ENGENHARIA DE COMPUTAÇÃO - Trabalho de Conclusão de Curso - 2017/01

CONTROLADOR DE TEMPERATURA UTILIZANDO O MÓDULO FUZZY DO RSLOGIX 5000

Alex de Oliveira Graciano (Orientador Edilberto Pereira Teixeira)
Universidade de Uberaba
alex.o.graciano@gmail.com; edilberto.teixeira@gmail.com

Resumo: O controle de temperatura tanto em ambientes residenciais como na indústria se torna importante devido aos gastos com energia, a interferência direta em processos industriais o que leva a necessidade de ambientes confortáveis e controlados para processos e pessoas. Por meio da *Fuzzy Logic* é possível projetar um controlador de temperatura usando como *hardware* um Controlador Lógico Programável (CLP) programado usando a linguagem de programação *Ladder.* Essa implementação se dá pelo módulo *fuzzy* do software RSLogix 5000 que é um software para programação de CLP's e tem seu módulo *fuzzy* que permite toda a modelagem *fuzzy*. Neste trabalho estuda-se e projeta por meio de simulações em máquina virtual um controlador de temperatura sem a necessidade da modelagem matemática que o algoritmo PID (Proporcional, Integrador, Derivativo) necessita. No módulo *fuzzy* do RSLogix 5000 é desenvolvido a modelagem com as variáveis linguísticas de entrada, a base de regras do processo e a variável linguística de saída. Posteriormente foi desenvolvido a lógica *Ladder* de controle das ações da modelagem *fuzzy* que trabalha com base nas regras do controlador nebuloso. O controlador foi submetido a testes em máquina virtual e apresentou um comportamento em sua saída dentro do esperado se mostrando eficiente e de fácil implementação.

Palavras-chave: *Fuzzy Logic*, Controlador Lógico Programável, Controle Inteligente de Temperatura.

1 Introdução

O avanço tecnológico do século passado se deu em partes pela globalização que consigo trouxe inúmeras mudanças principalmente na indústria. A partir desse momento surgiu a automação de processos fabris que anos mais tarde se estabeleceu também nas residências.

Com a automação surgiram inúmeras técnicas de controle de processos, e uma das mais eficientes e mais utilizadas é o PID (Proporcional, Integral, Derivativo). Porém o controle PID incorporado a processos não lineares, processos com incertezas, se torna muito caro e complexo, e uma das variáveis mais importantes é a temperatura, quando aplicado seu controle em ambientes ela se torna difusa, o que leva a necessidade da utilização de outras técnicas mais eficientes.

A temperatura do ambiente interfere no funcionamento de equipamentos e processos no chão de fábrica como também é importante no ambiente

residencial quando promove conforto de forma eficiente e inteligente.

Em vista da complexidade do PID no controle de temperatura, é possível implementar a *fuzzy logic* que é de fácil implementação e é economicamente mais viável desde que se tenha conhecimento do processo de controle de temperatura.

Este trabalho visa projetar um controlador de temperatura ambiente baseando-se nos princípios da técnica de inteligência artificial denominada *fuzzy logic aplicados ao módulo fuzzy do RSLogix 5000*.

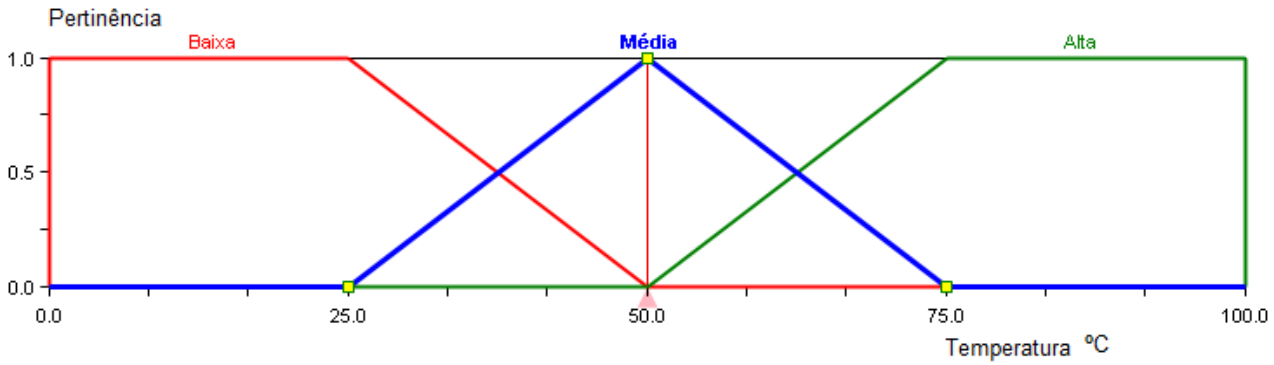
1.1 Fuzzy Logic

A *fuzzy logic* pertence ao grupo básico de controle utilizando Inteligência Artificial (IA). Seu emprego em sistemas com técnicas de IA está cada vez mais presente na indústria e na automação residencial devido a sua fácil implementação e custo de projeto. O desejo e os interesses econômicos e estratégicos de fazer com que as máquinas tomem decisões inteligentes tem motivado o desenvolvimento de várias vertentes de Inteligência Artificial (TEIXEIRA, 2015, P.3).

A *fuzzy logic* nasceu em 1965, através da teoria dos conjuntos nebulosos com o trabalho de Lotfi A. Zadeh onde os princípios base de seu estudo foi baseado na teoria dos conjuntos matemáticos, porém com a diferença que na teoria clássica dos conjuntos um elemento só pertence a um conjunto. “Já nas bases da teoria dos conjuntos nebulosos um elemento pode pertencer a mais de um conjunto” (Zadeh, 1960), o que leva a *fuzzy logic* ser eficiente no tratamento de processos que em seu controle envolva ambiguidade dos elementos trabalhados.

Uma forma de representar elementos variáveis (temperatura), é de forma gráfica, por se tratar de uma variável linguística de um conjunto de entrada de um sistema *fuzzy* (Figura 1).

Figura 1: Variável linguística de entrada



Fonte: Costa (2007), adaptada do próprio autor

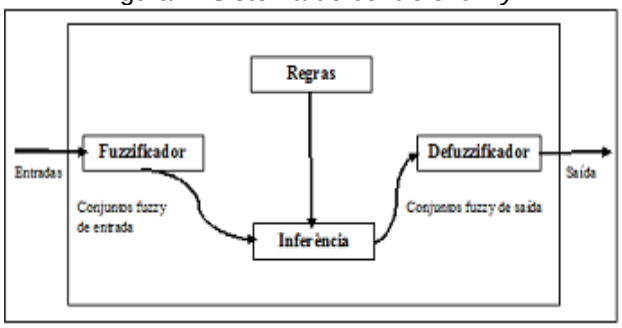
Na Figura 1, a temperatura é representada no eixo X e três funções (baixa, média e alta) representam a classificação conforme a leitura em ºC. A esse valor de leitura é atribuído um grau de pertinência (o grau de pertinência vai de 0 a 1 como é visto no eixo Y da Figura 1, e uma vez que a temperatura esteja em 30 ºC com uma pertinência de 0,8 esse valor pertence ao conjunto média de temperatura) que define quanto é possível para esse elemento temperatura pertencer a uma dessas três funções.

1.2 Sistemas de Controle Fuzzy

A combinação de elementos de entrada e saída que são definidos por variáveis linguísticas e estão integrados por uma base de regras sendo definida como sistema de controle *fuzzy.*

Um controle fuzzy não necessita da modelagem do processo, com todos os cálculos matemáticos, e sim, da compreensão das ações do processo. Compreensão essa que é modelada com base no conhecimento do funcionamento do processo. Isso torna o controle fuzzy diferente dos controles convencionais que usam algoritmos matemáticos baseando seu funcionamento em ações de derivação para correção de erros no controle. Na Figura 2 segue a estrutura de um sistema de controle fuzzy.

Figura 2: Sistema de controle *fuzzy*



Fonte: Tarig, 2001, adaptada do próprio autor

Um controlador *fuzzy* consegue analisar apenas variáveis *fuzzy*, ou seja, as informações de entrada que são lidas por sensores devem ser transformadas em conjuntos *fuzzy* pelo *fuzzyficador*.

A “fuzzificação” trabalha recebendo os valores de entrada do controlador (vindo da leitura de todos os sensores), após isso é feito um escalonamento que irá dimensionar os valores a universos de discursos normalizados e “fuzzifica” os valores, ou seja, transformando números em conjuntos *fuzzy* para torná-los instâncias de variáveis linguísticas (Tarig, 2001, p.37).

Segundo Tarig (2001, p.38), A base de conhecimento ou Base de Regras como também é chamada se caracteriza por um conjunto de regras que ditam estratégias, ações de controle a serem tomadas. Essa Base de Regras armazena informações importantes, pertinentes ao processo sobre as discretizações, as normalizações dos universos de discurso *fuzzy* dos espações de entrada e saída e também as funções de pertinência do controlador.

A inferência é um processo que atua diretamente sobre os dados de entrada do controlador *fuzzy,* juntamente com as regras, para inferir as devidas ações de controle, sendo isso feito usando o operador de implicação *fuzzy* e as regras de inferência da *Fuzzy Logic*. (Tarig, 2001, p.38).

A “defuzzificação” é um processo de inferência que atua sobre as ações de controle *fuzzy* transformando essas ações em ações de controle não *fuzzy* para a saída do controlador, com isso é feito um escalonamento para compatibilizar os valores normalizados vindos do passo de inferência da base de regras com seus respectivos valores dos universos de discursos reais das variáveis (Tarig, 2001, p.38).

Essas características são essenciais para o controle de temperatura ambiente que em seu modelo apresenta leituras ambíguas de temperatura dentro de uma faixa de trabalho tornando o sistema difuso.

1.3 Controladores Lógicos Programáveis

Os Controladores Lógicos Programáveis (CLP’s) são equipamentos eletrônicos muito utilizados nas industrias para controle de processos em sistemas

de automação com algumas aplicações residenciais, hospitalares, industriais, etc.

O Controlador Lógico Programável nasceu dentro da General Motors pelo engenheiro Richard Morley tendo como vantagens em relação aos equipamentos de controle da época:

- Menor espaço de utilização;
- Menor consumo de energia;
- Reutilizáveis;
- Programáveis;
- Maior confiabilidade;
- Maior flexibilidade;
- Maior rapidez na elaboração dos projetos;
- Interface de comunicação com outros CLP's e computadores;

1.3.1 Princípios de funcionamento

Os CLP's em sua estrutura básica trabalham em três etapas sendo as entradas, a unidade central de processamento (CPU) e as saídas.

Os sinais de entrada e saída podem ser analógicos ou digitais compostos de grupos de bits, associados em conjunto de 8 bits (1 Byte) conforme o tipo de CPU.

As entradas quando analógicas são módulos conversores A/D (Analógico para Digital) e as saídas analógicas são módulos conversores D/A (Digital para Analógico).

Os sinais dos sensores são aplicados às entradas do controlador como no caso da temperatura sendo a mesma uma leitura analógica e a cada ciclo (varredura) do controlador todos os valores são lidos e enviados para a CPU que são associados entre si e aos sinais internos conforme a programação carregada no CLP, e após o processamento as informações resultantes são enviadas a saída.

A programação que é carregada no CLP é desenvolvida em uma linguagem de programação conforme fabricante do CLP utilizado no projeto sendo que as linguagens mais utilizadas são a Lógica *Ladder, Function Block Diagram* (FBD), *Structured Text* (ST) e *Sequential Function Chart* (SFC).

1.4 Lógica Ladder

A linguagem *Ladder (em inglês quer dizer escada)* é baseada nos princípios de comandos elétricos ou contatos elétricos e foi a primeira linguagem desenvolvida para programação de CLP's.

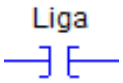
Ela associa os contatos em uma representação gráfica de forma horizontal, em linhas paralelas, que lembram os degraus de uma escada usada para representar esquemas elétricos. Apesar de ter sido a primeira linguagem desenvolvida para CLP's, ainda hoje é uma das mais utilizadas por engenheiros e técnicos da área de automação.

A linguagem *Ladder* que também é conhecida como Diagrama *Ladder* tem como recursos de programação elementos de contato normalmente aberto (NA), elementos de contato normalmente fechado (NF), elementos de bobina, instruções de temporização, elementos contadores, e elementos que representam funções lógica e matemáticas sendo esses últimos utilizados no projeto para programar o controlador em *Ladder*.

1.4.1 Elemento de contato normalmente aberto

A instrução de contato normalmente aberto (NA) determina se um bit está ativado, valor 1, sendo verdadeira essa condição a lógica de instrução será avaliada como verdadeira passando para o próximo passo da estrutura de programação. A Figura 3 ilustra o contato normalmente aberto.

Figura 3: Contato normalmente aberto

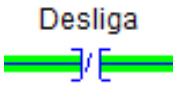


Fonte: Próprio autor

1.4.2 Elemento de contato normalmente fechado

A instrução de contato normalmente fechado (NF) determina se um bit está desativado, valor 0, sendo verdadeira essa condição a lógica de instrução será avaliada como verdadeira passando para o próximo passo da estrutura de programação. A Figura 4 ilustra o contato normalmente fechado.

Figura 4: Contato normalmente fechado

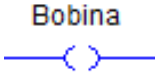


Fonte: Próprio autor

1.4.3 Elementos de bobina

O elemento de bobina recebe valor 1 podendo trabalhar como elemento de saída para acionar um motor ou como uma variável interna do programa. A Figura 5 ilustra o elemento bobina.

Figura 5: Elemento de saída ou bobina.

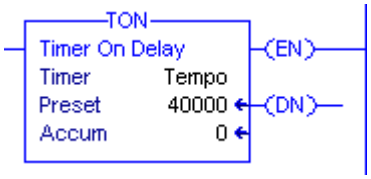


Fonte: Próprio autor

1.4.4 Temporizadores

Os temporizadores são utilizados quando se pretende associar um atraso no acionamento de um dispositivo ou em seu desligamento. O temporizador começa sua contagem quando a lógica de *rung* passa de falsa para verdadeira. O *Preset* serve como variável de referência para a contagem e o *Accum* acumula o valor de contagem comparando com o *Preset*, ao estourar esse valor a saída *Done* (DN) é acionada. A Figura 6 ilustra um exemplo de temporizador.

Figura 6: Temporizador do tipo Timer On Delay.

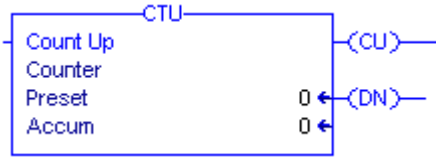


Fonte: Próprio autor

1.4.5 Contadores

Os contadores são utilizados para contar as mudanças de falso para verdadeiro na lógica de *rungs*. As transições de falso para verdadeiro decorrem de elementos presentes na lógica interna como também de elementos externos associados ao controle. O Contador possui um *Preset* que serve como referência para a contagem e um *Accum* que acumula o valor de contagem comparando com o *Preset*, ao estourar esse valor a saída *Done* (DN) é acionada. A Figura 7 ilustra um exemplo de contador.

Figura 7: Contador do tipo Counter Up

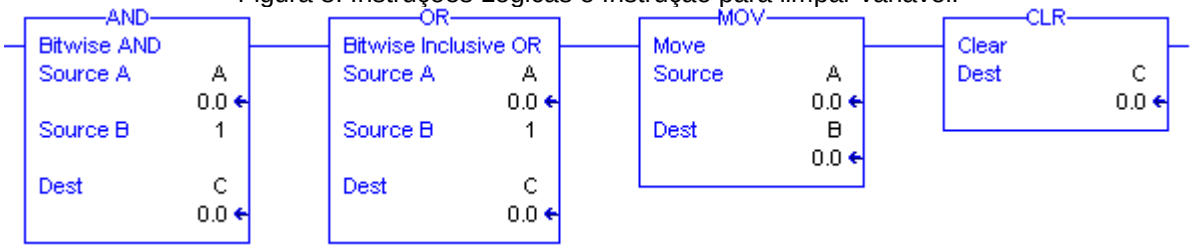


Fonte: Próprio autor

1.4.6 Funções matemáticas e Lógicas em Ladder

A linguagem *Ladder* traz opções de operadores matemáticos, relacionais e lógicos em sua programação. Os operadores lógicos trabalham como a lógica digital onde as instruções simulam o funcionamento de uma porta lógica como a AND, OR, e também notamos instruções de movimentação de um determinado valor para um destino, na Figura 8 podemos notar um exemplo dessas instruções onde o *Source* A necessariamente deve ser uma Tag e o *Source* B pode ser uma *Tag* ou valor *booleano*.

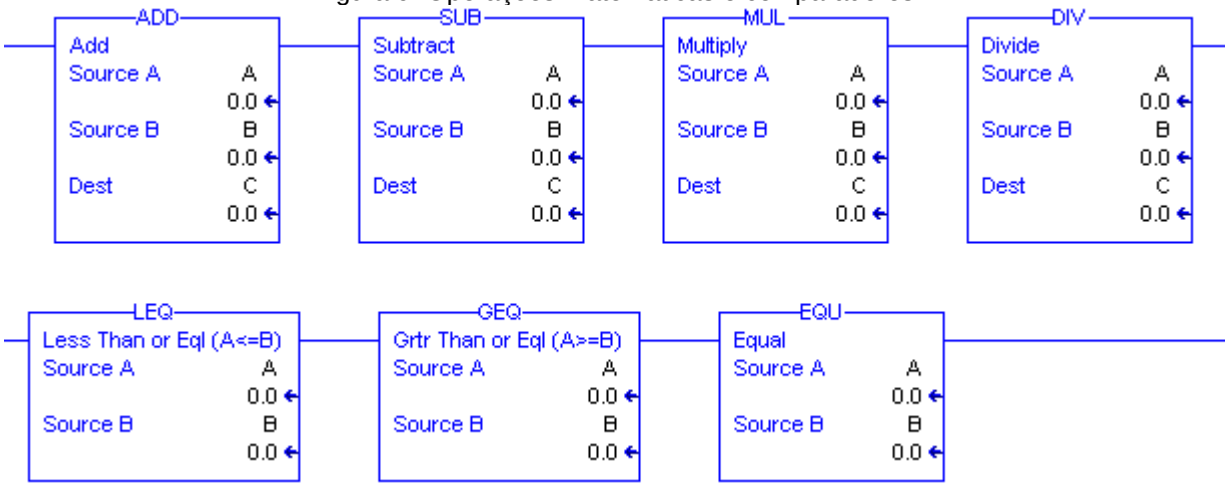
Figura 8: Instruções Lógicas e Instrução para limpar variável.



Fonte: Próprio Autor

As funções matemáticas disponíveis são de soma, subtração, multiplicação, divisão e há também instruções de comparação como podemos notar na Figura 9.

Figura 9: Operações matemáticas e comparadores



Fonte: Próprio Autor

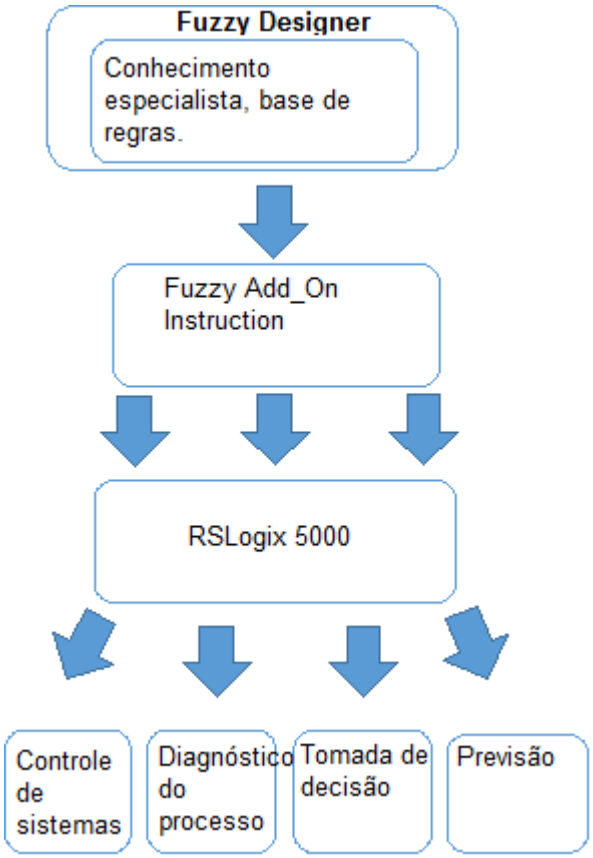
As instruções de comparação, comparam um valor de uma *Tag* no *Source A* com o valor de uma *Tag* no *Source B*.

1.5 Módulo Fuzzy RSLogix 5000

O *RSLogix* conta com um módulo para desenvolvimento de controladores *Fuzzy, o RSLogix 5000 Fuzzy Designer,* utilizado para a modelagem do processo a ser controlado. Esse software trabalha na

seguinte estrutura como é mostrado na hierarquia da Figura 10.

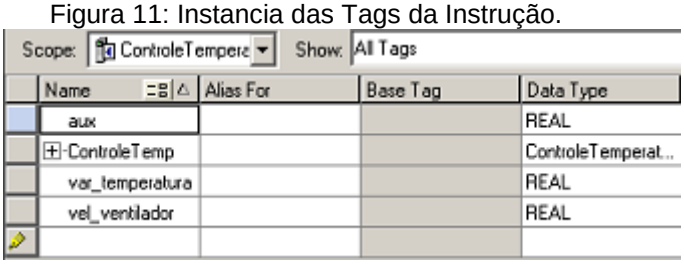
Figura 10: Funcionamento do Fuzzy Designer



Fonte: Rockwell Automation, Adaptada do próprio autor

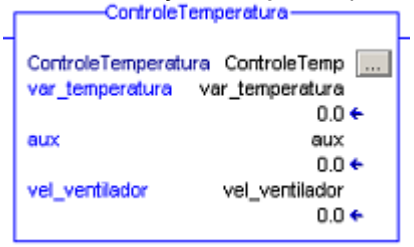
No *Fuzzy Designer* é construído o designer do controlador *fuzzy*, com as variáveis linguísticas de entrada e saída, e a base de regras. Após a modelagem é necessário gerar uma instrução *Fuzzy Add_On Instruction* no próprio *Fuzzy Designer,* que gera um arquivo com extensão L5X, esse arquivo deve ser importado para o *RSLogix* 5000. Após os procedimentos citados é necessário criar a instancia das *Tags* da *Add_On Instruction, Tags essas que se referem as variáveis linguísticas de entrada e saída do controlador, após criar suas instancias é necessário* encapsular o sistema de inferência com as duas variáveis de entrada e a saída centroide como mostra as Figuras 11 e respectivamente.

Figura 11: Instancia das Tags da Instrução.



Fonte: Próprio Autor

Figura 12: Instrução *Fuzzy Encapsulada*



Fonte: Próprio Autor

Após os procedimentos já citados e desenvolvido a lógica de controle no CLP é feito a comunicação do *RSLogix* 5000 com o *Fuzzy Designer*, essa comunicação é feito pelo protocolo de comunicação OPC *Server*. Estando tudo configurado corretamente a comunicação é feita deixando o Controlador Online onde é possível monitorar o funcionamento do mesmo através do próprio *Fuzzy Designer* ou por meio de um sistema supervisório.

2 Materiais e Métodos

O trabalho foi desenvolvido seguindo os preceitos do estudo exploratório e simulação em máquina virtual, por meio de pesquisas, que, segundo GIL (2008, p.50), “é desenvolvida a partir de material já elaborada como livros, artigos eletrônicos, revistas, etc.”

Em um primeiro momento foi realizado um estudo da aplicação da *Fuzzy Logic*

no controle de temperatura de modo geral.

Após o estudo geral da *Fuzzy Logic* no controle de temperatura foi realizado um estudo teórico do processo de controle de temperatura extraindo dados pertinentes ao controle dentro das faixas de temperatura para um ambiente residencial.

O controle de temperatura foi realizado utilizando o *RSLogix* 5000 e o *Fuzzy Designer* por meio de simulação em máquina virtual.

No *RSLogix* 5000 foi configurada a comunicação com o Controlador Lógico Programável (CLP) e o *Fuzzy Designer* por meio do protocolo de comunicação OPC *Server*, as portas de entrada e saída (Input/Output) que fazem a leitura dos valores de entrada e aciona as saídas conforme a programação desenvolvida na linguagem Ladder.

No *Fuzzy Designer* foi montada a estrutura do controlador *fuzzy* com os parâmetros gráficos de entrada e saída.

Ao final o próprio *Fuzzy Designer* proporciona monitorar as saídas referente a temperatura conforme a alteração na entrada do controlador, isso conforme a base de regras que foi definida com base no conhecimento do processo.

3 Resultados e Discussão

A função de transferência de um sistema de controle de temperatura ambiente pode ser aproximada para um sistema de primeira ordem com um atraso de transporte, como mostra a equação 1.

$$H(s) = \frac{K}{\tau .5 + 1}$$

Onde K é o ganho, τ é a constante de tempo.

Ao projetar o sistema, a implementação dos parâmetros da função de transferência do processo fora aplicada ao controlador por meio da linguagem Ladder no arquivo importado do *Fuzzy Designer* uma vez que o mesmo ao projetar o controlador passa os parâmetros (função de transferência) de controle na geração do código Ladder a ser carregado no Controlador Lógico Programável.

O Controlador foi implementado em máquina virtual por meio da modelagem *fuzzy, configuração e alterações de parâmetros da lógica Ladder no RSLogix 5000* tendo sua comunicação com seu módulo fuzzy pelo protocolo de comunicação OPC *Server.* Seu monitoramento realizado pelo próprio Fuzzy Designer.

Os conjuntos *fuzzy* utilizados na modelagem do controlador de temperatura do projeto são definidos na Tabela 1, onde a variável var_temperatura é a diferença da temperatura lida por um sensor e o valor do Set_Point podendo variar 15ºC abaixo e 15ºC acima do Set Point. A variável *aux* armazena a diferença do valor de temperatura lido pelo valor anterior dentro de uma faixa de -2ºC a 2ºC, e a variável vel_ventilador é a saída do sistema definida em %. As colunas **a**, **b**, **c** e **d** definem funções de pertinência do tipo trapezoidal com valores dentro das faixas de temperatura definidas e sua respectiva saída.

ENGENHARIA DE COMPUTAÇÃO - Trabalho de Conclusão de Curso - 2017/01

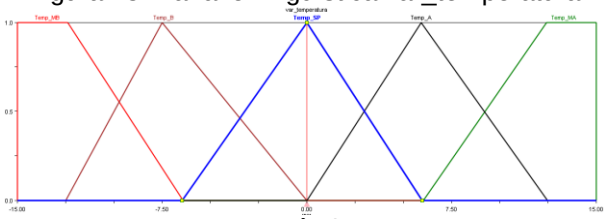
Tabela 1: Variáveis Linguísticas e seus conjuntos *fuzzy*

Variável	Conjunto	A	B	C	D
var_temperatura	Temp_MB	-15.000	-12.387	-15.000	-6.396
	Temp_B	-7.500	-7.500	-12.478	0
	Temp_SP	0	0	-6.442	5.991
	Temp_A	5.946	5.946	0	12.477
	Temp_MA	12.432	15.000	6.036	15.000
aux	Temp_CR	-2.000	-2.000	-2.000	-1.000
	Temp_CD	-1.000	-1.000	-2.000	0
	Temp_estável	0	0	-1.000	1.000
	Temp_SD	1.000	1.000	0	2.000
	Temp_SR	2.000	2.000	1.000	2.000
vel_ventilador	nula	0	0	0	25.00
	vel_baixa	25.00	25.00	0	50.00
	vel_media	50.00	50.00	25.00	75.00
	vel_alta	75.00	75.00	50.00	100.00
	vel_Muito_Alta	100.00	100.00	75.00	100.00

Fonte: Próprio Autor

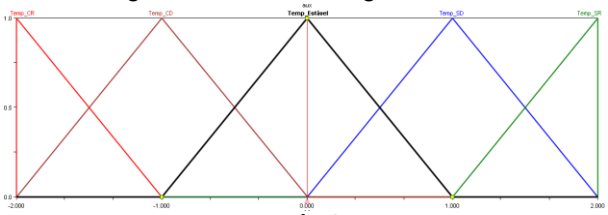
Essas variáveis (desenvolvidas no modelo de funções trapezoidais da *fuzzy logic*), modeladas de forma gráfica no *RSLogix* 5000 *fuzzy designer* fica da seguinte forma como é mostrado nas Figura 13 e respectivamente.

Figura 13: Variável linguística var_temperatura



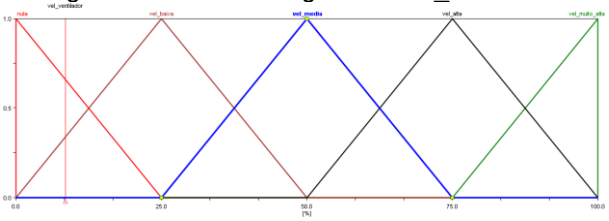
Fonte: Próprio Autor

Figura 14: Variável linguística aux



Fonte: Próprio Autor

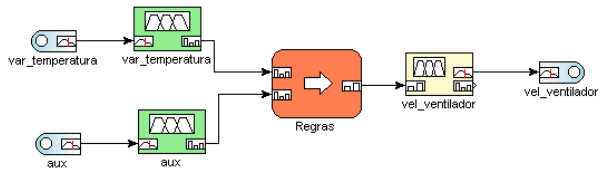
Figura 15: Variável linguística vel_ventilador



Fonte: Próprio Autor

A modelagem completa do Controlador *Fuzzy* é mostrada na Figura 16.

Figura 16: Modelagem Controlador *Fuzzy* Temperatura



Fonte: Próprio Autor

Com base no estudo do processo foi definido 25 regras para o Controlador como mostra a Tabela 2.

Tabela 2: Base de Regras do processo.

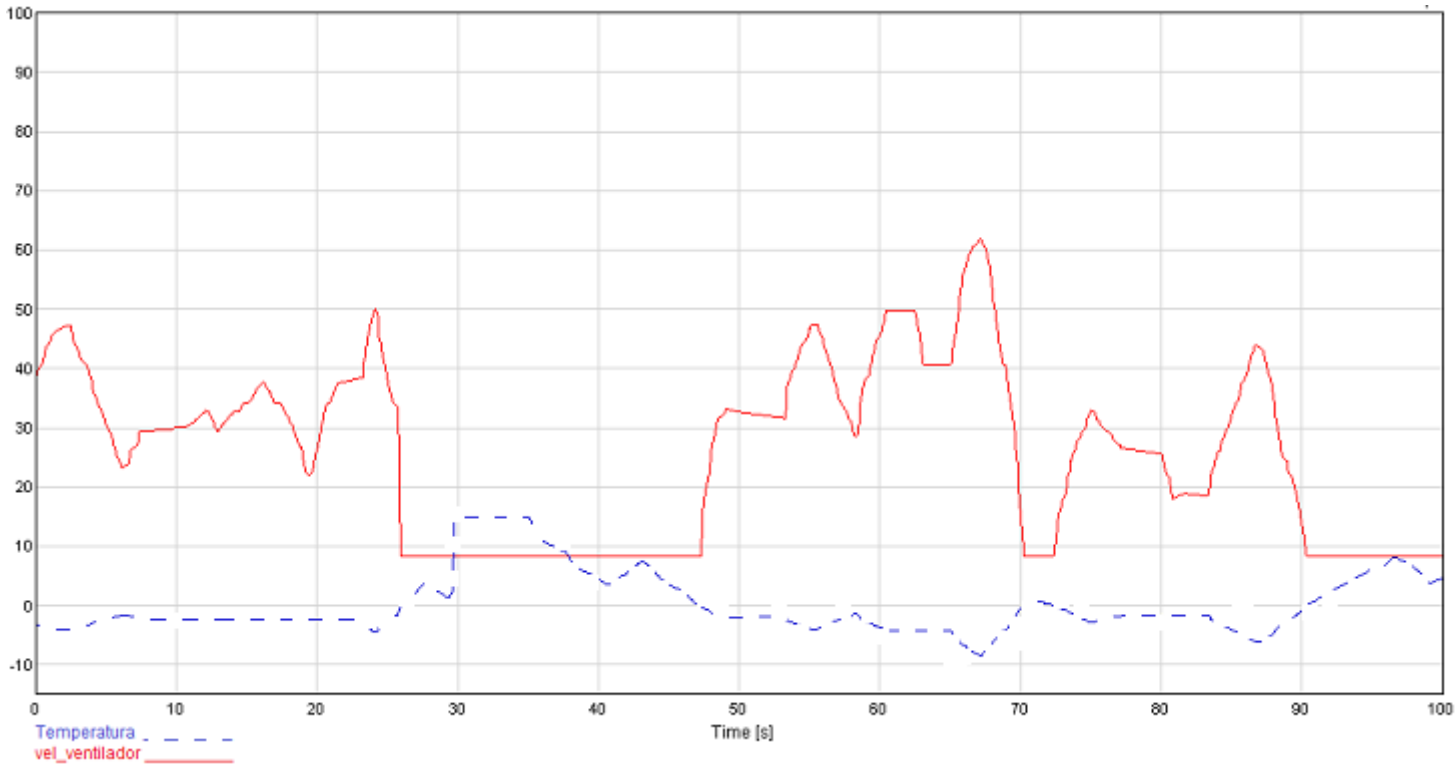
		var_temperatura				
		Temp_MB	**Temp_B**	**Temp_SP**	**Temp_A**	**Temp_MA**
aux	**Temp_CR**	vel_muito_alta	vel_muito_alta	vel_alta	vel_media	vel_baixa
	Temp_CD	vel_muito_alta	vel_alta	vel_media	vel_baixa	vel_baixa
	Temp_Estável	vel_muito_alta	vel_alta	nula	nula	nula
	Temp_SD	vel_alta	vel_media	nula	nula	nula
	Temp_SR	vel_media	vel_baixa	nula	nula	nula

Fonte: Próprio Autor

Ao submeter o controlador por um período de tempo em funcionamento foi possível notar o comportamento da temperatura e consequentemente o comportamento da saída do sistema em função da base de regras, isso podemos notar na Figura 12.

No gráfico nota se um comportamento da temperatura variando dentro de uma faixa de -15ºC a 15ºC do seu Set Point que está representado como 0 (valor simbólico). Em função dessa variação na entrada do controlador é feito um cálculo da diferença da temperatura lida e do Set Point e a variável aux armazena o valor atual de temperatura pelo valor anterior, esses valores são *fuzzificados* no controlador e avaliados pela base de regras que atua na saída do controlador como é possível notar no gráfico o comportamento da variável vel_ventilador que conforme as entradas devido a variação de temperatura irá acionar o ventilador com uma velocidade menor, maior ou de forma estável variando de 0 a 100% como é mostrado no gráfico da Figura 17.

Figura 17: Comportamento do Controlador *Fuzzy* de Temperatura



Fonte: Próprio Autor

4 Conclusão

Neste trabalho foi realizado um estudo exploratório do controle de temperatura utilizando os princípios da *Fuzzy Logic* implementada em um Controlador Lógico Programável (CLP) aplicados no controle de ambientes, e com base nesse estudo foi possível desenvolver o controlador em máquina virtual aplicando as características *fuzzy* em programação Ladder utilizando o RSLogix 5000 e seu módulo *fuzzy* para modelagem *fuzzy.*

O desenvolvimento desse controlador se mostra importante na implementação da *Fuzzy Logic* que é implementada de forma mais rápida e fácil que o PID. Todos esses fatos nos levam a um resultado satisfatório do controlador que obteve bons resultados em sua saída conforme a inferência de sua base de regras mostrando a eficiência no controle de processos desde a indústria até a automação residencial.

Com base nos resultados adquiridos é possível melhorar o controlador para ser implementado na indústria como no controle ambiente de uma sala de máquinas, isso deixa em aberto o projeto para futuras melhorias e implementação.

5 Referências

ALLEN BRADLEY. Logix5000 Controllers Generals Instructions Reference Manual.

Rockwell Automation Publication, 2016. 853 p.

ALLEN BRADLEY. RSLogix 5000 Fuzzy Designer. Publication LOGIX-UM004A-EN-P, 2007. 156 p.

BRAZ, P. H. A.; Viana, D. C. Implementação de Controladores Fuzzy em CLPS de pequeno porte. **XII Simpósio Brasileiro de Automação Inteligente (SBAI)**, 2015.

COSTA, Alex da; RODRÍGUEZ, Antônio Gabriel; SIMAS, Etiene P. Lazzeris; ARAÚJO, Roberto da Silva. Lógica Fuzzy: Conceitos e aplicações. Universidade do Vale do Rio dos Sinos, 2007.

GOMIDE, F. A. C.; Gudwin, R. R. Modelagem, Controle, Sistemas e Lógica Fuzzy. Universidade Federal de Campinas, Campinas, 1994.

OLIVEIRA, I. S. Controle Fuzzy PI de temperatura num modelo de edificação em escala reduzida. 2008. Monografia (Graduação em Engenharia de Controle e Automação) – Universidade Federal de Ouro Preto.

OLIVEIRA, D. N.; Braga, A. P. S.; Almeida, O. M. Fuzzy implementado em Ladder com funções de pertinência descontínuas. **XVIII Congresso Brasileiro de Automática,** Bonito, 2010.

TARIG, Ali Abdurrahman E. S. Controle de um braço robótico utilizando uma abordagem de agente inteligente. 2001. 98 f. Dissertação (Mestrado em Informática) – Coordenação Pós-Graduação em Informática, Universidade Federal da Paraíba, João Pessoa.

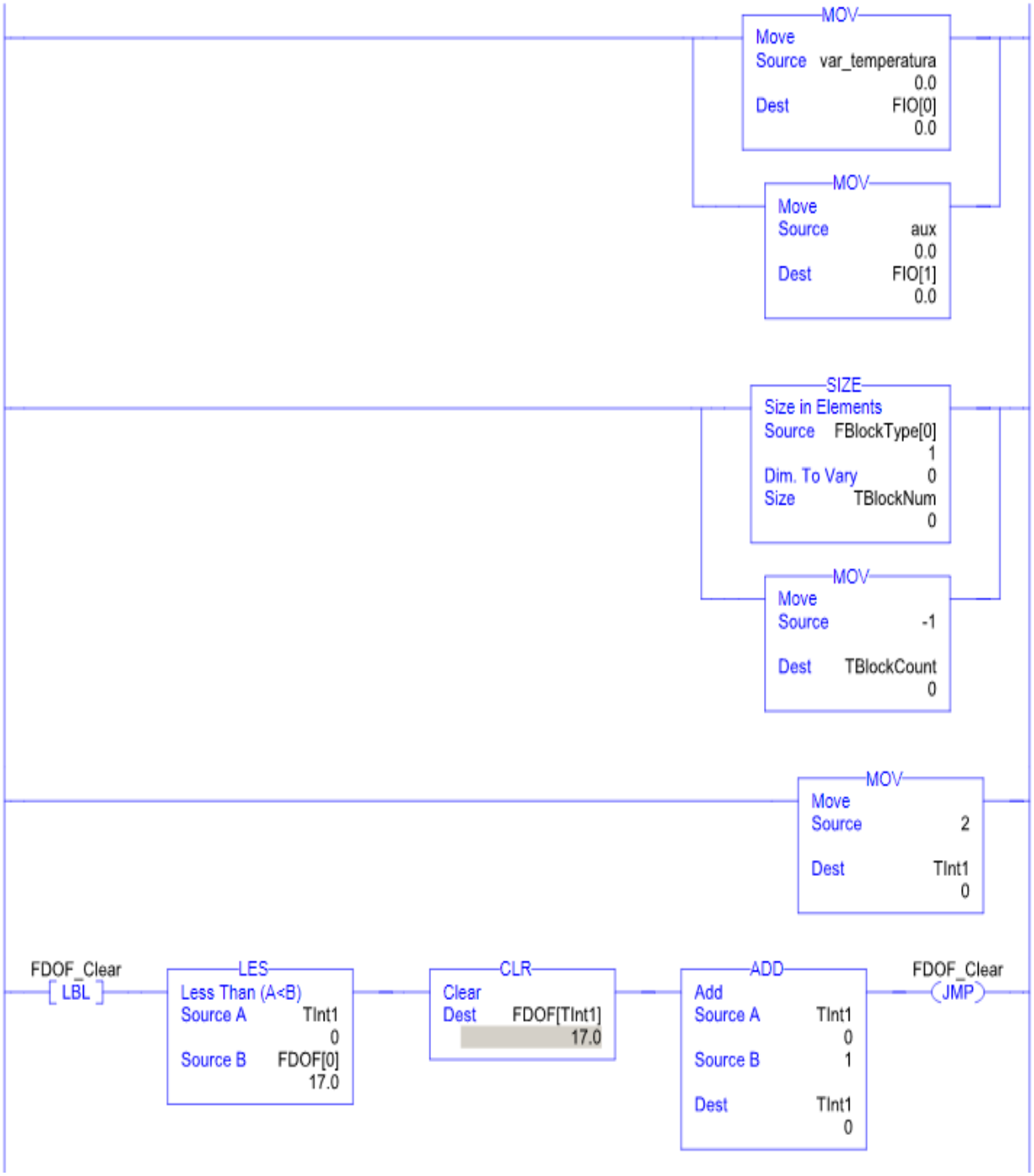
TEIXEIRA, E. P. Noções de Controladores Nebulosos. Universidade de Uberaba, Uberaba, 2012.

ZADEH, L. A. Fuzzy sets. Information and Control, v. 8, p. 338–353, 1965

ENGENHARIA DE COMPUTAÇÃO - Trabalho de Conclusão de Curso - 2017/01

Anexo I

Definindo variáveis de entrada para “fuzzificação”



ENGENHARIA DE COMPUTAÇÃO - Trabalho de Conclusão de Curso - 2017/01

Anexo II

Definindo a quantidade de regras