Utilizou-se a jFuzzyLogic, biblioteca Java, abordada na subseção 2.3.3, utilizada neste trabalho para desenvolver a FCL conforme o padrão citado.

#### Estrutura de Funcionamento da FCL

Conforme a FIG. 2.3 apresenta, a FCL é composta por uma base de conhecimento que compreende as informações usadas pelo operador especialista na forma de regras de controle linguístico. Uma interface de “fuzzificação” transforma valores das variáveis de entrada em conjuntos *fuzzy* que serão utilizados no processo de inferência. Os valores “fuzzificados” e a base de conhecimento são utilizados por um sistema de inferência para processar os dados. A “defuzzificação”, último processo, traduz os valores inferidos em valores nítidos para a variável de controle (CINGOLANI; ALCALÁ-FDEZ, 2013).

Figura 2.3: Estrutura genérica da FCL



Fonte: Adaptado de Cingolani e Alcalá-Fdez (2013)

A base de conhecimento inclui a base de dados (termos linguísticos e funções de associação) e a base de regras (conjunto de regras para o controle). A “fuzzificação” mapeia os valores da variável de entrada para que seja determinado o seu conjunto *fuzzy*. O sistema de inferência utiliza os conjuntos difusos para “raciocinar” o processo. A “defuzzificação” recebe o processamento da inferência e o traduz em valores nítidos para a variável de saída ou de controle.

### 2.3.3 jFuzzyLogic

Existem diversas ferramentas disponíveis para a modelagem de um sistema *fuzzy*. Um estudo realizado por Arruda et al. (2019) apresenta uma análise comparativa entre essas possibilidades, que possuem diferentes perfis de utilização num contexto de desenvolvimento. Aspectos como licença para utilização, linguagem de programação, recursos gráficos, exemplos de aplicações e documentação simples são características que fizeram optar pela utilização da jFuzziLogic.

A jFuzzyLogic é uma biblioteca Java de código aberto que permite desenvolver FCLs seguindo o padrão da norma IEC 61131-7. Ela oferece uma solução funcional e completa para a implementação de um sistema de inferência *fuzzy* (FIS). Apresenta vantagens como: redução no trabalho de programação e do nível de conhecimento do controle de lógica *fuzzy* por parte do especialista, facilita o entendimento dos desenvolvedores por possuir abordagem estritamente orientada a objetos, juntamente com uma estrutura de módulos, e permite desenvolver FCLs em qualquer sistema operacional que suporte Java (CINGOLANI; ALCALÁ-FDEZ, 2013).

Na norma citada, toda a padronização e processos necessários para que a FCL seja construída são determinados. São especificadas cinco formas de linguagens para controladores programáveis. Dentre as cinco, duas são textuais: Lista de Instrução e Estrutura de Texto; e três são gráficas: Diagrama de Escada, Diagrama de Bloco de Função e Gráfico Sequencial de Funções. Todas elas são modulares e possuem o módulo básico, ou unidade programável, que inclui programas, funções ou bloco de função. Geralmente o sistema é composto por várias unidades programáveis que podem ser implementadas por meio de linguagens textuais ou gráficas. A norma define os tipos de dados que podem ser utilizados (CINGOLANI; ALCALÁ-FDEZ, 2013). Diante dessa padronização, a programação se apresenta como um simples conjunto de instruções e variáveis.

### 2.3.4 Sistema de Inferência Fuzzy (FIS)

Geralmente um FIS é composto por um ou mais blocos de função. Cada bloco possui as seções de variáveis de entrada, variáveis de saída, associação de “fuzzificação” de funções, associação de “defuzzificação” de funções e a seção com o bloco de regras difusas. Na seção 4.2 é apresentada de forma detalhada como o FIS atua diante do contexto do propósito deste trabalho.

---

## ESTADO DA ARTE

---

Durante toda a busca por referências bibliográficas sobre o tema, a fim de se conhecer o estado da arte relacionado ao objetivo deste trabalho, foi possível encontrar algoritmos já produzidos para controle de automação, mas não um algoritmo específico para o controle das polias de um CVT utilizando lógica *fuzzy*. Alguns trabalhos sobre a mesma hipótese foram utilizados como fonte de consulta, de resultados e de metodologia. No entanto, toda a implementação do algoritmo teve que partir do zero, ou seja, sem referência de um código já implementado.

Os trabalhos encontrados sobre a aplicação de controle *fuzzy* em transmissão CVT de automóveis, que são comercializados por grandes multinacionais, envolvem áreas das engenharias internas dessas empresas, e o conteúdo explícito da implementação do sistema nestes casos não se torna público. O mesmo ocorre para todos os outros trabalhos relacionados, cujos códigos não são abertos.

Portanto, o estado atual de conhecimento sobre a hipótese analisada faz referência aos trabalhos relacionados a seguir, que possuem, em comum, a utilização de um controlador *fuzzy* Proporcional Integral (PI).

### 3.1 Proposta de Controlador Fuzzy para Transmissão Continuamente Variável

Trata-se de um artigo da dissertação de Gyozo Zoltán Danczkay, apresentado à Escola Politécnica da Universidade de São Paulo, para obtenção do Título de Mestre em Engenharia Automotiva (DANCZKAY; ÁLVES, 2015). Nesse artigo, o autor apresenta uma metodologia para o projeto de um controlador *fuzzy* aplicado a uma CVT para obter economia de combustível. No entanto, a motivação era a de implementar uma estrutura que utilizava uma base de regras equivalentes a de um controlador *fuzzy* PI e, não somente, um controlador *fuzzy*. Danczkay defendeu sua tese no ano de 2005, já fazendo parte da equipe de funcionários da engenharia

de manufatura da General Motors do Brasil. Concluiu, por meio de simulação, que a utilização do sistema *fuzzy* controlou o motor em seu ponto de eficiência máxima. O sistema também apresentou um bom controle sobre a velocidade do veículo e sobre a dinâmica de operação da CVT quando os parâmetros de carga e condições da estrada variavam. O controlador *fuzzy* atende integralmente às condições exigidas tanto pelo motor quanto pela CVT.

### **3.2 Controlador Fuzzy Aplicado a Veículos Elétricos com Transmissão Variável Contínua**

Trata-se de um artigo escrito por Marcelo A. C. Fernandes, professor no Departamento de Engenharia de Computação e Automação da Universidade Federal do Rio Grande do Norte (UFRN) no ano de 2014. Fernandes propõe uma estratégia de controle de eficiência energética em veículos elétricos equipados com CVT. A estratégia utiliza um controlador *fuzzy* que atua diretamente na relação de transmissão do CVT tendo como referência o valor de rotação do motor para máxima eficiência por meio de seus parâmetros elétricos e mecânicos. No entanto, essa proposta também utiliza estratégia *fuzzy* PI, cujas entradas do sistema *fuzzy* são caracterizadas pelo erro e variação do erro (FERNANDES, 2014).

Por meio de simulação utilizando o modelo de um veículo longitudinal acoplado a um motor de corrente contínua que possui um ímã permanente (PMCD - *Permanent Magnet Direct Current*), juntamente com o sistema CVT, seu trabalho apresentou resultados em que o controlador *fuzzy* permitiu uma maior eficiência no funcionamento do motor. Com isso, uma maior economia no consumo de potência. Com os resultados obtidos, Fernandes acredita que o sistema de controle proposto funciona de forma satisfatória e é bastante promissor para aplicações práticas.

### **3.3 Método e Sistema Utilizando Lógica Fuzzy para o Controle de uma Transmissão CVT**

Trata-se de uma patente de invenção estadunidense em que os autores analisaram as operações executadas pelo motorista juntamente com o estado em que o veículo se encontra. Um valor alvo de velocidade a ser utilizado na entrada da transmissão a fim de promover maior desempenho ao veículo é determinado com base nos valores gerados pelas ações do condutor e também pelo estado de funcionamento do automóvel. A lógica *fuzzy* é empregada para controlar a relação da velocidade a partir da utilização de diferentes algoritmos e conjuntos de regras, que tem como base a magnitude do desvio entre o valor alvo e o valor real (ADACHI et al., 1996). Tem como objetivo apresentar um sistema de controle *fuzzy* aplicado a uma transmissão CVT para suavizar a sensação de aceleração e desaceleração (trocas de marchas) pelo condutor do veículo, o que automaticamente evitará o desperdício de energia do motor.

Nesse trabalho, os autores realizam testes comparativos entre a dinâmica de um sistema de controle convencional, estado da arte na ocasião (1988), e os benefícios do método de controle *fuzzy* proposto aplicado à uma CVT. São relacionados vários resultados satisfatórios obtidos,

dentre esses a eliminação de mudanças bruscas na rotação do motor, a melhora na dirigibilidade e mais conforto para o motorista. A economia de combustível é também um resultado obtido devido à eficiência que o controle proposto realiza.



---

# METODOLOGIA

---

No intuito de validar a hipótese desse projeto, foi utilizado o método exploratório de estudo, a fim de compreender o funcionamento de todos os elementos que abrangem o sistema CVT. O capítulo Metodologia apresenta as ações realizadas para concretizar o sistema proposto e descreve, de forma técnica, as diretrizes utilizadas baseadas em toda a revisão bibliográfica.

## 4.1 Análise dos Métodos de Controle

No início da pesquisa sobre o controle da CVT utilizando a lógica *fuzzy*, verificou-se, na literatura, uma grande variedade de trabalhos relacionados aos controladores programáveis. Em grande parte desses trabalhos, a abordagem está mais relacionada ao conjunto completo de componentes que envolvem o sistema de funcionamento do CVT, que engloba toda a parte física de engenharia e automação. Para tal sistema, a maior parte dos trabalhos utiliza controladores tradicionais como Proporcional Integral (PI) e Proporcional Integral Derivativo (PID), que enfatizam o erro e a variação do erro como variáveis de entrada e que necessitam de uma análise matemática complexa.

Juntamente com um desses modelos de controladores, é também utilizado em conjunto um controlador *fuzzy*, cujas variáveis de entrada são parâmetros das funções de pertinência no pré-processamento e estão relacionados às variáveis que irão fornecer os dados para o ajuste do sistema (DANCZKAY; ÁLVES, 2015).

Para esse projeto, os controladores tipo PI e PID serão analisados de forma superficial. O foco da pesquisa é a área relacionada a conceitos e disciplinas abordadas no curso de Ciência de Computação e não das engenharias, que envolvem estudos mais profundos em relação a cálculos mais complexos. Tópicos esses que a implementação do controlador *fuzzy* dispensa por se tratar de uma técnica de controle mais prática, conforme Legaspe (2012):

[...]um dos grandes empecilhos em se aplicar a teoria clássica de controle reside

na dificuldade de cálculo do controlador. Para isso a lógica *fuzzy* facilita em muito a construção de controladores de processos industriais complexos, de forma mais simples e de comportamento mais "inteligente".

## 4.2 Modelagem do Sistema

A modelagem do sistema proposto parte do funcionamento geral das partes que compõem o funcionamento da transmissão CVT em um veículo. Dentre estas partes há: o motorista, o próprio veículo com suas partes atuantes, a transmissão CVT e o sistema de controle *fuzzy*.

Conforme Adachi et al. (1996) esse sistema é representado na FIG. 4.1. Trata-se da estrutura de funcionamento do controlador *fuzzy* em um veículo conduzido pela transmissão da força motriz de um motor (1) que é a fonte de energia para movimentar uma roda (5) a partir de uma embreagem (2), um conversor de torque (3) e uma correia de transmissão hidráulica de uma transmissão continuamente variável 4. Um microcomputador (8) obtém e emite um valor de saída (24) para o controle do motor de parada (9), com base nas informações de abertura do acelerador (21), que é medida por um sensor de operação do condutor do veículo (6). A informação de velocidade do veículo (22), medida pelo sensor de status de movimento do veículo (7) e a informação sobre a velocidade de rotação (23) da polia primária do CVT (4). O mecanismo da relação de velocidade do motor de passo (9) permite o controle simples e preciso da posição angular de seu eixo. Desta forma é possível obter também o controle preciso da velocidade e do torque do motor.

O motor de passo (9) diminui a relação de velocidade entre as polias primária e secundária movendo a válvula de controle da relação de velocidade (12) através do conversor (10) e da haste da válvula (11), e também inserindo uma alta pressão de óleo na metade móvel (19) da polia primária do CVT (4) ou aumentando a relação de velocidade através do retorno do óleo desta mesma metade móvel para o reservatório (15). Uma das extremidades da haste da válvula (11) fica em contato com a superfície cônica interna da metade móvel da polia primária (19), e a outra extremidade fica em contato com o eixo conversor (20) do conversor (10). O torque produzido pelo CVT (4) é transmitido sequencialmente da polia primária (17) para a correia de transmissão (16) e também para a polia secundária (18). Em seguida, o conteúdo de processamento do microcomputador, parte do sistema integrado pelo controlador *fuzzy*, objeto deste trabalho, opera com o objetivo de promover o controle de eficiência econômica de combustível desejada.

Conforme o fluxograma, o microcomputador (8) faz a leitura das informações processadas pelo sensor (6) sobre a abertura do acelerador (21), a leitura das informações sobre a velocidade do veículo (22) e também a informação sobre a velocidade de rotação da polia primária (23). A velocidade do veículo (22) e a velocidade de rotação da polia primária (23) são informações transmitidas pelo sensor de status de funcionamento do veículo (7). Em seguida, essas referências são medidas pelo bloco de leitura de dados (101), que calcula a velocidade alvo de rotação da polia primária. O bloco de leitura leva também em consideração as informações sobre a abertura do acelerador (21) e a velocidade do veículo (22). Após, o processo segue para o bloco

(102), que determina a diferença entre a velocidade alvo de rotação da polia primária (calculada pelo bloco (101)) e a velocidade de rotação corrente da polia primária.

Para o bloco (101) calcular a velocidade alvo de rotação da polia primária, o microcomputador (8) processa os dados de informação sobre a velocidade do veículo (22) e a abertura do acelerador (21) antecipadamente, por meio de um cálculo utilizando esses dados. No bloco de verificação do desvio (102), o microcomputador (8) determina a diferença entre a velocidade alvo de rotação da polia primária, determinada pelo bloco (101), e a velocidade de rotação da polia primária. A diferença entre essas velocidades é chamada de desvio. Sendo o resultado do valor absoluto do desvio apurado, esse é transmitido para o bloco (103) se for maior ou igual a uma constante predefinida. Ou para o bloco (104) do microcomputador (8), caso o valor seja menor do que essa constante. Diante disso, a diferença da posição alvo do motor de parada pode ser calculada. Para tal, existe um conjunto de regras da lógica *fuzzy* disponíveis, que relacionam as características das variáveis do sistema para obter um valor de saída a ser direcionado para o bloco que irá calcular o valor para operação do motor de partida (105). Como exemplo, pode-se citar a regra: Se o erro é positivo e pequeno e a variação do erro é positiva e pequena, o microcomputador (8) irá permitir que o motor de parada opere moderadamente na direção em que a relação da velocidade tende a diminuir.

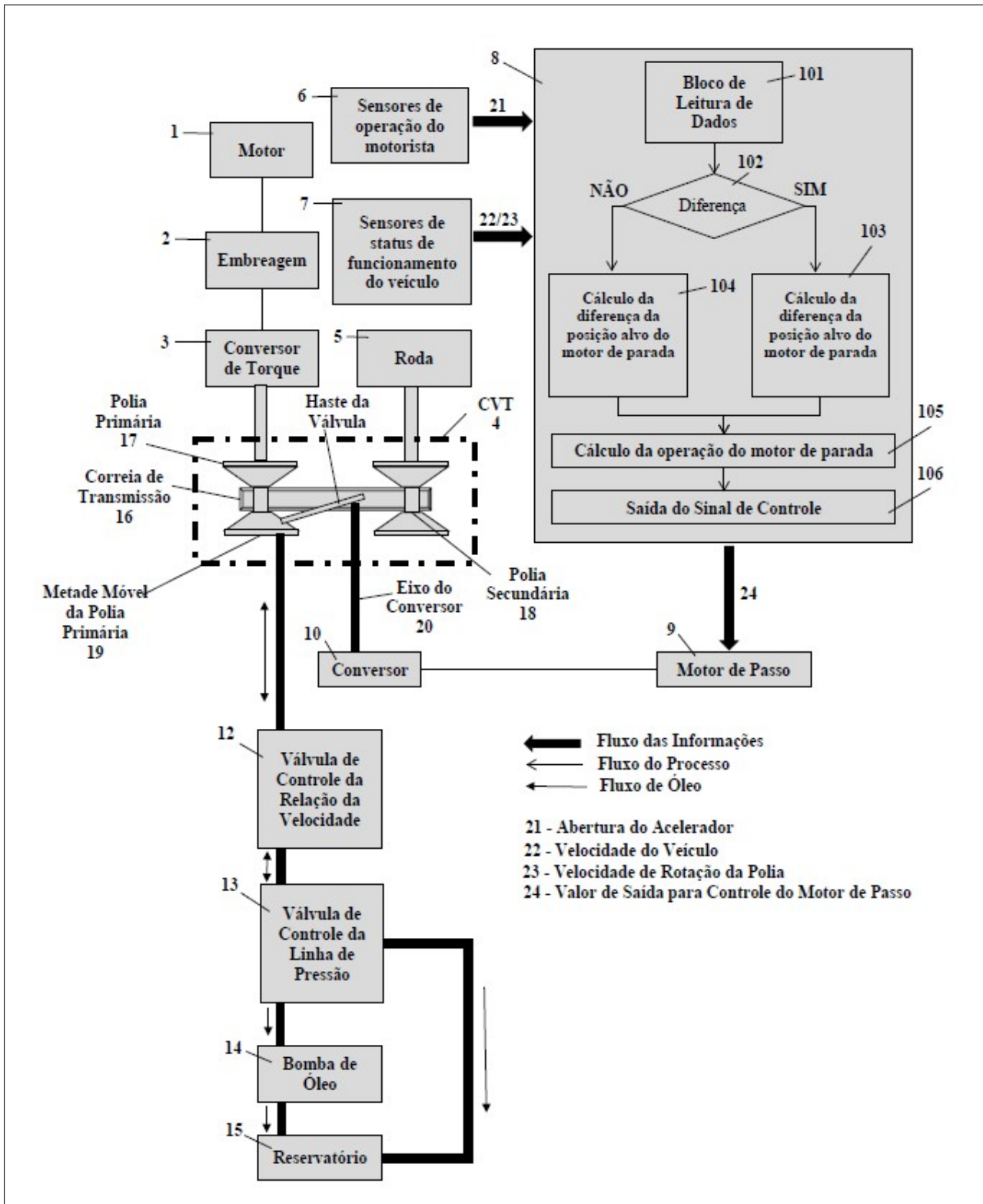
A descrição do processo conforme esse diagrama é feita de forma bastante resumida. Internamente, cada bloco possui estruturas com procedimentos detalhados. Nelas, cálculos matemáticos complexos operam para gerar um resultado que será transmitido em sequência. Esse mesmo resultado se torna um novo parâmetro para os próximos cálculos que serão realizados.

O processamento que ocorre no bloco (105) gera o valor de operação para o motor de parada. O bloco (106) recebe esse resultado e o analisa para a instrução seguinte. Quando o valor é 0, o microcomputador (8), de acordo com instruções internas, mantém o motor de passo desligado. Quando o valor é 1, o microcomputador (8) emite instruções para girar o motor de passo na posição em que a relação de velocidade diminua. Quando o valor é -1, o microcomputador (8) emite instruções para girar o motor de passo na posição em que a relação de velocidade aumente. Tem-se, então resultados baseados no erro e na variação do erro, além do sistema *fuzzy* que é agregado dentro dessa estrutura de controle.

Entender o funcionamento de todo o conjunto que abrange a operação da CVT, mesmo que dessa forma resumida, descrita com base no trabalho de Adachi et al. (1996), é importante para que, a partir de então, seja possível compreender como o controlador proposto irá atuar dentro desta sistemática.

O microcomputador (8) que emite o valor de saída para controle da CVT utiliza, como dito na subseção 2.3, diferentes algoritmos, conjuntos de regras complexas, dentre outros cálculos para processar os dados e inferir um resultado. O modelo de microcomputador, proposto nesse trabalho, utiliza também o controle *fuzzy*. Porém, leva em consideração apenas as características práticas e vantajosas do método de controle que utiliza o FIS, a FCL e a biblioteca jFuzzyLogic.

Figura 4.1: Diagrama de Blocos



Fonte: Adaptado de Adachi et al. (1996)

## 4.2.1 Linguagem de Programação Utilizada

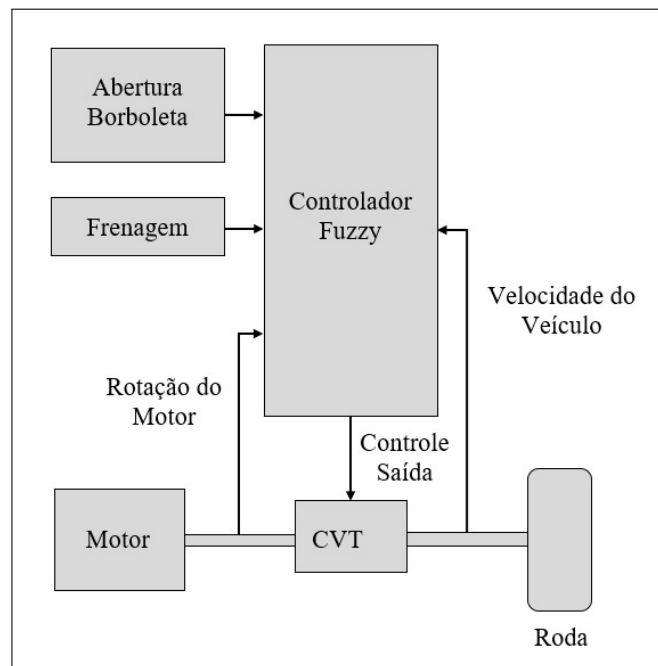
O estudo de Sebesta (2018) aponta que a implementação de um algoritmo em uma linguagem de programação inadequada para aquele domínio compromete a produtividade e o desempenho do projeto.

Possibilitar a conversão de códigos que precisam ser instalados em sistemas operacionais distintos por meio de uma máquina virtual, ou seja, ser multiplataforma; além de possuir uma grande comunidade e também grande quantidade de *frameworks* são fatores que contribuem para que a linguagem Java continue sendo a mais popular dos últimos vinte e cinco anos (ADEKZTI, 2022) (SOUJAVA, 2022).

## 4.2.2 Estratégia de Controle

A FIG. 4.2 detalha, em diagrama de blocos, a estrutura da estratégia de controle para fechamento e abertura das polias do CVT, ajustadas de forma dinâmica pela alteração da relação. A relação produz o torque que é transmitido para as rodas do veículo de acordo com a eficiência sobre o consumo de combustível promovido pelo controlador. Este controlador utiliza de estratégia *fuzzy*, cujas variáveis de entrada do sistema são detalhadas na subseção seguinte.

Figura 4.2: Diagrama de blocos da estratégia de controle



Fonte: Adaptado de Fernandes (2014)

## 4.2.3 Parâmetros de Entrada para o Controlador Fuzzy

Como já dito, o controle *fuzzy* é recomendado para problemas não lineares, pois possibilita operar sistemas por meio de seus parâmetros. Definir os parâmetros a serem utilizados no controlador proposto é um dos objetivos específicos deste trabalho, motivo este de ter sido feito

um levantamento teórico sobre o funcionamento da transmissão CVT em um veículo e todas as partes que a envolvem. Ao final da pesquisa bibliográfica, definiram-se quatro parâmetros do veículo que são as variáveis de entrada para o sistema: rotação do motor, abertura da borboleta (injeção ou consumo de combustível), frenagem e velocidade do veículo. Esses parâmetros ou argumentos são valores a serem inseridos no sistema e processados por uma função que faz parte do FIS.

Para a função de pertinência de cada variável linguística do FIS é preciso definir o grau de associação das características de cada variável, ou seja, o quanto esta característica pertence ao conjunto dentro dos valores difusos que a associação gera. Na ordenada o valor fica entre 0 e 1 (grau de associação) e, na abscissa, o valor se refere à faixa de domínio do parâmetro. O intuito é utilizar pares de valores para os pontos de associação de cada parâmetro que condizem com a realidade, no entanto, são valores supostos, com o sentido único de serem analisados na geração dos testes e resultados de forma experimental. Para as variáveis definidas, apresentam-se com as características que seguem:

1. Rotação do motor: faixa de intervalo de 0 a 6500 rotações por minuto (RPM). Sendo que a 900 RPM, a rotação do motor é caracterizada por estar em repouso; e, em 6500 RPM, a velocidade do veículo é máxima;
2. Abertura da válvula borboleta: atua por meio do acelerador, também interpretada como consumo de combustível. Sua faixa de operação se dá dentro do domínio de 0 a 100. Em 0 a abertura da válvula borboleta estará completamente fechada. A 100, sua abertura será total, posição em que a aceleração será máxima.
3. Frenagem: pressão de frenagem na faixa de 0 a 10 bar (unidade de pressão);
4. Velocidade do veículo: faixa de 0 a 180 quilômetros por hora (km/h).

#### 4.2.4 Método de Inferência

O método Mandani é uma das técnicas de inferência *fuzzy* mais utilizada (WILHELM, 2000). Neste método, o processo de “fuzzificação”, descrito na subseção 4.3.3, transforma valores reais de uma variável de entrada em um conjunto difuso, que, na sequência, é “defuzzificado”, processo descrito na subseção 4.3.5, e resulta em um valor real para a saída. Todo o processamento das informações é feito em termos linguísticos conforme o conjunto de regras modelado para o processo de inferência (subseção 4.3.4), que se caracteriza por utilizar regras condicionais SE e ENTÃO (MACHADO, 2003).

SE  $x$  é A (premissa)  
ENTÃO  $y$  é B (consequente)

Sendo  $x$  a variável de entrada e  $y$  a variável de saída, A e B são os termos linguísticos que descrevem essas variáveis.

Com uma estrutura mais apropriada para aplicações em engenharias, em que as entradas e saída também são variáveis reais, como no método Mandani, foi proposto por Takagi e Sugeno (1985) o sistema difuso Takagi-Sugeno (TS), que dispõe das mesmas diretrizes SE e ENTÃO, mas o consequente é substituído por uma fórmula que facilita a combinação de regras (COSTA, 2018 apud TAKAGI; SUGENO, 1985).

$$\begin{aligned} \text{SE } x \text{ é } A \text{ (premissa)} \\ \text{ENTÃO } y = Cx \text{ (consequente).} \end{aligned}$$

O método TS suporta variações de parâmetros devido à característica não linear do sistema. Portanto, possui a estrutura adequada para aplicações em controle.

A norma IEC61131-7 (2000) especifica os métodos que podem ser utilizados para gerar o valor de saída na “defuzzificação”: COG (*Centre of Gravity*), COGS (*Centre of Gravity for Singletons*), COA (*Centre of Area*), LM (*Left Most Maximum*) e RM (*Right Most Maximum*).

Os princípios e definições utilizados neste trabalho são válidos para um caso simplificado de base de regras *fuzzy* e leva em consideração a inferência Mamdani. Aplicações de inferência mais complexas não estão no domínio deste modelo.

O COGS calcula o centro de gravidade para *singletons* ou seja, a centróide da área composta que representa o termo de saída *fuzzy*, que é formada pela união de todas as possibilidades promovidas pelo bloco de regras. O cálculo desse método é dado por:

$$U_{(t_k)} = \frac{\sum_{i=1}^P [U_i * a_i(t_k)]}{\sum_{i=1}^P [a_i(t_k)]}$$

Onde:

$U_{(t_k)}$  : resultado da “defuzzificação”

$a_i$  : valores *singletons* das regras individuais

Para o caso de *singletons*, o centro da área não é aplicável. A FIG. 4.3 apresenta um exemplo da condição em que quatro *singletons* fazem parte de uma função em que o resultado da “defuzzificação” ( $U_{(t_k)}$ ) não é o centro da área.

Onde:

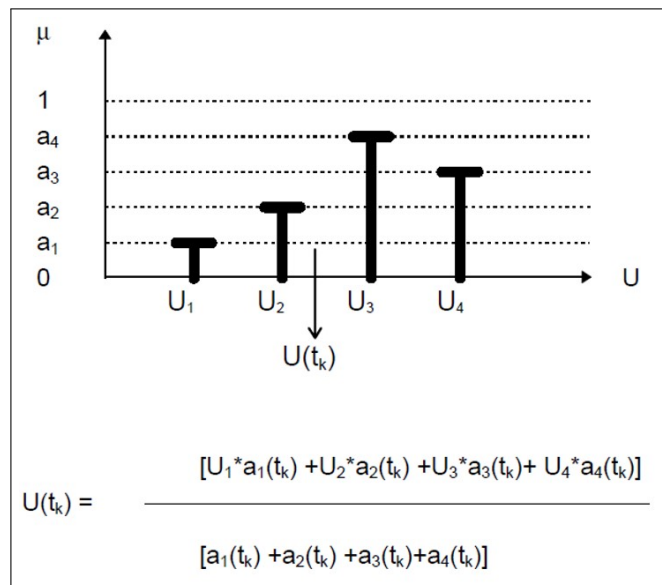
$\mu$  : função de associação após acumulação

$U$  : defuzzificação

$U_{(t_k)}$  : resultado da defuzzificação

A FIG. 4.4 demonstra a relação de como o resultado para um processo que utiliza o centro da área (método COA) e o resultado para o que utiliza o centro de gravidade (método COG) se comportam. No caso do COA, a saída inferida é o valor da abcissa no ponto de encontro da função de pertinência entre duas áreas de igual tamanho, conforme a situação exemplificada.

Figura 4.3: Exemplo em que o centro da área não é aplicável

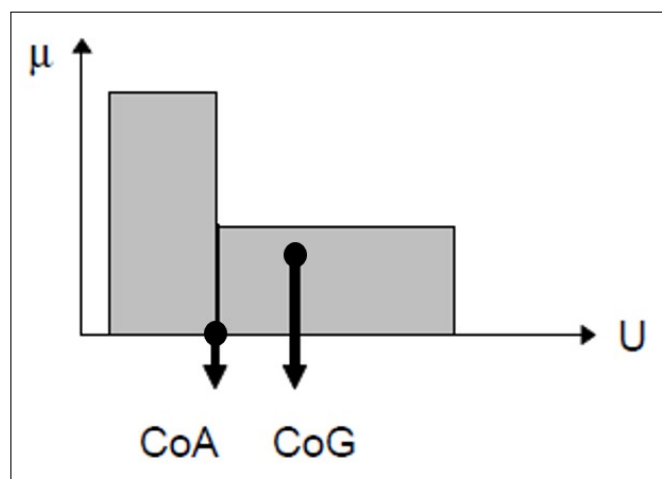


Fonte: IEC611131-7 (2000)

Ou seja, o cálculo utilizado nesse método leva em consideração o bissetor (divisão em duas partes iguais). O COA não se aplica quando *singletons* são utilizados. Já no caso do COG, o cálculo se dá por meio da centroide da área que engloba todo o conjunto difuso em análise da função de pertinência. (IEC611131-7, 2000).

O método utilizado para o controle no sistema proposto é o COGS. Entende-se o seu funcionamento da mesma forma que o COG, porém, com a utilização de *singletons* e não de termos para a função de pertinência de uma variável.

Figura 4.4: Relação entre os resultados para o método COA e COG



Fonte: Adaptado de IEC611131-7 (2000)

Conforme as características de cálculos que o método COGS apresenta, é possível abranger todo o subsídio de dados disponíveis na implementação do arquivo de controle a fim de obter um

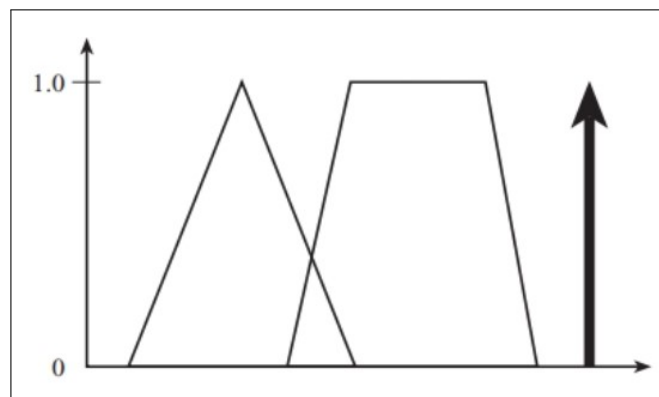


resultado adequado para a saída. Os métodos que envolvem características máximas ignoram as áreas das funções de pertinência das variáveis de entrada valorizando apenas a pertinência de forma pontual (SIMÕES; SHAW, 2007).

Considerando os aspectos que o COGS implementa, este é o método utilizado para inferir o resultado de controle no simulador proposto.

Utiliza-se o formato triangular, para o qual são definidos três pontos de coordenadas para o termos linguísticos nas funções de associação. Empregar função triangular caracteriza boa prática para a formação dos conjuntos difusos que irão relacionar as variáveis de entrada como mostrado na FIG. 4.5. Funções complexas nem sempre apresentam melhores resultados para a saída (SIMÕES; SHAW, 2007). A variável de saída tem *singletons* para a sua função de pertinência. Para este caso, o método COGS aplica um cálculo diferenciado, levando em consideração regras individuais para esses pontos. Esta característica particular de cálculo será abordada com detalhes na subseção 4.3.5.

Figura 4.5: Exemplos de funções de pertinência triangular e trapezoidal



Fonte: Simões e Shaw (2007)

## 4.3 Simulador

A implementação do simulador permite testar o algoritmo. Pode-se criar e também manipular situações em que o sistema da CVT possa ser submetido. Tais circunstâncias podem relacionar as variáveis de entrada com a aplicação de valores específicos para cada uma delas, com o objetivo de gerar um cenário de percurso ou contexto em que o veículo esteja inserido.

Implementou-se o controle promovido por esse simulador a partir da ferramenta para modelagem de lógica *fuzzy*, a jFuzzyLogic, descrita em detalhes na subseção 2.3.3, que é uma biblioteca escrita em linguagem de programação Java e implementa a FCL. Utiliza-se da linguagem de programação Java para executar o algoritmo pelo simulador por meio do ambiente de desenvolvimento NetBeans IDE na plataforma Windows. Construiu-se o simulador para gerar valores realistas para o domínio do problema. De acordo com UOL (2022) “Não existe uma rotação ideal para todos os veículos. O ideal é que você saiba qual é a faixa de rotação na qual seu veículo apresenta o melhor rendimento. Na maioria dos modelos compactos, esta

faixa costuma ser entre 1.800 rpm e 2.000 rpm”. Wilhelm (2000) diz que cada motor possui sua curva de consumo específica. Tal curva apresenta a faixa de rotação em que o motor consome menos. Essa, então, é a faixa ideal para que a transmissão trabalhe em prol da economia de combustível.

Consideram-se, para esse simulador, os valores para a rotação do motor no intervalo de 900 a 6500 rotações por minuto (RPM). Esse intervalo corresponde à rotação do motor em repouso e em aceleração máxima, respectivamente.

Os blocos de funções criados para o arquivo FCL conforme IEC611131-7 (2000) especificam as variáveis de entrada e a de saída, seus parâmetros, as regras e as declarações específicas para o controle *fuzzy*.

Todos os dados e informações inseridos no arquivo FCL são processados pelo sistema *fuzzy* de controle, que no primeiro processamento, “fuzzifica” (as variáveis de entrada, juntamente com seus parâmetros de associação são convertidas em funções de pertinência), e, na sequência, “defuzzifica” (a variável linguística é convertida em valor para gerar a saída). Como forma de simplificar a “defuzzificação” foram utilizados pontos únicos, ou *singletons* para a função de associação apenas para a variável de saída. A vantagem de utilizar *singletons* para a função de pertinência para a variável de saída é comentado na subseção 4.3.5.

O método COGS utilizado no processo *fuzzy* deste projeto, como descrito na subseção 4.2.4, dispõe de toda a parte estrutural do arquivo FCL criado, portanto, do bloco de função, que possui dados sobre todos os elementos (variáveis de entrada, variável de saída e conjunto de regras) necessários para o processamento e execução do sistema.

As regras *fuzzy* devem ser criadas por especialistas (SIMÕES; SHAW, 2007). Para a implementar o simulador não se utilizou da expertise desse tipo de profissional, mas sim do conhecimento dos dados obtidos neste trabalho por meio das referências teóricas e análises realizadas sobre o funcionamento da CVT. Criaram-se regras de forma objetiva para gerar o resultado e realizar testes sobre os aspectos mais importantes e que influenciam de forma efetiva no controle da abertura e fechamento das polias da transmissão.

### 4.3.1 Implementação do simulador

O início do processo de implementação do simulador se dá pela construção do bloco de função por meio do arquivo FCL. Criou-se esse arquivo com a utilização do software livre Notepad++, editor de texto e de código para estruturação do texto. A FIG. 4.6 apresenta a forma de como a declaração do bloco de função é construída e pode ser representada.

No arquivo FCL, que será tratado em detalhes na próxima subseção, estão especificados os parâmetros de entrada e saída, além do tipo de controle *fuzzy*, bloco de regras e declarações.

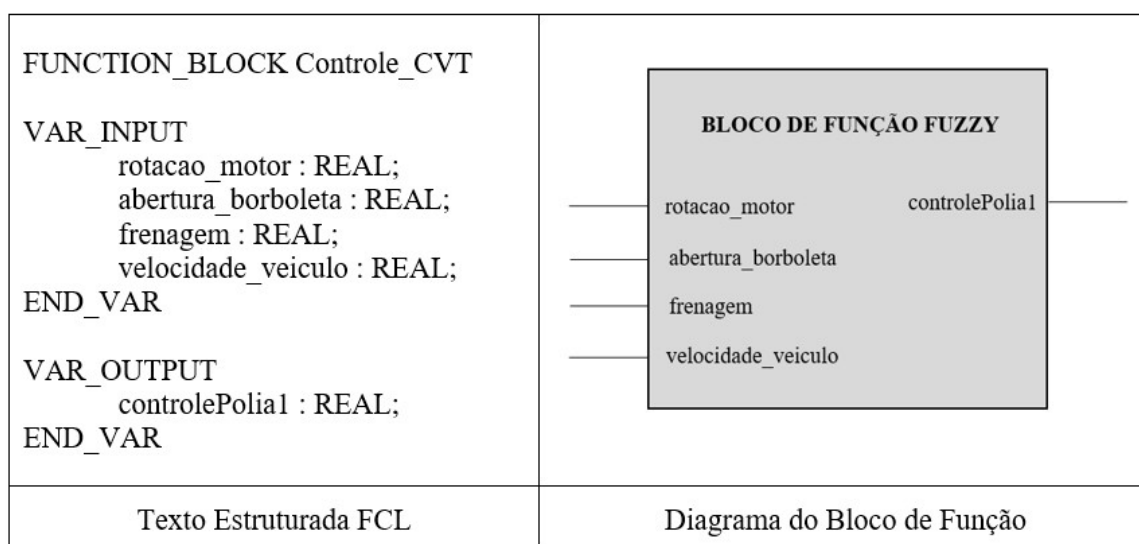
As variáveis de entrada possuem características de faixas de operação distintas que variam de acordo com suas especificações técnicas. As funções de pertinência dessas variáveis foram ajustadas para operar dentro de valores realistas para o domínio do problema.

### 4.3.2 Arquivo FCL

No arquivo FCL, no qual o bloco de funções, as variáveis e o conjunto de regras são criados, tem-se:

- a) Variáveis de entrada: rotação do motor, abertura da válvula borboleta (entendida como consumo de combustível), frenagem do veículo e velocidade do veículo;
- b) Variável de saída: controle de abertura e fechamento da polia primária da CVT;
- c) Conjunto de regras que relacionam as 4 variáveis de entrada entre si, juntamente com a variável de saída.

Figura 4.6: Exemplo da interface de declaração do Bloco de Função em linguagem estruturada de texto e em diagrama



Fonte: Adaptado de IEC611131-7 (2000)

Um trecho do bloco de função escrito e estruturado para o simulador conforme normas da linguagem de controle *fuzzy*, de acordo com a parte 7 da norma IEC 1131 (IEC611131-7, 2000) é apresentado da seguinte forma:

```
FUZZIFY rotacao_motor TERM minima := (0, 0) (0, 1) (900, 1) (3000, 0);
TERM media := (900, 0) (4750,1) (6500,0);
TERM maxima := (4750, 0)(6000, 1) (6500, 1);
END_FUZZIFY

FUZZIFY abertura_borboleta TERM pequena := (0, 0) (0, 1) (50, 0);
TERM media := (25, 0) (50, 1) (75, 0);
TERM alta := (50, 0) (90,1) (100, 1);
END_FUZZIFY

FUZZIFY frenagem TERM fraca := (0, 0) (0, 1) (5, 0);
TERM media := (2.5, 0) (5,1) (7.5,0);
TERM forte := (5, 0) (7.5, 1) (10,1);
END_FUZZIFY

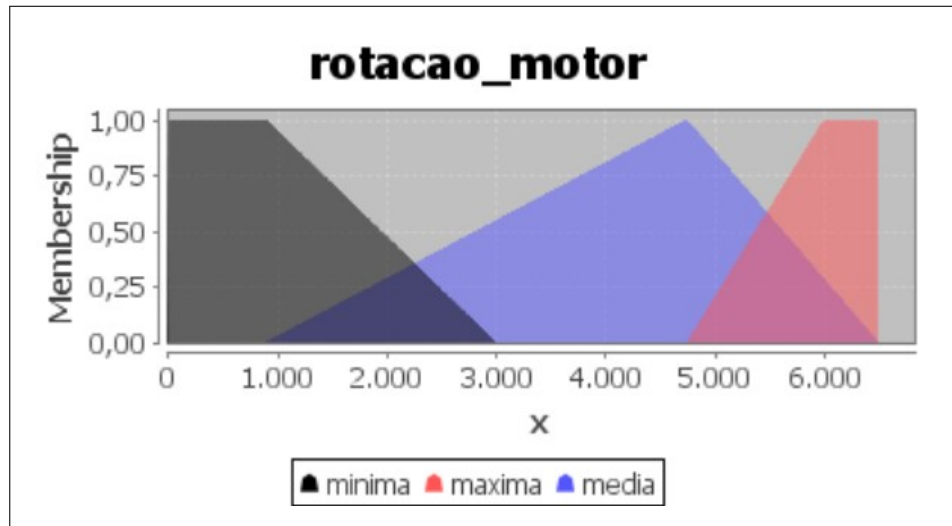
FUZZIFY velocidade_veiculo TERM baixa := (0, 0) (0, 1) (20, 1) (40, 0);
TERM media := (20, 0) (60,1) (80,0);
TERM alta := (60, 0) (90, 1) (180,1);
END_FUZZIFY

DEFUZZIFY controlePolia1 TERM reduzir := 0;
TERM aumentar := 100;
METHOD : COGS ;
DEFAULT := 0;
END_DEFUZZIFY

RULEBLOCK No1
RULE 1 : IF rotacao_motor IS minima THEN controlePolia1 IS reduzir;
RULE 2 : IF rotacao_motor IS media THEN controlePolia1 IS aumentar;
RULE 3 : IF rotacao_motor IS maxima THEN controlePolia1 IS aumentar;
RULE 4 : IF abertura_borboleta IS pequena THEN controlePolia1 IS reduzir;
RULE 5 : IF abertura_borboleta IS media THEN controlePolia1 IS aumentar;
RULE 6 : IF abertura_borboleta IS alta THEN controlePolia1 IS aumentar;
```

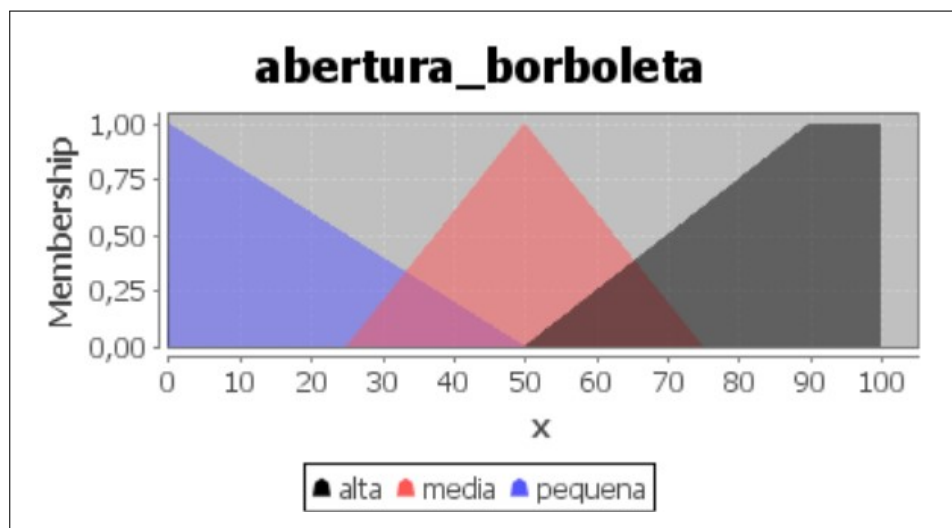
Quando o simulador é executado é possível apresentar os gráficos dos conjuntos difusos de cada variável linguística criada, inclusive o da variável de saída, que diferente das demais não possui faixa numérica e sim pontos únicos (*singletons*) mínimo e máximo. Como exemplos, a FIG. 4.7, a FIG. 4.8 e a FIG. 4.9 apresentam os gráficos para as funções de entrada rotação do motor, abertura da borboleta e o gráfico da variável de saída controle da polia primária, respectivamente.

Figura 4.7: Gráfico função de pertinência da variável de entrada rotação do motor



Fonte: Elaborado pelo autor a partir de Cingolani e Alcalá-Fdez (2013)

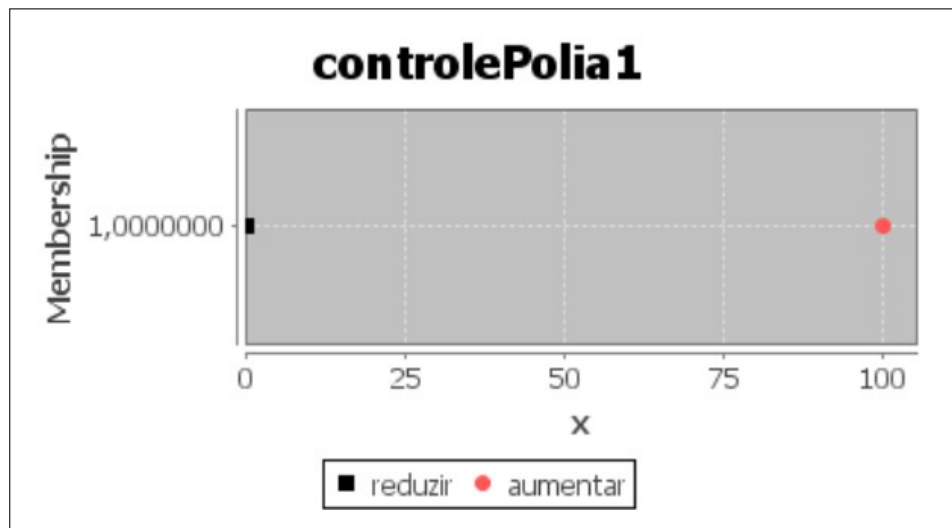
Figura 4.8: Gráfico função de pertinência da variável de entrada abertura da borboleta



Fonte: Elaborado pelo autor a partir de Cingolani e Alcalá-Fdez (2013)

No console, o resultado sobre o que o sistema infere, ou seja, em qual percentual de abertura a polia primária deve operar no intervalo  $[0, 100]$  é apresentado, por exemplo, da seguinte forma:

Figura 4.9: Gráfico função de pertinência da variável de saída controle da polia primária



Fonte: Elaborado pelo autor a partir de Cingolani e Alcalá-Fdez (2013)

T1 - Veículo Parado  
Rotação do Motor: 900.0  
Abertura da Borboleta: 0.0  
Frenagem: 10.0  
Velocidade do Veículo: 0.0  
Saída Controle ==> 0.0

Onde:

T1 - Veículo Parado: contexto em que o veículo se encontra;

Rotação do Motor: variável de entrada definida com valor de 900 RPM;

Abertura da Borboleta: variável de entrada definida com o valor 0. Portanto, não existe aceleração para esta situação;

Frenagem: variável de entrada definida com o valor 10. Portanto, a frenagem é máxima para esta situação;

Velocidade do Veículo: variável de entrada definida com o valor 0. Portanto, o veículo não está em movimento;

Saída Controle: variável de saída. Trata-se do resultado inferido pelo simulador. Este é o valor numérico gerado pelo sistema *fuzzy*. Para esta situação, sendo inferido o valor 0, o sistema emite o sinal de controle para que a abertura da polia primária se mantenha completamente fechada. Portanto, sua abertura será de 0%.

### 4.3.3 Processo de Fuzzificação

Cada variável de entrada possui sua função de pertinência em formato triangular, exceto a variável de saída (seção 4.2.4). Para cada função, definem-se os termos linguísticos, ou atributos, que a variável possui como características predefinidas para que seja “fuzzificada”.

Em que:

rotação do motor: mínima, média, máxima;

abertura da borboleta: pequena, média, alta;

frenagem: fraca, média, forte;

velocidade: baixa, média, alta;

controle da polia primária: reduzir, aumentar.

Os valores numéricos que são atribuídos a esses termos são convertidos em graus de pertinência na função de pertinência de cada variável. Essa conversão é descrita na FCL entre as palavras-chave FUZZIFY e END\_FUZZIFY (IEC611131-7, 2000)

```
FUZZIFY rotacao_motor
TERM minima := (0, 0) (0, 1) (900, 1) (3000, 0);
TERM media := (900, 0) (4750,1) (6500,0);
TERM maxima := (4750, 0)(6000, 1) (6500, 1);
END_FUZZIFY
```

### 4.3.4 Processo de Inferência

“Em lógica, inferência ou ilação é a operação por meio da qual se chega a uma conclusão ou se faz um raciocínio lógico com base em evidências circunstanciais e em conclusões já tidas como verdadeira.” (MICHAELIS, 2022). Conforme o próprio significado da palavra, o processo de inferência é a forma do sistema *fuzzy* operar para gerar uma proposição conforme afirmações ou informações interpretadas a partir do conjunto de regras criado.

No sistema *fuzzy* a inferência executa a base de regras conforme as seguintes subfunções:

1. Agregação: determina o grau de cumprimento de cada condição. Esse processo unifica a saída de todas as regras;
2. Ativação: ativa a conclusão IF-THEN para todas as regras considerando os fatores de ponderação;
3. Acumulação: é a combinação dos todos os resultados ponderados das regras em apenas um resultado geral.

O valor de saída do simulador para o controle, ou valor inferido pelo sistema, é gerado no processo de inferência. É nessa fase que os operadores lógicos IF, AND e THEN são utilizados para determinar as relações *fuzzy* que modelam a base de regras. Criou-se para a inferência um bloco com 111 regras para o controle, em que a estrutura de cada regra e as relações entre elas são descritas no arquivo FCL. O trecho abaixo apresenta as três primeiras regras do conjunto, que relacionam a variável rotação do motor com a variável de saída controle da polia primária

```
RULE 1 : IF rotacao_motor IS minima THEN controlePolia1 IS reduzir;  
RULE 2 : IF rotacao_motor IS media THEN controlePolia1 IS aumentar;  
RULE 3 : IF rotacao_motor IS maxima THEN controlePolia1 IS aumentar;
```

A regra 1 especifica que: se a rotação do motor for mínima, então o resultado para a variável de saída, controle da polia primária, será processado pelo FIS de forma que resulte em um valor em que o domínio de sua função pertença ao termo linguístico “reduzir”.

No caso específico desse projeto, tem-se variáveis de entrada que não podem ter relação entre si para a criação de regras. Por exemplo, se a frenagem atua, é prudente que o condutor não acione o pedal do acelerador. Portanto, não é necessário criar regras que associam a frenagem com a abertura da borboleta.

Quando o grau de pertinência for 0 para todos os termos linguísticos de uma variável de saída, significa que nenhuma regra para esta variável está ativa. Nesse caso é possível definir um valor *default*, pois a “defuzzificação” não será capaz de validar um resultado. Esse é um detalhe importante para a realização de testes e conseqüentemente aprimoramento das regras. Para o caso do simulador proposto, a variável de saída possui *singletons*. A faixa operação, ou domínio, varia de “reduzir”, cujo o valor mínimo é 0, e “aumentar”, cujo o valor máximo é 100. Portanto, é pouco provável ocorrer uma situação em que “reduzir” e “aumentar” sejam iguais a 0 e ter uma regra a ser ativada para esta condição. Motivo esse que não fará o valor *default* ser apresentado.

```
DEFAULT := 0;
```

Também é realizado no processo de inferência um mapeamento das funções de pertinência de todas as variáveis de entrada, o que determina quais as regras do bloco devem ser ativadas. Nesse processamento, são obtidos os conseqüentes e é realizada uma comparação entre os graus de pertinência dos termos das variáveis de entrada. Na seqüência, os conseqüentes serão convertidos em um valor real pelo método de “defuzzificação” COGS.

Diferentemente do método tradicional de inferência TKS, na FCL não é necessário criar a representação das regras em tabelas. O próprio sistema *fuzzy* de controle da linguagem se encarrega dessa tarefa. Na literatura consultada sobre a estrutura da lógica *fuzzy* e também sobre os métodos tradicionais, as fases do sistema *fuzzy* são bem definidas (fuzzificação, inferência e defuzzificação). O mesmo não ocorre na norma IEC611131-7 (2000) que, conforme descreve,



trata apenas de dois processos, a fuzzificação e a defuzzificação. Esses dois últimos englobam a inferência. Na FCL ela não é tratada de forma separada. Pode-se entender que o processo de inferência é propriamente a FCL, que contém os elementos para ilação.

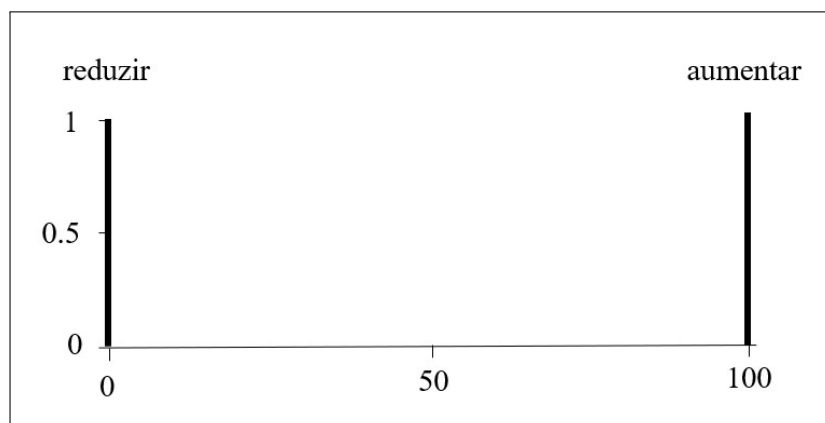
### 4.3.5 Processo de Defuzzificação

Uma variável linguística deve ser convertida em um valor. Tal valor é tratado como a variável de saída. Esta conversão ocorre entre as palavras-chave DEFUZZIFY e END\_DEFUZZIFY. O método de “defuzzificação” utilizado deve ser definido pelo elemento de linguagem METHOD.

```
DEFUZZIFY controlePolia1
TERM reduzir := 0;
TERM aumentar := 100;
METHOD : COGS ;
END_DEFUZZIFY
```

Conforme descrito na subseção 4.2.4, existe a possibilidade de aplicar diferentes métodos para se obter o valor da saída. Neste trabalho é utilizado o COGS (*Centre of Gravity for Singletons*). *Singletons* são funções de pertinência especiais usadas para saídas e que simplificam a “defuzzificação”. O valor para os termos linguísticos é descrito por apenas um único número. Para a “defuzzificação” o simulador dispõe da variável de saída, controle da polia primária, que gera o sinal para controlar a abertura ou fechamento das polias do CVT. Essa variável de saída possui *singletons* como função de pertinência definidos pelos termos reduzir e aumentar dentro do domínio de intervalo [0, 100], conforme apresentado na FIG.4.10.

Figura 4.10: Demonstra os valores *singletons* para a função de pertinência da variável de saída



Fonte: Adaptado de IEC611131-7 (2000)

---

# Testes e Resultados

---

Realizaram-se testes com o simulador por meio da definição específica de valores que compõem as funções de pertinência das variáveis de entrada. A etapa de testes do projeto permitiu avaliar os resultados que traduzem o comportamento da abertura e fechamento das polias da transmissão.

## 5.1 Avaliação dos Resultados

Oito diferentes situações de trajeto foram criadas com o intuito de observar o comportamento das polias:

Trajeto 1: Veículo parado;

Trajeto 2: Veículo Arrancando;

Trajeto 3: Veículo em alta velocidade reduzindo para parar ;

Trajeto 4: Veículo em velocidade média em aceleração para aumento da velocidade ;

Trajeto 5: Veículo em alta velocidade em aclave;

Trajeto 6: Veículo em alta velocidade em declive ;

Trajeto 7: Veículo em aclave acentuado;

Trajeto 8: Veículo em declive acentuado.

É grande o número de possibilidades para realizar testes nas condições em que o veículo possa se apresentar. Os oito trajetos criados permitiram avaliar situações gerais em estados básicos do automóvel. Tais condições oportunizam verificar uma resposta já esperada para

o comportamento de abertura ou fechamento das polias. Respostas essas que confirmam as premissas básicas de funcionamento da transmissão CVT. Por exemplo, quando a polia primária está com abertura tendendo a 0, demonstra que o motor do veículo está em rotação “media” ou “mínima”, e que o veículo está também em velocidade “media” para “baixa”, ou até mesmo parado.

Possíveis valores, dentro da faixa de domínio, podem ser aplicados para os termos das variáveis de entrada. Isso no intuito de obter um retorno de saída mais preciso referente ao percentual de abertura ou fechamento da polia primária. Desta forma, pode-se comprovar a eficiência do controle que as regras criadas para o bloco de funções têm sobre o resultado de inferência.

Para o trajeto 1, o trecho de código abaixo indica os valores que são definidos para gerar a saída de controle quando o veículo está parado. Conforme apresentado nos parâmetros, o comportamento sobre o percentual de abertura da polia primária é dedutível. É de se esperar que o resultado para a saída seja 0, isto é, a polia primária deve estar totalmente fechada. E esse é exatamente o resultado para o controle emitido pelo simulador. Essa parte do código define que a rotação do motor está a 900 RPM, portanto em repouso, a abertura da borboleta é nula, portanto não existe aceleração, a frenagem é máxima e a velocidade do veículo é 0 km/h.

```
t1.setNome("T1 - Veículo Parado");  
t1.setRotacaoMotor(900);  
t1.setAberturaBorboleta(0);  
t1.setFrenagem(10);  
t1.setVelocidadeVeiculo(0);
```

Resultado apresentado no console pelo simulador para o trajeto 1 :

```
T1 - Veículo Parado  
Rotação do Motor: 900.0  
Abertura da Borboleta: 0.0  
Frenagem: 10.0  
Velocidade do veículo: 0.0  
Saída Controle == => 0.0
```

O trajeto 2 apresenta a situação em que o veículo, anteriormente em estado de repouso, passa a ter uma aceleração média que reflete diretamente na rotação do motor e na velocidade, ambas ainda baixas, cujos valores fazem parte de suas funções de associação. Nessas condições, a saída de controle infere o valor de abertura da polia primária em 42.342%. Ou seja, ocorre a abertura da polia, que deixa a posição de torque máximo (totalmente fechada). O sistema “raciocina” que o carro, a 10 km/h, não necessita de torque máximo na condição de desenvolver aceleração.

Portanto, é sinalizado o valor para abertura diante das regras criadas para o controle.

Baseado em um modelo específico de automóvel com câmbio manual, Weihermann et al. (2015) criaram uma tabela que apresenta as velocidades, em km/h, desse mesmo veículo em cada marcha. Apenas com o intuito de relacionar uma determinada velocidade a uma marcha, não necessariamente no caso da CVT, utilizar-se-á essa tabela para relacionar a marcha com a velocidade em que o veículo se encontra. É possível calcular a velocidade em cada marcha de acordo com a rotação do motor. Logo, se o veículo em questão operasse com um câmbio manual e estivesse a 10 km/h e com a rotação do motor a 1.500 RPM, estaria na 1<sup>a</sup> marcha. No caso da CVT e para o simulador desse trabalho, a marcha é uma situação dinâmica e não estática. Tendo a CVT infinitas relações, o resultado para o câmbio operar nas condições que esse trajeto apresenta é com 42.342% de abertura da polia primária. Qualquer alteração em valores das variáveis que atuam no sistema irão alterar imediatamente o comportamento da abertura ou fechamento da polia primária e conseqüentemente a resposta para a movimentação do automóvel. Como não se troca a marcha e não se tem uma faixa de velocidade específica para um grau de abertura preestabelecido para a polia primária, pode ser que o valor de controle seja igual ou aproximado em uma condição totalmente diferente no que diz respeito aos valores definidos para as variáveis de entrada no simulador.

```
t2.setNome("T2 - Veículo Arrancando");
t2.setRotacaoMotor(1500);
t2.setAberturaBorboleta(40);
t2.setFrenagem(0);
t2.setVelocidadeVeiculo(10);
trajetos.add(t2);
```

Resultado apresentado no console pelo simulador para o trajeto 2:

```
T2 - Veículo Arrancando
Rotação do Motor: 1500.0
Abertura da Borboleta: 40.0
Frenagem: 0.0
Velocidade do veículo: 10.0
Saída Controle ==> 42.34269885239415
```

Na situação do trajeto 3, em que o veículo se encontra em alta velocidade e precisa reduzir, o acionamento do freio, definido em 8 bar, faz com que a polia primária se feche. O sistema infere o valor de controle 43.494%. Caso a frenagem não seja aplicada, ou seja, a condição de parar não exista, o simulador gera a saída de 100 para controle, ou seja, abertura máxima (100%) para essa condição.

```
t1.setNome("T3 - Veículo em alta velocidade reduzindo para parar");
t1.setRotacaoMotor(5500);
t1.setAberturaBorboleta(0);
t1.setFrenagem(8);
t1.setVelocidadeVeiculo(90);
```

Resultado apresentado no console pelo simulador para o trajeto 3:

```
T3 - Veículo em alta velocidade reduzindo para parar
Rotação do Motor: 5500.0
Abertura da Borboleta: 0.0
Frenagem: 8.0
Velocidade do veículo: 90.0
Saída Controle == => 43.49442379182156
```

No trajeto 4, o resultado para o controle é de certa forma intuitivo. O comportamento do veículo insinua que não deve haver resistência por parte da transmissão. A rotação do motor e a velocidade do veículo são médias, não existe frenagem e a aceleração é quase máxima. Portanto, deve haver torque, mas, nas devidas proporções e de acordo com uma avaliação para que não exista desperdício de energia na fonte e, conseqüentemente, haja economia de consumo na injeção do combustível.

```
t4.setNome("T4 - Veículo em velocidade média em aceleração para aumento da velocidade");
t4.setRotacaoMotor(4750);
t4.setAberturaBorboleta(90);
t4.setFrenagem(0);
t4.setVelocidadeVeiculo(60);
trajetos.add(t4);
```

Resultado apresentado no console pelo simulador para o trajeto 4:

```
T4 - Veículo em velocidade média em aceleração para aumento da velocidade
Rotação do Motor: 4750.0
Abertura da Borboleta: 90.0
Frenagem: 0.0
Velocidade do veículo: 60.0
Saída Controle ==> 85.71428571428571
```

Os resultados para as saídas de controle no trajeto 4 e trajeto 5 são bem próximos. Trata-se de uma interessante relação de controle para condições de estados bem parecidos do veículo. No trajeto 5, a aceleração é máxima e a velocidade é quase máxima. Os trajetos possuem uma pequena diferença entre seus valores para a rotação do motor. As saídas para o controle são quase as mesmas para ambos os trajetos. Não existe uma diferença relevante para a aceleração entre um trajeto e outro. A velocidade no trajeto 5 é mais que o dobro da velocidade no trajeto 4. É provável que exista uma relação maior para a saída de controle com a variável rotação do motor. Existe então a facilidade de consultar as regras criadas que relacionam essas variáveis e proceder a testes mais detalhados, no intuito de observar melhor esse comportamento para casos necessários.

```
t5.setNome("T5 - Veículo em alta velocidade em active");
t5.setRotacaoMotor(4500);
t5.setAberturaBorboleta(100);
t5.setFrenagem(0);
t5.setVelocidadeVeiculo(150);
trajetos.add(t5);
```

Resultado apresentado no console pelo simulador para o trajeto 5:

```
T5 - Veículo em alta velocidade em active
Rotação do Motor: 4500.0
Abertura da Borboleta: 100.0
Frenagem: 0.0
Velocidade do veículo: 150.0
Saída Controle ==> 85.16377649325626
```

O trajeto 6 apresenta uma condição extrema para forçar o resultado de saída que já é esperado. Diante dos valores definidos para as variáveis de entrada, é previsível que a saída de controle seja para abertura máxima da polia primária, e, portanto, 100%.

```
t6.setNome("T6 - Veículo em alta velocidade em declive");
t6.setRotacaoMotor(6500);
t6.setAberturaBorboleta(0);
t6.setFrenagem(0);
t6.setVelocidadeVeiculo(180);
trajetos.add(t6);
```

Resultado apresentado no console pelo simulador para o trajeto 6:

```
T6 - Veículo em alta velocidade em declive
Rotação do Motor: 6500.0
Abertura da Borboleta: 0.0
Frenagem: 0.0
Velocidade do veículo: 180.0
Saída Controle ==> 100.0
```

O trajeto 7 apresenta um cenário semelhante ao trajeto 2 em termos de valores definidos para as variáveis de entrada. A rotação do motor para ambos os trajetos são iguais. No entanto, no trajeto 7, o veículo está em aclave e ainda tenta ganhar velocidade, definida a abertura da borboleta em 90. Observa-se, então, uma relação entre a variável de saída e a variável de entrada abertura da borboleta. A saída de controle infere o percentual de abertura da polia primária em 47.094%, que sugere um maior trabalho de torque para o aumento da velocidade. Mesmo assim, a abertura da polia primária, no trajeto 2, é um pouco menor que no trajeto 7. Isso devido a relação que a variável de saída também tem com a variável de entrada velocidade do veículo, que, no trajeto 7, é um pouco maior.

```
t7.setNome("T7 - Veículo em aclave acentuado");
t7.setRotacaoMotor(1500);
t7.setAberturaBorboleta(90);
t7.setFrenagem(0);
t7.setVelocidadeVeiculo(20);
trajetos.add(t7);
```

Resultado apresentado no console pelo simulador para o trajeto 7:

```
T7 - Veículo em aclave acentuado
Rotação do Motor: 1500.0
Abertura da Borboleta: 90.0
Frenagem: 0.0
Velocidade do veículo: 20.0
Saída Controle ==> 47.09418837675351
```

Para o trajeto 8, com o mesmo propósito de simulação para o trajeto 6, aplicam-se características inversas na definição dos valores para as variáveis de entrada. Força-se, então um comportamento, também previsível, para a polia primária. No intuito de apurar um indicativo para o torque máximo ou próximo a isso, o simulador retorna o valor para a saída de controle com abertura da polia primária em 0%.

```
t8.setNome("T8 - Veículo em declive acentuado");
t8.setRotacaoMotor(900);
t8.setAberturaBorboleta(0);
t8.setFrenagem(6);
t8.setVelocidadeVeiculo(5);
trajetos.add(t8);
```

Resultado apresentado no console pelo simulador para o trajeto 8:

```
T8 - Veículo em declive acentuado
Rotação do Motor: 900.0
Abertura da Borboleta: 0.0
Frenagem: 6.0
Velocidade do veículo: 5.0 Saída Controle ==> 0.0
```

Na Tabela 5.1, são apresentados os dados para os valores dos termos das variáveis de entrada e também o resultado gerado (Saída Controle) referente aos oito trajetos criados para os testes no simulador. Esta tabela agrupa os dados permitindo avaliar de forma prática o comportamento do simulador conforme os dados de entrada.

O simulador também produz os gráficos das resultantes de controle para todos os trajetos. A FIG. 5.1, como exemplo apresenta o gráfico dessa variável de saída para o trajeto 3. Como a função de pertinência para a variável de saída é dada por *singletons*, podem-se observar estes pontos na função cujo valor resultante de 43.494% foi gerado pelo método de “defuzzificação” COGS do sistema *fuzzy* (subseção 4.2.4).

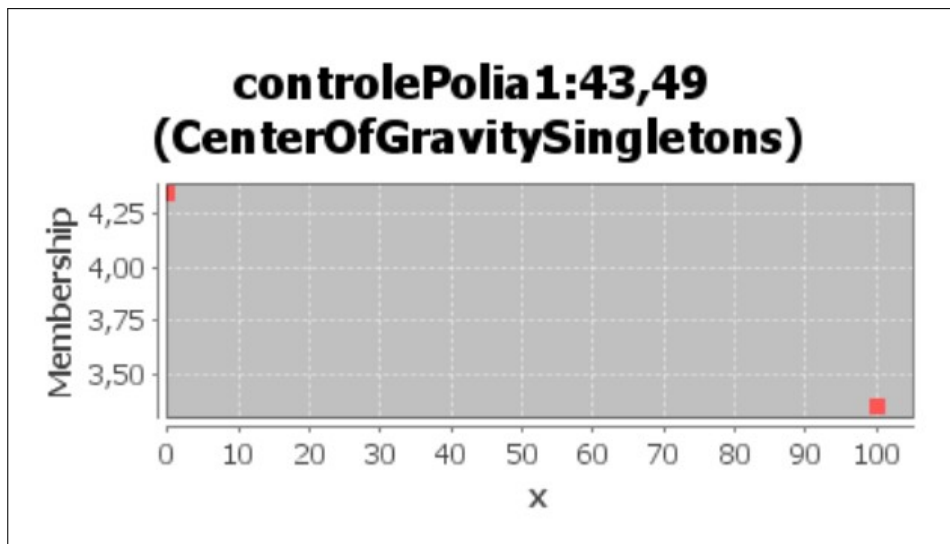


Tabela 5.1: Resultados das saídas para o controle em cada trajeto

Trajeto	Rotação Motor	Abertura Borboleta	Frenagem	Velo. Veículo	Saída Controle
1	900.00	0.00	10.00	0.00	0.000
2	1500.00	40.00	0.00	10.00	42.342
3	5500.00	0.00	8.00	90.00	43.494
4	4750.00	90.00	0.00	60.00	85.714
5	6500.00	100.00	0.00	150.00	85.163
6	6500.00	0.00	0.00	180.00	100.000
7	1500.00	90.00	0.00	20.00	47.094
8	900.00	0.00	6.00	5.00	0.000

Fonte: Autoria própria

Figura 5.1: Resultado para a saída no trajeto 3



Fonte: Elaborado pelo autor a partir de Cingolani e Alcalá-Fdez (2013)

---

## Conclusões

---

Um dos maiores desafios deste trabalho foi relacionar o conjunto de variáveis e seus atributos. A ferramenta jFuzzyLogic tanto como o sistema FIS para este tipo de aplicação, que permite converter cálculos complexos em combinações de termos linguísticos, é de grande utilidade. No entanto, é necessário conhecimento técnico especializado e um estudo minucioso sobre o relacionamento que uma variável possui com alguma outra específica ou com um conjunto de variáveis diversas. A partir desse estudo detalhado, testes devem ser feitos um a um e seus resultados avaliados, tendo em mente os consequentes e também os efeitos colaterais que a produção de uma regra pode criar em uma outra.

Inicialmente, o simulador foi implementado para gerar valores aleatórios para os termos dentro de suas faixas de operação. Tal objetivo foi alterado a fim de criar testes que pudessem ser analisados com mais previsão e precisão. Para isso, os valores deixaram de ser aleatórios e passaram a ser definidos conforme pretensão.

### 6.1 Considerações Finais

A criação do conjunto de regras envolve um cruzamento de informações que relacionam às variáveis presentes dentro do domínio do problema entre si. Quanto mais detalhado for este cruzamento e a comunicação entre os atributos destas variáveis, que geram os conjuntos difusos, aumentam-se as possibilidades de se obter um resultado mais preciso para o valor objetivo da saída.

### 6.2 Contribuições

São numerosos os trabalhos realizados sobre controladores *fuzzy* e suas aplicações em diversas áreas e propósitos. Estudos com controladores aplicados em transmissões automáticas, e dados mecânicos relacionados à engenharia de um motor ou sobre o funcionamento de um câmbio, são

de difícil obtenção. Este trabalho contribui no sentido de facilitar a compreensão da aplicação e operação da lógica *fuzzy* em uma transmissão automática, apresentar uma ferramenta que favorece a modelagem do sistema *fuzzy*, e colabora para trabalhos futuros baseados na intensa pesquisa sobre um assunto que é pouco difundido entre acadêmicos.

### 6.3 Trabalhos Futuros

O resultado obtido com o simulador proposto foi baseado na criação de regras produzidas por um humano não especialista. Por mais *expertise* que um especialista possua, erros podem ocorrer no desenvolvimento do conjunto de regras. A criação de um sistema para geração automática de regras é interessante para o propósito desse trabalho ou para outros similares. Estudos para o desenvolvimento de algoritmos para esse fim já são encontrados na literatura (ARRUDA et al., 2019) (MOHAMMADIAN; STONIER, 1994). Trata-se da geração automática de regras *fuzzy* utilizando um algoritmo genético. Este tipo de processo pode ser muito útil para evitar falhas, cercar o maior número de possibilidades e promover resultados mais precisos.

No simulador foram propostos trajetos básicos para possíveis situações do veículo com o intuito de cumprir os objetivos específicos apresentados. Não foram consideradas variações ou parâmetros de entrada para situações mais complexas conforme o que Danczkay e Alves (2015) diz:

Quando um veículo encontra uma perturbação, correspondente à variação da carga na roda ou na solicitação de maior velocidade, ele tem sua velocidade real diferente da velocidade desejada. Esta alteração demanda na alteração de potência. Isto é conseguido pela variação na injeção de combustível.

Tais condições podem ser incluídas em trabalhos futuros.

## Referências Bibliográficas

---

- ADACHI, M. et al. *Method and system using fuzzy logic for controlling a CVT transmission*. [S.l.]: Google Patents, 1996. US Patent 5,558,596.
- ADEKZTI. *Vantagens da programação em Java*. 2022. Último acesso em 22 de julho de 2022. Disponível em: <[www.adekz.com/post/vantagens-da-programacao-em-java](http://www.adekz.com/post/vantagens-da-programacao-em-java)>.
- ARRUDA, D. M. et al. Análise comparativa de ferramentas computacionais para modelagem de lógica fuzzy. *X Simpósio de Excelência em Gestão e Tecnologia, Resende*, 2019.
- CALDAS, L.; LEONARDI, F. Controle fuzzy. 1999.
- CINGOLANI, P.; ALCALÁ-FDEZ, J. jfuzzylogic: a java library to design fuzzy logic controllers according to the standard for fuzzy control programming. p. 61–75, 2013.
- COSTA, S. R. F. d. O. *Controlo do tipo Takagi-Sugeno Difuso Robusto em sistemas não lineares não afins*. Tese (Doutorado) — Universidade Federal de Goiás, 2018.
- DANCZKAY, G. Z.; ÁLVES, M. A. L. Proposta de controlador fuzzy para transmissão continuamente variável. 2015.
- FELIX, L. *Carro automático salta de 12em 10 anos*. 2021. Último acesso em 06 de julho de 2021. Disponível em: <[www.quatrorodas.abril.com.br/noticias/carro-automatico-salta-de-12-para-50-do-mercado-brasileiro-em-10-anos/](http://www.quatrorodas.abril.com.br/noticias/carro-automatico-salta-de-12-para-50-do-mercado-brasileiro-em-10-anos/)>.
- FERNANDES, M. A. Controlador fuzzy aplicado a veículos elétricos com transmissão variável contínua. 2014.
- IEC611131-7, I. E. C. *International Standard*. 2000. Último acesso em 01 de abril de 2022. Disponível em: <[www.rpa.energy.mn/wp-content/uploads/2017/03/IEC-61131-7-Programmable-controllers-Fuzzy-control-programming.pdf](http://www.rpa.energy.mn/wp-content/uploads/2017/03/IEC-61131-7-Programmable-controllers-Fuzzy-control-programming.pdf)>.
- KIA, B. *O que é transmissão automática? Quais as suas vantagens?* 2022. Último acesso em 02 de abril de 2022. Disponível em: <[www.kia.com.br/kianews/o-que-e-transmissao-automatica-quais-as-suas-vantagens](http://www.kia.com.br/kianews/o-que-e-transmissao-automatica-quais-as-suas-vantagens)>.
- LEGASPE, E. P. *Controlador fuzzy de código aberto para uso em controladores programáveis*. Tese (Doutorado) — Universidade de São Paulo, 2012.
- MACHADO, E. R. M. D. Modelagem e controle de sistemas fuzzy takagi-sugeno. Universidade Estadual Paulista (UNESP), 2003.

- MICHAELIS. *Inferência*. 2022. Último acesso em 2a de agosto de 2022. Disponível em: <[www.michaelis.uol.com.br/busca?id=0LGao](http://www.michaelis.uol.com.br/busca?id=0LGao)>.
- MICHEL, G. *Programmable logic controllers: architecture and applications*. [S.l.]: John Wiley & Sons, Inc., 2007.
- MOHAMMADIAN, M.; STONIER, R. Generating fuzzy rules by genetic algorithms. In: *Proceedings of 1994 3rd IEEE International Workshop on Robot and Human Communication*. [S.l.: s.n.], 1994. p. 362–367.
- MORAES, C. C. d.; CASTRUCCI, P. d. L. *Metodologia de pesquisa para ciência da computação*. [S.l.]: Grupo GEN, 2006.
- OLIVEIRA, D. d. *Alta demanda por câmbios automáticos faz os manuais sumirem das concessionárias*. 2021. Último acesso em 06 de julho de 2021. Disponível em: <[www.jornaldocarro.estadao.com.br/carros/demanda-por-cambios-automaticos-segue-alta-mas-sem-producao-local/](http://www.jornaldocarro.estadao.com.br/carros/demanda-por-cambios-automaticos-segue-alta-mas-sem-producao-local/)>.
- PORTAL, S. F. *Transmissão CVT*. 2022. Último acesso em 15 de fevereiro de 2022. Disponível em: <[www.portalsaofrancisco.com.br/mecanica/transmissao-cvt](http://www.portalsaofrancisco.com.br/mecanica/transmissao-cvt)>.
- RAGASSI EDSON; LALLI, F. *Transmissão: Como funciona o câmbio CVT?* 2021. Último acesso em 07 de julho de 2021. Disponível em: <[www.omecanico.com.br/transmissao-como-funciona-o-cambio-cvt/](http://www.omecanico.com.br/transmissao-como-funciona-o-cambio-cvt/)>.
- RIGNEL, D. G. d. S. e. a. *Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica*. 2021. Último acesso em 07 de julho de 2021. Disponível em: <[www.logicafuzzy.com.br/wp-content/uploads/2013/04/uma\\_introducao\\_a\\_logica\\_fuzzy.pdf](http://www.logicafuzzy.com.br/wp-content/uploads/2013/04/uma_introducao_a_logica_fuzzy.pdf)>.
- SEBESTA, R. W. *Conceitos de Linguagens de Programação-11*. [S.l.]: Bookman Editora, 2018.
- SETLUR, P. et al. Nonlinear control of a continuously variable transmission (cvt). *IEEE Transactions on control systems technology*, IEEE, v. 11, n. 1, p. 101–108, 2003.
- SILVA, R. A. C. *Inteligência Artificial Aplicada a Ambientes de Engenharia de Software: Uma Visão Geral*. 2021. Último acesso em 07 de julho de 2021. Disponível em: <[www.infocomp.dcc.ufla.br/index.php/infocomp/article/view/109](http://www.infocomp.dcc.ufla.br/index.php/infocomp/article/view/109)>.
- SIMÕES, M. G.; SHAW, I. S. *Controle e modelagem fuzzy*. Porto Alegre: Editora Blucher, 2007.
- SOUJAVA. *Sociedade de Usuários da Tecnologia Java*. 2022. Último acesso em 22 de julho de 2022. Disponível em: <[www.soujava.org.br](http://www.soujava.org.br)>.
- TAKAGI, T.; SUGENO, M. Fuzzy identification of systems and its applications to modeling and control. *IEEE transactions on systems, man, and cybernetics*, IEEE, n. 1, p. 116–132, 1985.
- UOL, C. *Qual a rotação ideal para economizar combustível?* 2022. Último acesso em 25 de julho de 2022. Disponível em: <[www.uol.com.br/carros/faq/como-economizar-combustivel-marchas-rotacao-ideal-e-outras-duvidas.htm](http://www.uol.com.br/carros/faq/como-economizar-combustivel-marchas-rotacao-ideal-e-outras-duvidas.htm)>.
- WEIHERMANN, H. W. et al. *Estudo sobre aplicação de transmissão continuamente variável para veículos de pequeno porte*. 2015.

WILHELM, V. *TP034-Tópicos Especiais de Pesquisa Operacional I, Conjuntos Difusos ? Inferência*. 2000. Último acesso em 08 de abril de 2022. Disponível em: <[www.docs.ufpr.br/~volmir/Fuzzy\\_7.pdf](http://www.docs.ufpr.br/~volmir/Fuzzy_7.pdf)>.

ZANCAN, M. D. *Controladores Programáveis*. 2011. Último acesso em 01 de abril de 2022. Disponível em: <[www.proedu.rnp.br/bitstream/handle/123456789/319-/controladores\\_programaveis.pdf](http://www.proedu.rnp.br/bitstream/handle/123456789/319-/controladores_programaveis.pdf)>.