CENTRO UNIVERSITÁRIO DO SUL DE MINAS
ENGENHARIA ELÉTRICA
LUIS FERNANDO DA SILVEIRA

CONTROLE DE UM SISTEMA DE PRESSÃO DE COMPRESSOR A VAPOR DE AMÔNIA

Varginha
2016

LUIS FERNANDO DA SILVEIRA

CONTROLE DE UM SISTEMA DE PRESSÃO DE COMPRESSOR A VAPOR DE AMÔNIA

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia Elétrica do Centro Universitário do Sul de Minas – UNIS/MG como pré-requisito para obtenção do grau de bacharel sob orientação do Prof. Me. Eduardo Henrique Ferroni.

Varginha

2016

LUIS FERNANDO DA SILVEIRA

CONTROLE DE UM SISTEMA DE PRESSÃO DE COMPRESSOR A VAPOR DE AMÔNIA

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia Elétrica do Centro Universitário do Sul de Minas – UNIS/MG como pré-requisito para a obtenção do grau de bacharel pela Banca examinadora composta pelos membros:

Aprovado em / /

_____

Prof. Me. Eduardo Henrique Ferroni

_____

Prof. Dr. Vinicius Miranda Pacheco

_____

Prof. Ma. Ivana Prado de Vasconcelos

OBS.:

Dedico este trabalho a todas as pessoas importantes em minha vida.

AGRADECIMENTOS

Agradeço aos meus pais, Maria Aparecida e Anízio pelo carinho e apoio e dedicação a minha trajetória, sem eles nada disso seria possível.

Agradeço ao professor Eduardo Henrique Ferroni, meu orientador pela ajuda no desenvolvimento deste trabalho e auxilio fornecido.

Agradeço o meu irmão Paulo, pelos momentos de apoio e auxilio em minhas ideias.

Agradeço a todos meus amigos e colegas, em especial Ewerton e Larissa, que sempre me ajudaram e que tornaram esse caminho menos cansativo e doloroso.

"A inteligência é o poder que torna possível realização dos sonhos"

Rubem Alves

RESUMO

Este estudo tem como objetivo propor a automatização de um sistema de refrigeração empresarial de uma empresa localizada na cidade de São Gonçalo do Sapucaí/MG. O atual sistema é manual, não atendendo assim as necessidades da empresa; além de apresentar efeitos colaterais de funcionamento, já que afeta diretamente a linha de produção acarretando em grandes prejuízos. A temperatura da água que é controlada pelo mecanismo atual, não pode sofrer variações superiores a 2%, desta forma o controle manual torna-se ineficiente, necessitando de um controle automático. Para sanar tal problema, é necessária a análise de todo o sistema de controle de refrigeração, a fim de propor soluções ou recursos viáveis minimizando ou em alguns casos extinguindo os problemas existentes no sistema atual.

Palavras-chave: Automatização. Tecnologia. Controle

ABSTRACT

This study has the objective to propose an automation of a refrigeration company system located in São Gonçalo do Sapucaí/MG city. The current system is manual and don't meets the company needs, besides of presenting the side effects of working, once that affects the production line directly resulting in high losses. The water temperature is controlled by an actual mechanism and cannot suffer variations higher than 2%, in this way the manual control becomes inefficient, and needs an automatic control. To resolve this problem, it's necessary to analyse all refrigeration control system, to propose solutions or feasible resources to minimize or in some cases extinguish the problems present on the actual system.

Keywords : *Automation . Technology. Control*

LISTA DE FIGURAS

LISTA DE ABREVIATURAS

RTD - ResistanceTemperature Detector

NTC - Negative TemperatureCoefficient

PTC - Positive TemperatureCoefficient

PID - Proportional Integral Derivative

IA - Inteligência Artificial

JVM - *Java* Virtual Machine

FC - *FuzzyControl* ou *Fuzzy controle*

COG - Center ofGravity ou centro de gravidade

APIs - *ApplicationProgramming Interface*

PC - Controladores Programáveis

T - Temperatura

S - Entropia

K - Ganho proporcional

U - Conjunto universal

A - Conjunto

SUMÁRIO

1 INTRODUÇÃO

Os sistemas térmicos de troca de calor podem vir a apresentar defeitos oriundos de interferências vindas do meio externo. Na indústria surge a necessidade de aumentar os investimentos em soluções para atender aos critérios de desempenho, como por exemplo, investindo em seus equipamentos de refrigeração permitindo assim a realização desta operação de maneira automática, rápida e eficiente.

As consequências das falhas em sistema de refrigeração atingem diretamente a produção, causando atraso no desempenho e gerando prejuízos materiais e financeiros. Tal desconforto gera a necessidade de analisar este problema criando soluções viáveis para a situação em questão.

Nota-se a importância de controlar o sistema, uma vez que, quando bem projetado e dimensionado, evita maiores desconfortos em sua plena operação, reduzindo perdas e prejuízos na linha de produção. Existem relatos de grandes perdas e de dificuldades enfrentadas para solucionar o mais rápido possível este incomodo amenizando o período de não produção (LOPES; JAFELICE, 2005).

Fazer com que o sistema de controle seja manipulado de forma simples e eficaz, facilitando a interface de homem-máquina proporciona uma resposta mais eficiente na presença de eventuais mudanças dos parâmetros no equipamento (CINGOLANI, 2012).

Com o controle adequado do equipamento torna-se possível a obtenção da redução do consumo de energia elétrica do sistema de refrigeração, já que os equipamentos realizam o trabalho em forma de estágios. Com a redução do fluído refrigerante de cada estágio que o equipamento trabalha ocorre também a redução da força exercida no pistão do compressor, proporcionando assim que o motor trabalhe com um nível menor de carga em seu eixo, reduzindo em torno de 10 a 15% a corrente elétrica consumida.

Este trabalho tem propósito de apresentar soluções a fim de melhorar a situação atual do sistema de refrigeração encontrado na empresa. Faz-se necessário realizar estudos de várias situações existentes, como o diagnóstico dos problemas encontrados, os dados referentes aos equipamentos de refrigeração e todo seu sistema de controle.

Todo o sistema em si deve ser analisado a fim de propor soluções viáveis e plausíveis para este feito (RICH; KINGHT, 1993).

2 INTELIGÊNCIA ARTIFICIAL

O ser humano desde a antiguidade desenvolve ferramentas com a finalidade de facilitar o seu dia a dia. Com o avanço da tecnologia percebe-se a necessidade de tomadas de decisões cada vez mais rápidas e precisas.

A criação do computador foi um marco histórico que facilitou o dia a dia e permitiu que as máquinas ajudassem na tomada de decisões difíceis, dando início a era da tecnologia e do relacionamento homem *versus* máquina (RICH; KINGHT, 1993).

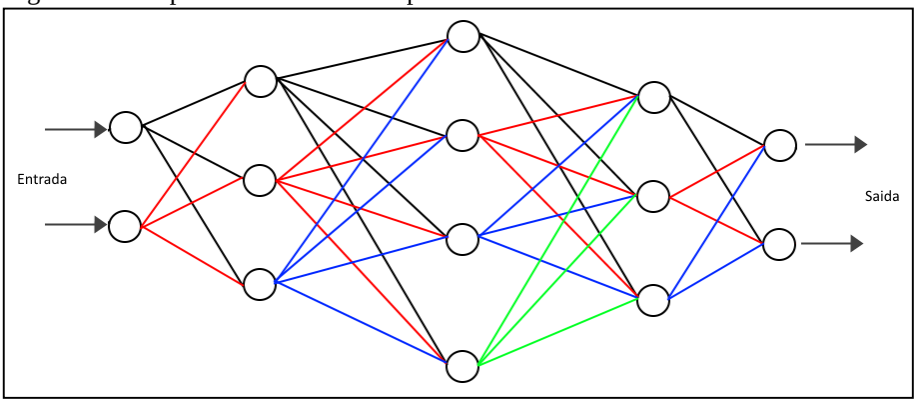
A Inteligência Artificial (IA) é a tecnologia que utiliza técnicas de programação computacional, armazenamento de dados e análise das melhores opções de solução, permitindo que as mesmas sejam tomadas nos sistemas de controles e processos (RICH; KINGHT, 1993).

2.1 Rede neurais

No cérebro humano o neurônio recolhe sinais e impulsos através de várias séries de acontecimentos que se dividem em milhares de ramificações. No final de cada ramo converte-se a atividade e transmitem-se estímulos para outro ramo de outro neurônio, assim estimulam-se as atividades nos neurônios interligados. A rede de neurônios do cérebro forma um sistema de processamento de informação massivamente paralelo. A aprendizagem ocorre alterando a eficácia dos ramos de modo que a influência de um neurônio tenha em outro, de uma forma o cérebro aprende, alterando os pontos fortes de conexões entre os neurônios e adicionando ou excluindo conexões entre os neurônios (TONELLI NETO, 2012).

No desenvolvimento de armazenamento de dados e análises para obter os melhores desempenhos, o neurônio artificial é um modelo computacional inspirado nos neurônios naturais e têm ajudado na eficácia das tomadas de decisões. O sinal neural natural ocorre através de uma entrada do ramo localizado no neurônio, ramo este que, quando ativado emite um sinal de saída. Este sinal pode ser enviado para outra entrada de neurônio e pode ativar outros neurônios como representado na Figura 1.

Figura 1. Exemplo de rede neural simples



Fonte: O autor.

A complexidade dos neurônios reais é altamente abstraída na modelagem artificial dos neurônios. Estes consistem basicamente de entradas que são multiplicadas por valores e então calculadas por uma função matemática que determina a ativação do neurônio. Outra função calcula a saída do neurônio artificial, combinando-os entre si a fim de processar a informação. Quanto maior for a influência de um neurônio artificial maior será o valor de entrada, devido à multiplicação entre eles. A entrada também pode ser negativa, por isso podemos dizer que o sinal é inibido pela influência negativa. Dependendo dos valores, o cálculo do neurônio será diferente.

Ao ajustar os valores de um neurônio artificial podemos obter a saída esperada para entradas específica, mas, quando há uma rede neural composta por centenas ou milhares de neurônios fica bastante complicado definir e modificar todos os valores necessários. Porém, através de algoritmos, tais valores podem ser ajustados na tentativa de obter o valor desejado de saída. Este processo de ajustamento das ponderações é chamado de aprendizagem ou formação (BATISTA, 2013).

A rede neural com algoritmo de aprendizagem tem a capacidade de reduzir o erro, que consiste na diferença entre o resultado real e o esperado, até que a rede neural aprenda com várias operações de interações de dados. A aprendizagem começa com valores aleatórios, e o objetivo é ajustá-los para que o de erro seja o mínimo. Os neurônios artificiais são organizados em camadas a fim de enviar sinais e erros, podendo haver uma ou mais camadas de neurônios antes do resultado final. O algoritmo que reprograma dos neurônios supervisiona. Significa que pode ser fornecido o algoritmo de entradas e saídas que são esperados para um determinado sistema, tal algoritmo funciona como base, a fim de reduzir os erros (RICH; KINGHT, 1993).

2.2 Algoritmos genéticos

Os Algoritmos Genéticos (AGs) são baseados na linguagem da genética natural e da evolução biológica. No acoplamento molecular cada variável de estado corresponde a um gene, essas ligações das variáveis formam conjuntos, conjuntos os quais apresentam a orientação e a configuração que mantêm unido o acoplamento. Então o arranjo é ligado a estes conjuntos (RICH; KINGHT, 1993).

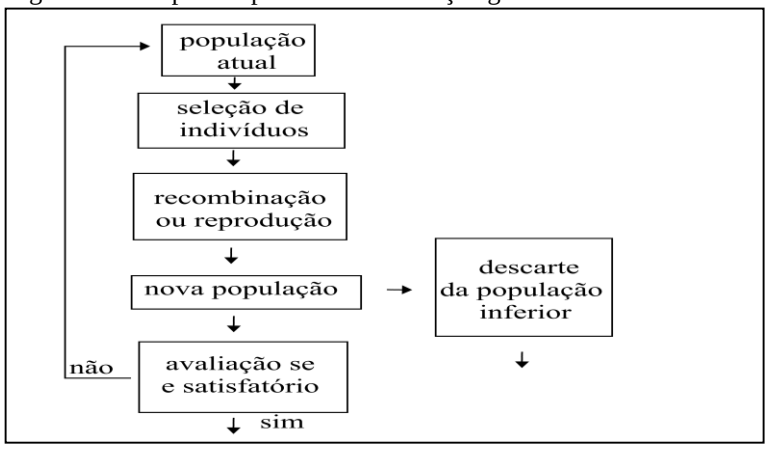
No sistema biológico molecular selecionam-se pares aleatórios de indivíduos que são acoplados utilizando um processo de cruzamento, onde novos indivíduos herdam genes de ambos os pais. Além disso, alguns descendentes sofrem mutação, um gene muda por uma questão aleatória. A seleção da prole da geração atual ocorre com base no indivíduo que tenha aptidão para soluções que promovam melhores adaptações ao seu ambiente de reprodução, e então, os mais inferiores não se adaptam e acabam por morrer (RICH; KINGHT, 1993).

A seleção de forma iterativa é um método que utiliza um grupo de indivíduos para criar novos candidatos evoluindo a população em busca da otimização de problemas. Ocorre da mesma forma como na natureza, onde o indivíduo torna-se melhor adaptado ao seu ambiente através da evolução de sua população (CASTRO, 2001).

Como na evolução biológica os AGs pertencem a uma classe de métodos para encontrar soluções, eles são conhecidos como programação evolucionária, onde utiliza-se o conceito de evolução para buscar uma solução eficiente para determinado problema através da repetida aplicação de mutação, seleção e reprodução (CASTRO, 2001).

A programação do algoritmo genético é feita através do seguinte ciclo de avaliação e aptidão de todos os indivíduos da população: cria-se uma nova população por operações que executam a reprodução e mutação dos indivíduos, após isso descarta-se a população velha e por fim interage-se novamente com a nova população criada, conforme pode ser demonstrado mais simplificadamente na Figura 2 (RICH, 1993).

Figura 2. Exemplo simplificado de evolução genético



Fonte: O autor.

A repetição do ciclo reprodução faz com que os filhos fiquem idênticos aos pais. Em alguns casos, portanto, os indivíduos da população, através da mutação, podem ganhar características mais evoluídas através de uma nova interação, gerando uma população mais eficiente (CASTRO, 2001).

2.3 Logica *Fuzzy*

O significado do termo do *Fuzzy* é nebuloso, difuso, alguma coisa que não se encontra legível, e se refere ao fato de não compreendermos completamente os sistemas em si que estamos analisando. Nos problemas que se encontram na física e matemática, não se encontra dificuldades em classificar os elementos encontrados como pertencentes ou não a um dado conjunto. Assim, se temos um conjunto A e um elemento x do conjunto universo U conseguimos muitas vezes dizer se $x \in A$ ou $x f \in A$ (SANTOS, 2003).

Com o crescimento e a necessidade de técnicas de Inteligência Artificial, para melhorias na solução de problemas mais difíceis, ocupando cada vez mais espaço e o foco em pesquisas na área de controle de processos industriais, aos poucos, a Inteligência Artificial começa a ser implantadas em plantas industriais (SANTOS, 2003).

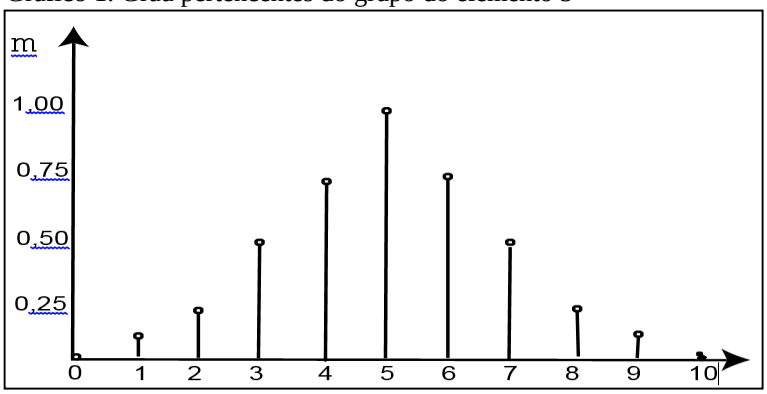
A descrição completa de um sistema real em muitos casos exige informações extremamente detalhadas. Assim, para ser descritas, necessitam destes termos imprecisos que não podem ser modelados pela matemática tradicional de conjuntos. Podemos citar exemplos simples, que estão em nosso cotidiano (NOBREGA SOBRINHO, 2011).

Existem inúmeras situações em que não conseguimos definir em qual conjunto pertence o elemento que será controlado para obter a tomada de decisão, pois estes não estão bem definidos. A intenção de *Zadeh* foi justamente modelar de forma a se obter uma flexibilidade

para que os elementos se adaptem a um dos conjuntos, criando um conceito de grau pertencente. Podendo, assim, um determinado elemento pertencer parcialmente a um conjunto. Este conceito foi publicado por *Zadeh* em 1965, mesmo ano em que a teoria dos conjuntos *Fuzzy* foi criada (NOBREGA SOBRINHO, 2011).

Para criar um conjunto de elementos aproximados da lógica nebulosa, temos que definir o grau pertencente deste grupo, como por exemplo, pode-se definir conjuntos iguais a 5. Conforme demonstrado no Gráfico 1 esta resposta dependerá do contexto, logo *Zadeh* utiliza uma função que deverá fornecer o quanto determinado número pertence ao conjunto considerado. Com isto denomina-se K o conjunto dos números aproximados ao valor estipulado, pertencentes aos números naturais. Podemos propor, por exemplo, uma função de pertinência onde: 10 ∈ K com grau de pertinência 0,00 (corresponde a não pertinência clássica) e 2 e 8 ∈ K com grau de pertinência 0,25 (SANTOS, 2003).

Gráfico 1. Grau pertencentes do grupo do elemento 5



Fonte: Alberto, 2011.

A extensão da função característica da lógica tradicional para o intervalo [0,1], se derivou criando os conjuntos *Fuzzy* que ajudaram e possibilitam, entre outras coisas, a utilização de variáveis linguísticas, que facilitam a compreensão e explicação do conhecimento humano para a criação e desenvolvimento de vários sistemas (SANTOS, 2003).

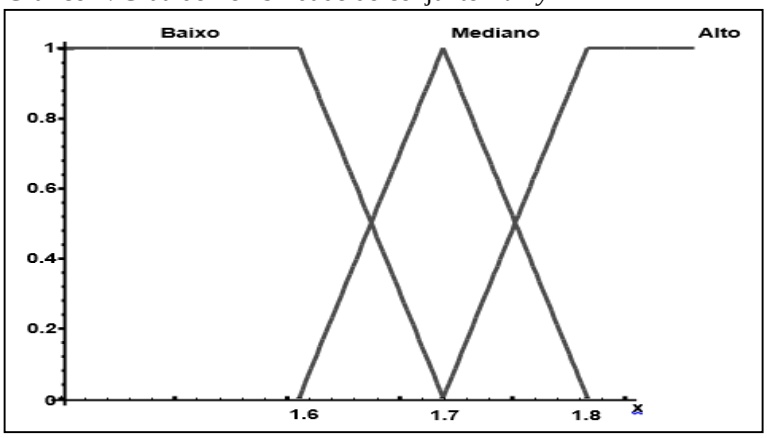
2.4.1 Conjuntos *Fuzzy*

Na teoria dos conjuntos *Fuzzy* ou nebulosos, existe um grau de pertinência para cada elemento assistente a um determinado conjunto. Nos conjuntos *Fuzzy* há uma flexibilidade de não possuir limitações nas definições que foram introduzidas, isso ocorre porque os conjuntos

tradicionais são limitações para lidar com problemas onde ocorre a transição (NOBREGA SOBRINHO, 2011).

Os conjuntos nebulosos podem ser representados pelo seguinte exemplo: a partir da informação de que uma pessoa apresenta 1,74 metros de altura é possível afirmar que esta pessoa pertence ao grupo de pessoas altas? E se uma pessoa tiver 1,75 metros de altura? Tomando como padrão a definição de homem alto pela medida maior ou igual a 1,80 metros, mediano pela medida 1,70 metros e baixo pela medida 1,60 metros, fica claro que não existem limites bem definidos que separem os elementos do conjunto das pessoas altas dos elementos do conjunto das pessoas não altas, conforme pode ser representado pela Gráfico 2 (SANTOS, 2003).

Gráfico 2. Grau de flexibilidade do conjunto *Fuzzy*



Fonte: Alberto, 2011.

2.3.2 Regras *Fuzzy*

As regras *Fuzzy* são estruturas utilizadas em várias abordagens diferentes na teoria *Fuzzy* e que podem ser entendidas de diversas maneiras. Com isso, as regras *Fuzzy* descrevem situações a serem analisadas especificamente, que podem ser aplicadas em uma análise de uma planta industrial, cuja inferência nos conduz a algum resultado desejado. As inferências que são retiradas das regras *Fuzzy* podem também ser entendidas como mapeamentos de conjuntos de entradas do sistema para um conjunto de saídas (como em um esquema de interpolação) (SANTOS, 2003).

A regra *Fuzzy* é capaz de armazenar algum conhecimento específico das características do sistema, sendo capaz de descrevê-lo em suas variadas possibilidades de atuação. As regras nebulosas apresentam uma afirmação clássica, e são compostas por uma parte antecedente (a

parte Se) e uma parte consequente (a parte Então), resultando em uma estrutura do tipo antecedentes e consequentes (NOBREGA SOBRINHO, 2011).

Os antecedentes têm a característica de descrever a condição de análise detalhando-se no estudo, enquanto a parte consequente descreve uma conclusão ou uma amostra de um comportamento que pode ser demonstrado quando as análises terminam. A diferença entre os antecedentes de uma regra *Fuzzy* e os de uma regra tradicional é que os primeiros descrevem uma condição que pode ser parcialmente satisfeita ou satisfatória, enquanto os últimos descrevem uma condição rígida (a regra aplicada não funciona se os termos não são satisfeitos completamente) (NOBREGA SOBRINHO, 2011).

Os antecedentes têm a necessidade de definir uma região *Fuzzy* no espaço das variáveis de entrada do sistema. Já os consequentes descrevem uma região no espaço das variáveis de saída do sistema. A construção dos antecedentes do sistema muitas vezes resulta na classificação, assim, a elaboração do sistema se foca no conhecimento empírico sobre a dinâmica do sistema. Com isto se espera que a elaboração dos consequentes de uma regra seja mais complexa do que a dos antecedentes (NOBREGA SOBRINHO, 2011).

Ao constituir os conjuntos de regras *Fuzzy* necessita-se de um compilador de inferência para extrair dela a resposta final. Existem diversos métodos de inferência, a escolha do método depende do sistema no qual ele está sendo aplicado (NOBREGA SOBRINHO, 2011).

As regras são processadas em paralelo, ou seja, todas as regras (circunstâncias) são consideradas ao mesmo tempo, e, ao final obtemos uma resposta que pode ser tanto um valor numérico clássico, quanto um conjunto *Fuzzy* ou um funcional, de acordo com o tipo de consequente utilizado (NOBREGA SOBRINHO, 2011).

3 SISTEMAS DE CONTROLE

Os sistemas de controle têm desempenhado um papel essencial no crescimento e avanço da engenharia e da ciência para a melhoria de processos (OGATA, 2010).

Com os avanços tecnológicos, o sistema de controle fornece as melhores opções possíveis para atingir o desempenho ideal de sistemas dinâmicos. Com a melhoria e eficiência do controle de um sistema ocorre a melhora na produtividade aliviando o trabalho (OGATA, 2010).

Um sistema consiste da combinação de componentes que agem em conjunto com o intuito de se obter uma saída desejada (OGATA, 2010).

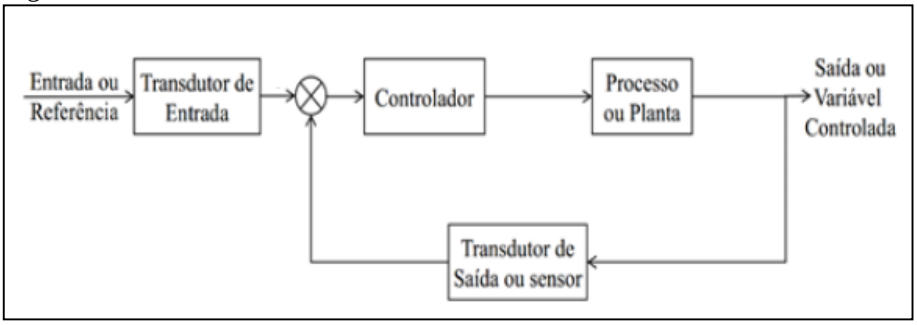
ESTÍMULO → SISTEMA DE CONTROLE → SAÍDA

Existem vários tipos de sistemas de controle, um deles é o sistema com realimentação que faz a comparação entre a saída e a entrada, denominada de sistema de controle de malha fechada como demonstrado na Figura 3. O sinal de erro realimenta o controlador de modo que se minimize o erro ajustando a saída do sistema ao valor desejado (OGATA, 2010).

As características do sistema de malha fechada são:

a) A resposta do sistema é relativamente insensível a distúrbios e variações internas nos parâmetros do sistema;
b) São precisos quando devidamente projetados;
c) São complexos e, consequentemente, caros;
d) A estabilidade é um aspecto importante a ser considerado, pois pode apresentar uma tendência de correção de erros além do necessário, causando oscilações de amplitude constante ou variável (OGATA, 2010).

Figura 3. Sistema de malha fechada



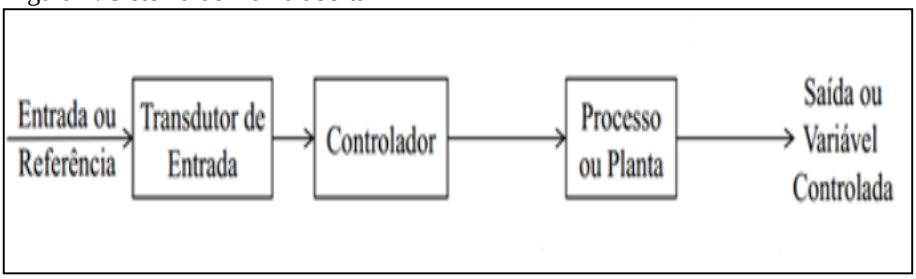
Fonte: O autor.

Outro tipo de sistema é o sistema de controle de malha aberta, que pode ser demonstrado na Figura 4. Tal sistema que não possui uma realimentação em sua saída, consequentemente não há comparação da saída com a entrada (OGATA, 2010).

Os sistemas de malha aberta possuem as seguintes características:

a) Cada entrada de referência corresponde uma condição fixa de operação. Assim, a precisão do sistema depende de uma calibração;
b) Na presença de distúrbios, o sistema em malha aberta não vai executar a tarefa desejada;
c) Utilizado na prática se a relação entre a entrada e a saída for conhecida e se não houver nenhum distúrbio;
d) O sistema é fácil de ser construído e tem manutenção fácil;
e) Não apresentam problemas de estabilidade (OGATA, 2010).

Figura 4. Sistema de malha aberta



Fonte: O autor.

O problema do controle de temperatura de um sistema de refrigeração, pode ser resolvido de várias formas diferentes, cada uma com suas vantagens e desvantagens. A resolução pode ser realizada tanto no desenvolvimento inicial projeto quanto em relação ao resultado final desejado do projeto. O método mais simples do controle é o liga e desliga, aquele no qual o sistema liga quando uma determinada condição atinge o valor que se espera e desabilita no momento em a condição seja diferente da esperada. Para um controle mais eficiente e sofisticado, usam-se o controle integral, proporcional, derivativo, ou a combinação deles, que melhora consideravelmente o desempenho e a resposta do sistema (OGATA, 2010).

Um tipo de lógica muito utilizada é a *Fuzzy*, que é utilizada para facilitar o controle em sistemas que apresentam a necessidade de um projeto matemático. No decorrer deste Trabalho de Conclusão de Curso será feito um resumo simples de cada uma dessas estratégias de controle de sistema.

3.1 Controlador Liga-Desliga ou *On-Off*

O sistema de controle liga e desliga é bastante simples de se obter. O controlador ou equipamento tem a função de chaveamento que liga assim que o sistema atinge um valor máximo e desliga quando atinge um valor mínimo estabelecido. Em um projeto no qual o controlador necessitará manter a temperatura de um sistema de refrigeração por amônia relativamente constante a variável de interesse é a própria temperatura da água que se deseja controlar, e os limites são definidos pela aplicação, com a finalidade de atender aos requisitos propostos (OGATA, 2010).

Um sistema que queira manter a temperatura da água sempre em torno de 1°C com histerese de 0,5°C pode ligar o sistema de refrigeração quando a temperatura estiver superior a 1,5°C e desligar quando a temperatura alcançar um valor em torno de 0,5°C. Este controlador é bastante utilizado em diversos sistemas de refrigeração, pois atende a necessidade e a eficiência fazendo com que a temperatura acompanhe a referência do *setpoint* para condição estabelecida do sistema, é portanto, um sistema bem simples de se controlar (OGATA, 2010).

O controlador liga e desliga apresenta a característica da álgebra booleana. Surgiu em torno de 1854 por *George Boole*. Nos meados de 1938, tal lógica foi aplicada em circuitos elétricos e de chaveamento com intuito de mostrar as propriedades que podem assumir dois valores distintos (OGATA, 2010).

A particularidade da álgebra booleana é que ela pode assumir dois valores possíveis, sempre tendo uma saída desejada (y) que pode ser denotada como 0 e 1, falso ou verdadeiro (F,V) ou *High and low* (OGATA, 2010).

A lógica binária pode assumir somente dois valores: verdadeiro ou falso. Há uma incerteza devido à grande ramificação de valores que podem ser assumidos neste intervalo de verdadeiro ou falso, como por exemplo: Se uma pessoa tem como altura 1,80 m a pessoa será alta, se tiver 1,50 m será baixa, então a pessoa com 1,75 m não se encontra no conjunto de altos e baixos. Na tentativa de solucionar essa questão, a lógica nebulosa cria vários conjuntos para definir incertezas, como: não tão alto/ não tão pequeno ou muito alto/ muito pequeno entre outros (CAMBOIM, 2008).

Um ponto negativo é o consumo de energia e a deficiência de uma resposta mais estável. Essa é uma estratégia que não apresenta como objetivo manter um valor mais próximo do desejado, mas sim, manter a temperatura o mais próxima do referencial. Não que o sistema como um todo não atenda a necessidade que se espera, o que acontece é que o sistema não é muito eficaz no aspecto de manter um valor (LOPES; JAFELICE, 2005).

3.2 Controle PID

Os controladores PID são compostos por três partes, proporcional, integral e derivativa. Os controladores proporcionais introduzem um ganho no sistema e diminuem o erro em regime, desde que o ganho não seja muito alto, pois poderá tornar o sistema instável. O controlador derivativo é utilizado para amenizar o erro da resposta transitória, já o controlador integral é usado para retirar o erro em regime permanente. Quando bem projetado, o controle PID é aquele que apresenta uma característica de resposta do sistema de acordo com as especificações, pois trabalha com o erro entre a variável controlada e a resposta atual do sistema visando (OGATA, 2010).

O maior problema na característica do controlador PID é o modelo matemático do sistema. Se o modelo não for satisfatório, a resposta do sistema ficará comprometida. O modelo do sistema controlado nem sempre é fácil de ser encontrado e, no caso do controlador de temperatura, existem muitas variáveis difíceis de serem medidas por causa das trocas de calor que acontecem entre os elementos do meio ambiente que será controlado. O projeto muitas vezes acaba exigindo cálculos excessivos e de alto grau de complexidade, com modelos de ordem elevada. É uma estratégia de controle muito utilizada em sistemas de automação que precisam atender a necessidades da forma mais completa possível. Para sistemas térmicos, nos quais alguma variação na resposta não gera grandes problemas, não é muito utilizado (OGATA, 2010).

O controlador proporcional é um amplificador que tem um ganho que permite ser ajustado e é representado pela letra K. O aumento do ganho *K*, afeta o sistema diminuindo o erro de regime, aumenta ao máximo o sobre-sinal e torna o sistema mais oscilatório, podendo levá-lo a uma instabilidade (OGATA, 2010).

O controlador integral afeta o sistema eliminando por completo o erro de regime estacionário, mas pode piorar a resposta transitória do sistema, inclusive levando a instabilidade. O sistema fica mais oscilatório, apresentando um sobre-sinal mais elevado (OGATA, 2010).

Uma ação de controle derivativo é obtida quando se acrescenta um controlador proporcional, permitindo a obtenção de um controlador de alta sensibilidade, prevendo os acontecimentos do erro atuante e antecipando uma ação corretiva, aumentando a estabilidade do sistema, não afetando diretamente o erro estacionário, mas aumentando o amortecimento do sistema, permitindo o uso de um valor mais elevado do ganho *k* (OGATA, 2010).

3.3 Controlador *Fuzzy*

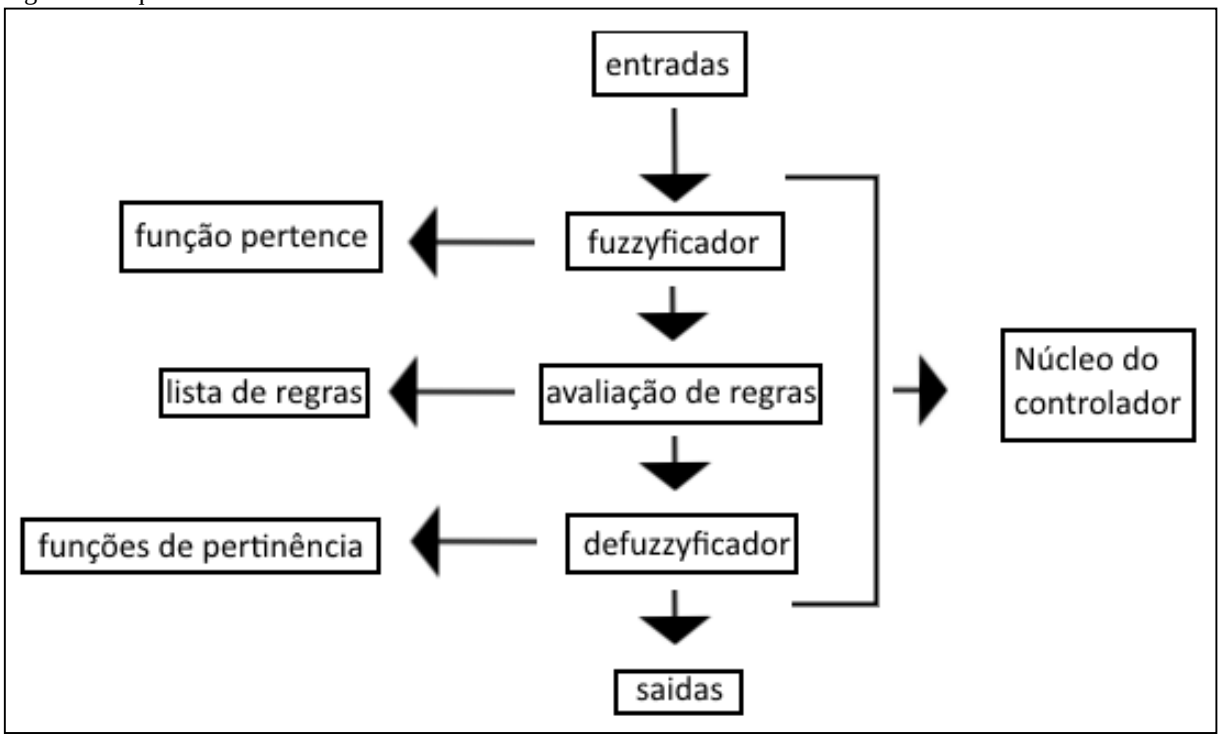
O controlador *Fuzzy* tem como função melhorar a resposta do sistema, sem a necessidade de exigir um modelo matemático muito complexo. Diferente das lógicas tradicionais, que manipulam com condições que são verdadeiras ou falsas, a lógica *Fuzzy* é mais voltada para decidir se as variáveis podem ser parcialmente verdadeiras ou parcialmente falsas (PINTO, 2010).

Para dar um exemplo, quando se pensa em fazer um controlador de temperatura de um sistema de refrigeração utilizando a lógica *Fuzzy*, modela-se a temperatura não como uma variável numérica, mas sim como um estado físico. Se a temperatura estiver em torno de 10°C, para a maioria das pessoas pode ser muito gelado, já para outras, um pouco gelado (PINTO, 2010).

Analisando o exemplo acima, e montado um sistema chamado fuzzyficação, transforma-se as variáveis físicas em variáveis fuzzyficadas na tentativa de obter uma solução para controlar o sistema. Ao final do processo, obtêm-se a desfuzzyficação que é o valor que irá gerar a ação do controlador do sistema (PINTO, 2010).

Todo o sistema de controle baseado na lógica *Fuzzy* precisa encomendar sua estrutura lógica na inferência *Fuzzy* no núcleo do controle, representado na Figura 5. Na representação, o núcleo consiste em três blocos, onde cada função é especificada dentro do algoritmo. Os três blocos são denominados de: Fuzzyficador ou fuzzyficação, regras de avaliação e defuzzyficador ou desfuzzyficação. O bloco Fuzzyficador é formado por diversos e diferentes conjuntos *Fuzzy* de entrada, que são responsáveis por transformar o não distorcido valor de entrada. Informações Fuzzyficadas entram nas regras de bloqueio de avaliação, onde irão relacionar conjuntos de entrada difusos com uma saída distorcida em relação às regras. As informações obtidas à partir das regras de avaliação entram no bloco defuzzyficador para transformar, difundir esta entrada em um valor não *Fuzzy* à tomada de núcleo. Nestes blocos de funções de pertinência é definida uma lista de regras e funções de pertinência, formando um sistema de base de conhecimento (PINTO, 2010).

Figura 5. Esquema básico de um sistema difuso



Fonte: O autor.

4 SENSORES

Sensores são dispositivos sensíveis às formas de energia encontradas no ambiente, tais como pressão, temperatura de fluídos, umidade, vibração, entre outros e que são utilizados para monitoramento de fenômenos físicos. Os sensores possuem um elemento transdutor e fornecem um sinal de tensão ou corrente que corresponde à grandeza física. Por outro lado, um dispositivo transdutor tem a finalidade de converter uma forma de energia em outra ou alterar uma determinada grandeza elétrica, como capacitância, resistência e indutância dependendo de onde ela será aplicada (ARENY, 2003).

O sinal de resposta emitido por um sensor, normalmente, deve ser modificado antes de ser utilizado por um sistema de controle, pois nem sempre a corrente e a tensão estão no nível de funcionamento requerido por este sistema de controle. Para isso, se utiliza um circuito de condicionamento de sinal. Os sensores podem ser divididos em dois tipos: os analógicos e os digitais (ARENY, 2003).

Os sensores analógicos podem assumir qualquer valor de tensão ao longo do tempo em seu sinal de saída dentro de sua faixa de operação. Por outro lado, os sensores digitais podem apenas assumir dois valores distintos de tensão, que são interpretados como zero ou um lógico (ARENY, 2003).

4.1 Sensores tipo termo par

Os sensores termopar ou sensor par termoelétricos são construídos por dois diferentes materiais, que produzem o efeito de *seebeck* onde há uma diferença de potência na junção de dois materiais devido a diferentes temperaturas (ARENY, 2003).

O princípio de funcionamento do termopar é que seu material é submetido a um gradiente térmico em uma de suas extremidades. Com isto, a região exposta à fonte de calor começará a ter mais energia se acumulando na extremidade devido à migração dos elétrons. Já na junção de dois materiais, quando uma extremidade é submetida a uma fonte térmica os elétrons tendem a migrar ocorrendo uma conversão de energia térmica em energia elétrica. Com a variação de temperatura se tem a variação da diferença de potencial, então podemos medir a temperatura da diferença de potencial (ARENY, 2003).

a) Existem vários tipos de termopares, sendo os mais fáceis de encontrar no mercado:
b) O par termoelétrico topo (*k*) e constituído por cromel (ligação níquel 90% mais cromo 10%) polo positivo e alumel (ligação de alumínio 5% e níquel 95%) polo negativo.

c) O termo par tipo (*E*) e consistido cromel polo positivo e constantan (ligação de cobre mais ou menos 53% a 57%, níquel 0,5% a 1,2% e ferro 0,5%) polo negativo.

d) O termo par Tipo (*J*) tem em sua composição ferro polaridade positivo e Constantan polaridade negativa (OMEGA, 2015).

O Quadro 1 abaixo simplifica mais fácil alguns tipos de termo par

Quadro1. Valores e constituição de sensores par

Modelo	Constituição	Faixa de trabalho (°C)
B	Platina/ 30% Ródio-Platina	0 – 1800 ºC
C	Tung-5% Rénio/Tung-26% Rénio	0 – 2320 ºC
E	Cromel / Constantan	-270 – 1000 ºC
G	Tungsténio/Tung-26% Rénio	0 – 2300 ºC
J	Ferro / Constantan	-210 – 750 ºC
K	Cromel / Alumel	-270 – 1370 ºC
N	Nicrosil /Nisil	-270 – 1300 ºC
R	Platina / 13% Ródio-Platina	-50 – 1750 ºC
S	Platina / 10% Ródio-Platina	-50 – 1750 ºC
T	Cobre / Constantan	-270 – 400 ºC

Fonte: OMEGA, 2015.

4.2 Sensores tipo *PT100*

Os tipos de sensores *PT100* ou sensores de resistência variável funcionam com o princípio da variação da resistência devido à variação de temperatura, ou seja, variando a temperatura a resistência também se altera (OMEGA, 2015).

Os *Resistance Temperature Detector* (RTDs) podem ter termômetros do tipo metálico ou semicondutores, conhecidos como termistores. Os termômetros metálicos são constituídos por um fio ou enrolamento de metal de alto grau de pureza, geralmente sendo de cobre, platina ou níquel. A platina é um dos metais mais recomendados na sua fabricação, pois é um metal quimicamente inerte, assim mantendo suas características físicas em altas temperaturas. Os RTDs de platina se tem o nome como PT (resistência à temperatura de 0°C). Os mais comuns são o *PT-25.5, PT-100, PT-120, PT-130* e o *PT-500*, sendo o *PT-100* o mais conhecido e utilizado industrialmente(ARENY, 2003).

4.3 Termistores

Os termistores têm sua resistência elétrica alterada de acordo com a variação de temperatura, e são construídos, tipicamente, a partir de óxidos semicondutores que são sintetizados e prensados de várias formas, e são encapsulados em vidro ou epóxi. A principal característica destes dispositivos é a alta sensibilidade devido ao seu elevado coeficiente de temperatura (ARENY, 2003).

Os termistores PTC possuem um coeficiente de temperatura positivo, ou seja, seu valor de resistência aumenta com a elevação da temperatura. Por sua vez, os termistores do tipo NTC possuem um coeficiente de temperatura negativo, ou seja, seu valor de resistência diminui com o aumento da temperatura (ARENY, 2003).

5 SISTEMAS DE REFRIGERAÇÃO

Os sistemas de refrigeração têm a função de transferir continuamente energia térmica de uma região de baixa para uma região de alta temperatura através do trabalho. Nos sistemas de refrigeração a vapor ocorre a elevação da pressão e da temperatura do fluído refrigerante e este vai para o condensador expulsando o calor para o meio. O fluído vai para um recipiente no qual se expande e então retira o calor do ambiente ou do sistema que deverá ser refrigerado, completando o ciclo (SALVADOR, 1999).

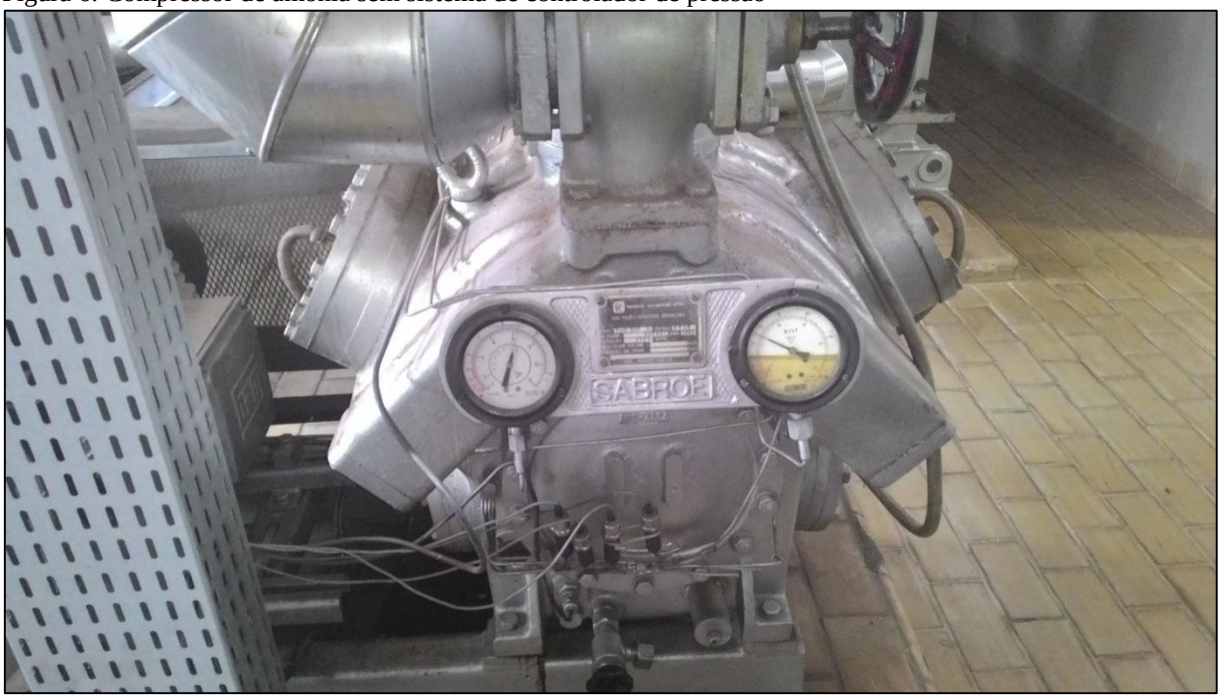
A refrigeração é utilizada desde à antiguidade. Os fluídos refrigerantes retiram o calor da superfície ou do local que deverá ser resfriado, e são escolhidos de acordo com o sistema que necessita de resfriamento, pois nenhum fluído apresenta todas as características ideais. Os fluidos refrigerantes podem ser divididos em 3 grupos de acordo com suas características. O primeiro grupo são os fluidos que realizam o resfriamento dos materiais através da absorção latente, o segundo grupo é constituído de fluidos que resfriam as substâncias através da absorção de seus calores sensíveis. Já o terceiro grupo, que se encaixa no quadro em estudo são os fluidos que são soluções que contêm vapores absorvidos de agentes liquidificáveis ou meios refrigerantes, como por exemplo a amônia (GENIÊR, et al. , 2013).

5.1 Compressores a vapor

Os compressores a vapor são elementos do sistema de refrigeração e são utilizados para elevação de pressão do fluído refrigerante na entrada da pressão de aspiração do compressor demonstrado na Figura 6. O vapor da saída da pressão de descarga vai para o condensador, assim ele perde calor na troca de fluídos, mudando do estado de vapor para líquido ou condensado. Logo após a saída, o fluido se expande no evaporador e sua pressão cai. Com isso a temperatura cai ficando em baixa pressão. O ciclo se encerra quando o compressor utiliza o vapor que sai do evaporador para mandar fluido para o condensador novamente (OLIVEIRA, 2012).

Outros componentes são necessários para que este sistema funcione perfeitamente, mas o princípio básico é a retirada de calor de um local ou de um produto, e transferência dele pelo gás refrigerante para outro lugar (OLIVEIRA, 2012).

Figura 6. Compressor de amônia sem sistema de controlador de pressão

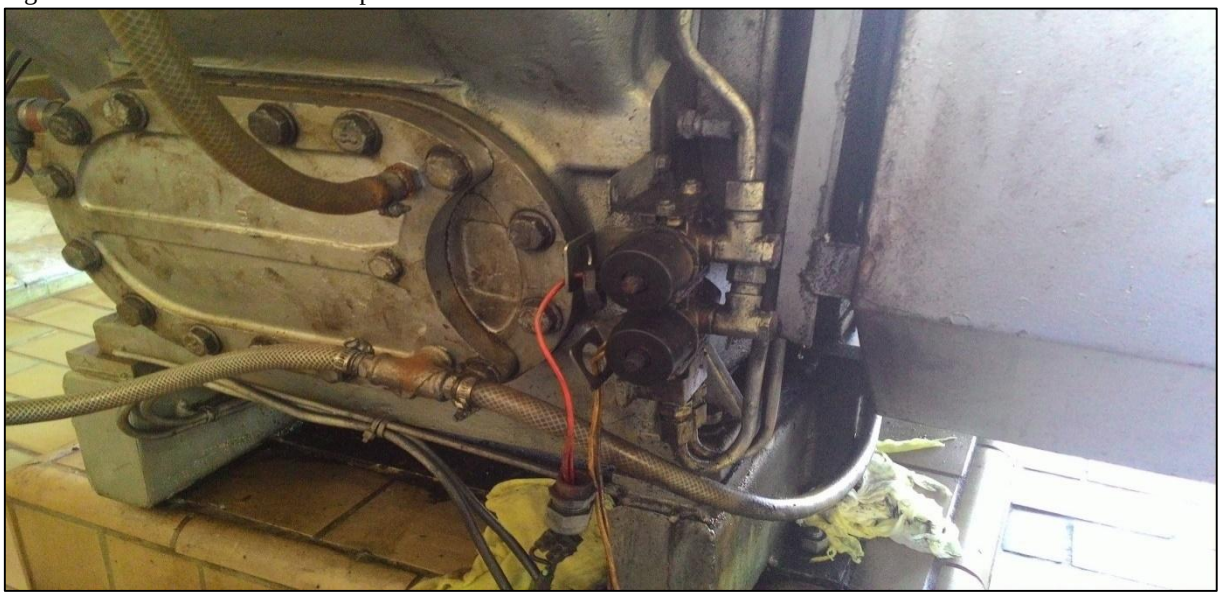


Fonte: O autor.

O compressor de amônia não apresenta controle de pressão, podendo ser medido nos estágios de 25% que é o mínimo do trabalho realizado pelo mesmo, assim tendo 50%, 75% e por fim na sua plena carga 100% (OLIVEIRA, 2012).

A pressão do compressor da empresa analisada é acionada por solenoides, que tem a função de liberar o fluído refrigerante para o pistão, com isto a pressão de descarga aumenta e a de aspiração tem uma queda e o coeficiente do compressor aumenta com a mesma porcentagem de trabalho da máquina representado na Figura 7.

Figura 7. Válvulas solenoides do pistão



Fonte: O autor.

A medição da temperatura da água da empresa deste estudo é executada através de um sensor modelo *PT100* sendo na região da saída do trocador de calor representado na Figura 8. Esta temperatura é praticamente a mesma da amônia na descarga do compressor.

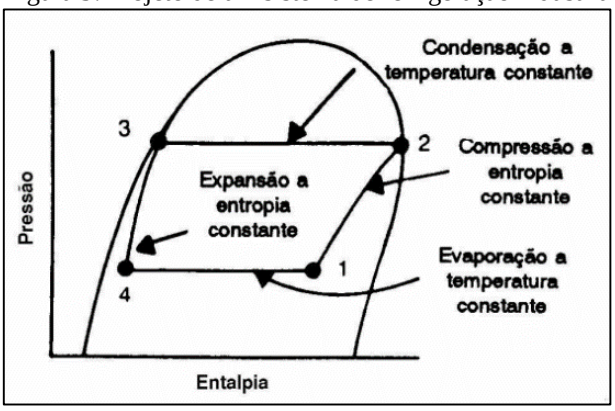
Figura 8. Trocador de calor, amônia com água

Fonte: Oautor.

5.2 Funcionamento do sistema

O diagrama de *Mollier*é uma representação gráfica da pressão absoluta e com a abcissa a entalpia (energia máxima de um sistema termodinâmico), com o ciclo representativo da refrigeração dos compressores.

Figura 9. Projeto de um sistema de refrigeração industrial



Fonte:NASCIMENTO, 2005.

Os estágios da Figura 9 demonstram como o compressor com evaporador e condensador realiza o trabalho térmico.

a) Do primeiro estágio ao segundo estágio. O compressor eleva a pressão até a pressão isobárica. Há alteração da entropia.

b) Do segundo estágio ao terceiro estágio. A pressão é constante, e permite ser alterada da entropia e condensação do fluido refrigerante.

c) Do terceiro estágio ao quarto estágio. Ocorre diminuição da pressão assim a entropia vai permanece constante.

d) Do quarto estágio ao primeiro estágio. Ocorre a evaporação do fruído refrigerante, assim e tendo a pressão de baixa constante.

Outros dois componentes essenciais são os condensadores e evaporadores. Há uma variedade de condensadores diferentes dependendo de sua aplicação (SILVA, 2005).

Os condensadores e evaporadores têm a finalidade de condensar o fluido refrigerante, com outro fluido dependendo do tipo de condensador. Os condensadores liberam a energia térmica gerada pelo compressor, assim e libera para outro lugar, ou trocada com outro fruído (SILVA, 2005).

O condensador troca calor de maneira natural, através da convecção, processo onde o fluido refrigerante que adquiriu calor circula internamente do condensador, trocando sua energia térmica com o ar de menor temperatura (SILVA, 2005).

No condensador por circulação forçada, onde o fluido refrigerante necessita de uma troca de calor mais eficiente que a natural para diminuir a temperatura mais rápido, podem-se acrescentar ventiladores para melhorar a circulação do ar com o qual será trocado calor. Há também outros tipos de condensadores por circulação forçada que vem a utilizar líquidos ao invés de ventiladores para a trocar calor com o fruído refrigerante (RUBENS, 1999).

Os evaporadores são o inverso dos condensadores em seu funcionamento. Assim como o condensador, existem diversos tipos de evaporadores com diferentes características de acordo com a situação na qual serão aplicados (SILVA, 2005).

5.3 Modelagem do comportamento do sistema

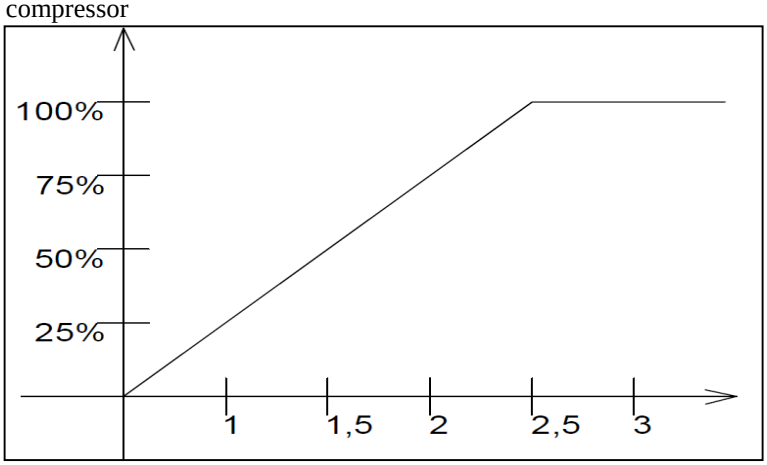
Para descrever a modelagem de um sistema é necessário descrever o seu comportamento, observando suas ações para uma análise.

As equações se relacionam e descrevem a dinâmica do sistema de forma a simplificar o seu comportamento. Para descrever mais facilmente o comportamento, é possível utilizar as saídas que se esperam ter. Se a entrada do sinal *x* tem relação com a saída *f(x)* que é a resposta da saída do sistema existe uma função é um gráfico que demonstra visualmente seu comportamento (LOPES; JAFELICE, 2005).

Devido ao fato da função *f(x)* descrever exatamente o comportamento do sistema não são indicados para sistemas não lineares, pois são difíceis de descrição por serem muito complexos (LOPES; JAFELICE, 2005).

Com a entrada *x* tendo o valor já definido é possível modelar graficamente o seu comportamento. Se for adotado a saída com a temperatura de 0°C, o compressor ficará desligado, ou se a temperatura for 2,5°C o compressor irá realizar trabalho. Com estes dados podemos desenhar um Gráfico 3.

Gráfico 3. Comportamento dos valores temperatura por estagio do compressor



Fonte: O autor.

Com estes valores de temperatura dotados se pode criar regras como:

Regra 1: se *x* é *0°C* então *f(x)* é *0%*

Regra 2: se *x* é *1°C* então *f(x)* é *25%*

Regra 3: se *x* é *1,5°C* então *f(x)* é *50%*

Regra 4: se *x* é *2°C* então *f(x)* é *75%*

Regra 5: se *x* é *2,5°C então f*(x) é *100%*

Regra 6: se *x>2,5°C então f(x)* é *100%*

A regra pode ser escrita da seguinte maneira:

Regra 1: se x está no entorno de 1°C
Então f(x) está em torno de 25%

Regra 2: se x está no entorno de 1,5°C
Então f(x) está em torno de 50%

Regra 3: se x está no entorno de 2°C
Então f(x) está em torno de 75%

Regra 4: se x está no entorno de 2,5°C
Então f(x) está em torno de 100%

A inferência *Fuzzy* e a defuzzyficação proporcionam uma forma razoável e simples de interpretação com os mesmos dados inexatos, exibindo um comportamento similar àquele descrito por matemáticos.

5.3 Modelagem do controlador *Fuzzy*

O Controlador *Fuzzy* corresponde à quantia de inferências responsáveis por calcular os valores mais satisfatórios para a saída à partir das variáveis de entradas fornecidas, considerando todos os valores de parâmetros que são utilizados como base dos conhecimentos *Fuzzy*. Assim o controlador irá determinar qual porcentagem que a máquina irá trabalhar para cada situação (PINTO, 2010).

O objetivo é estabelecer um algoritmo inteligente capaz de identificar a variação de entrada de sua temperatura e determinar uma resposta que corrija a variação. A temperatura pode atingir uma determinada posição *x* em relação à temperatura estabelecida como referência, podendo variar em escalas. O sistema de inferências *Fuzzy* irá atuar demonstrando a variação e como se fará a correção da variação de temperatura, especificando qual válvula será acionada para corrigir a variação (PINTO, 2010).

Os sistemas de inferência *Fuzzy* são modelados de forma semelhante ao raciocínio humano, tendo sempre um meio termo para cada situação. Para descrever os passos das decisões que o controlador tomaria, seria:

a) Se a temperatura está perto de 1°C, a válvula deve estar em 25%
b) Se a temperatura está perto de 1,5°C, então a válvula se abre para 50% de trabalho da máquina térmica
c) Se a temperatura está perto de 2°C, então a válvula se abre para 75% da sua capacidade de trabalho.
d) Já se a temperatura atingir superior ou 2,5°C, a máquina trabalhará na sua capacidade máxima de 100%.

Ao considerar uma variável de entrada como a temperatura *x*, o sistema irá calcular a inferência do valor da variável de *x* que corresponde à temperatura que deverá atingir. E irá estabelecer a sua condição para que atue e alcance a temperatura desejada (PINTO, 2010).

Devido à quantidade de informações e inferências que existem é necessário a utilização de ferramentas que facilitem o processo como o *jFuzzyLogic*, uma biblioteca que alicia os cálculos necessários da lógica Fuzzy (PINTO, 2010).

5.3.1 *Jfuzzylogic*

Para o desenvolvimento do controlador, é utilizado *jFuzzyLogic* junto com a ferramenta eclipse. *jFuzzyLogic* é uma biblioteca lógica *Fuzzyopen source*, que é uma implementação de

padrões da indústria para simplificar os desenvolvimentos sistemas *Fuzzy* (CINGOLANI, 2012).

jFuzzyLogic é um pacote de lógica *Fuzzy* que é escrito em Java. *jFuzzyLogic* implementa *FuzzyControl*(FC) Linguagem (FCL), e uma biblioteca bem completa que simplifica o desenvolvimento da lógica *Fuzzy*. FC é uma tecnologia que pode melhorar as capacidades da automação industrial, e é adequado para tarefas de nível de controle geralmente realizada em Controladores Programáveis (PC). Ela pode ser baseada no conhecimento e aplicação prática representada pela chamada regra linguística bases, ao invés de modelos analíticos, ou seja, podem ser empíricas ou teóricas. FC pode ser usado quando existe uma experiência que pode ser expressa no seu formalismo, permitindo ter oportunidade de conhecimento disponível para melhorar os processos e executar uma variedade de tarefas (CINGOLANI, 2012).

Outra característica vantajosa do controle *Fuzzy* é que a experiência humana pode ser incorporada de uma maneira simples. Também não é necessário modelar todo o controlador com FC: por vezes FC apenas interpola entre uma série de modelos lineares localmente, ou dinamicamente, adaptando os parâmetros de um controlador linear, tornando-a assim não linear, ou, alternativamente, apenas focando em uma determinada característica de um controlador existente que precisa ser melhorado (CINGOLANI, 2012).

A vantagem de ter uma linguagem padrão para programar sistemas *Fuzzy* é que na ausência da biblioteca lógica *Fuzzy* de uma FCL, cada implementação apresentará diferentes maneiras de definir funções de pertinência, regras e outras maneiras. Na maioria das vezes utilizar *ApplicationProgramming Interface* (APIs) que são rotinas padrão de programação, poder ser incompatíveis a partir de um sistema para outro (CINGOLANI, 2012).

Além disso, o uso de APIs para criar sistemas é complicado e propenso a erros. FCL simplesmente elimina todos estes problemas. Além disso, FCL é bastante simples, em relação ao APIs (CINGOLANI, 2012).

O *jFuzzyLogic* é escrito em Java com linguagem de programação C e Pascal. É solicitado quadro quando a intenção é compilar um programa. O código fonte é compilado para uma plataforma em um sistema operacional diferenciado. Geralmente, o próprio código fonte é desenvolvido orientado uma única plataforma. Esse código executável (binário) resultante será executado pelo sistema operacional e, no qual, ele deverá entender e dialogar com o sistema operacional em questão. Desta maneira terá um código executável para cada sistema. Por isso entende-se a necessidade de compilar uma vez para *Windows*, outra para o *Linux*, e outros programas. Já o *Java*, se baseia no entendimento de máquina virtual, onde há interação entre o sistema operacional e a aplicação, uma camada a mais responsável por “traduzir”, mas não

apenas a aplicação, faz as respectivas chamadas do sistema onde ela estará a funcionar no momento (CINGOLANI, 2012).

O aplicativo funciona sem nenhum envolvimento com o sistema. Sempre dialogando exclusivamente com *Java Virtual Machine* (JVM). Essa característica passa a ser de enorme interesse, como tudo passa pela JVM, ela pode tirar métricas, ter tomadas de decisões onde é melhor alocar a memória, etc. Uma JVM isola totalmente a aplicação do sistema operacional. Se uma JVM termina bruscamente, apenas os aplicativos que ali rodavam irão finalizar, ou seja, tal acontecimento não irá afetar outras JVMs que estejam funcionando no mesmo computador, nem afetará o sistema operacional (SILBERSCHATZ; BAER; GAGNE , 2008).

5.4 Construção das participações *Fuzzy*

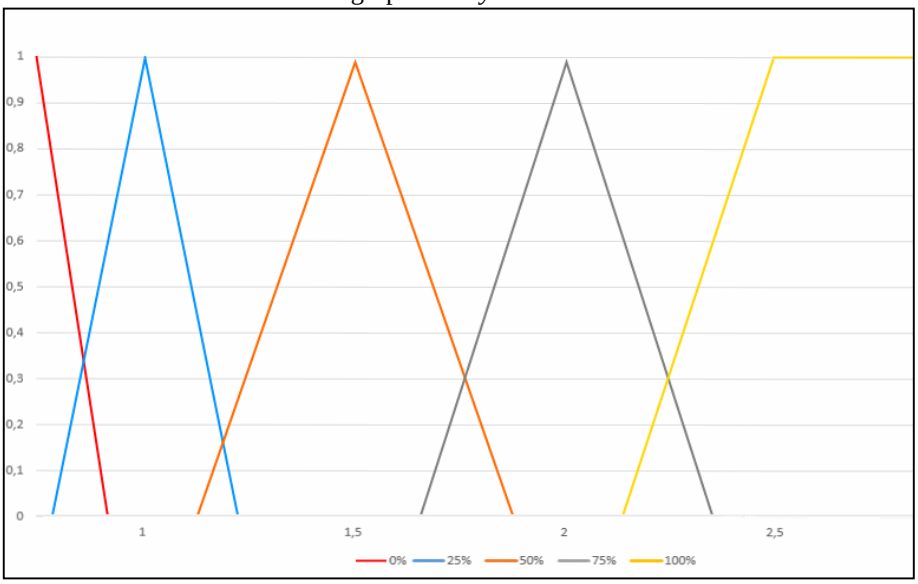
Para a identificação e construção de conjuntos nebulosos de um sistema de inferências se determina cada variável de entrada. Assim cada variável representa um intervalo de dados, que será trabalhado podendo associar um termo linguístico a cada uma delas (TONELLI NETO, 2012).

O nível de partes que o sistema do conjunto *Fuzzy* ajuda a estabelecer um nível necessário de detalhes que é de interesse em determinado ambiente de modelagem. A forma das informações divididas torna-se um importante recurso de design do modelo *Fuzzy*, contribuindo para a sua estrutura (TONELLI NETO, 2012).

Na divisão das variáveis de entrada sua participação representa o valor da temperatura, podendo calcular qual ação tomara como saída, é feita uma interpretação dos dados, que gera diversas possibilidades de resultados para cada variável de entrada.

Os conjuntos podem ser descritos em cinco partes nebulosas, onde que são 0%, 25%, 50%, 75%, 100% podendo variar no eixo horizontal referente à temperatura. No Gráfico 4 o grau de inclusão aos quais os grupos pertencem pode ser visualizado.

Gráfico 4. Grau de inclusão dos grupos *Fuzzy*



Fonte: O autor.

Como representado no Gráfico 4, o formato trapezoidal ou triangular define os intervalos de cada conjunto. Como exemplo, o conjunto 0% tem um grau pertencente inicial de zero, aproximadamente inicial no ponto aproximado de 0,75°C até o ponto 0°C onde o seu gral de pertencente aquele grupo e de 1, este grupo é definido nos pontos aproximados de {0°C, 0,75°C}. Que é um pouco diferente do grupo dos 50% que está definido por três pontos aproximados como {0,125°C, 1,5°C 1,875°C}.

As partes dos conjuntos nebulosos para cada variável de saída e entrada podem ser demonstradas resumidamente no Quadro 2:

Quadro2. Partições nebulosas

Partições nebulosas		
Variável de entrada temperatura °C	**Nomes das variáveis de entrada dos conjuntos**	**Variável de saída em porcentagem de trabalho do equipamento**
{0; 0,875}	Desligado	0%
{0,625; 1; 1,375}	BaixaEficiência	25%
{1,125; 1,5; 1,875}	MediaEficiência	50%
{1,625; 2; 2,375}	AltaEficiência	75%
{2,125; superior a 2	Máximo	100%

Fonte: O autor.

6 METODOLOGIA

Neste estudo foi realizado um estudo de caso em uma empresa localizada no município de São Gonçalo do Sapucaí-MG. Tal empresa apresenta o sistema de refrigeração a vapor por amônia. Buscando uma melhor eficácia e resposta do processo, foi implementado um sistema de controle para teste através da aplicação da lógica *Fuzzy*.

Primeiramente foi realizada uma busca na literatura sobre os temas: inteligência artificial, sistemas de refrigeração a vapor, sistemas de controle e lógica Fuzzy. Esta revisão bibliográfica foi realizada através da busca em bancos de dados na internet, artigos acadêmicos e livros sobre o assunto.

Após a realização da revisão bibliográfica sobre o tema, foi realizado um estuda na empresa em questão sobre o ambiente no qual seria aplicado o teste. Após a coleta dos dados, a simulação foi realizada através do software *JFuzzyLogic*, aplicando diversos testes sobre o sistema de controle em questão.

7 RESULTADOS E DISCUSSÃO

7.1 Simulação do controlador com *JFuzzyLogic*

O protético do simulador do controlador, as inferências e o processamento das diversas interações matemáticas são realizados pela biblioteca *open source JFuzzyLogic* que facilita e evita reescrever os conjuntos algoritmos na utilização dos conjuntos que são necessários para alcançar o objetivo do processo (TONELLI NETO, 2012).

O formato texto do arquivo FLC como demostrado na Figura 10 define as variáveis e partições *Fuzzy*, que se baseiam nas regras pertinência. Cada parte do arquivo FLC é escrito separadamente.

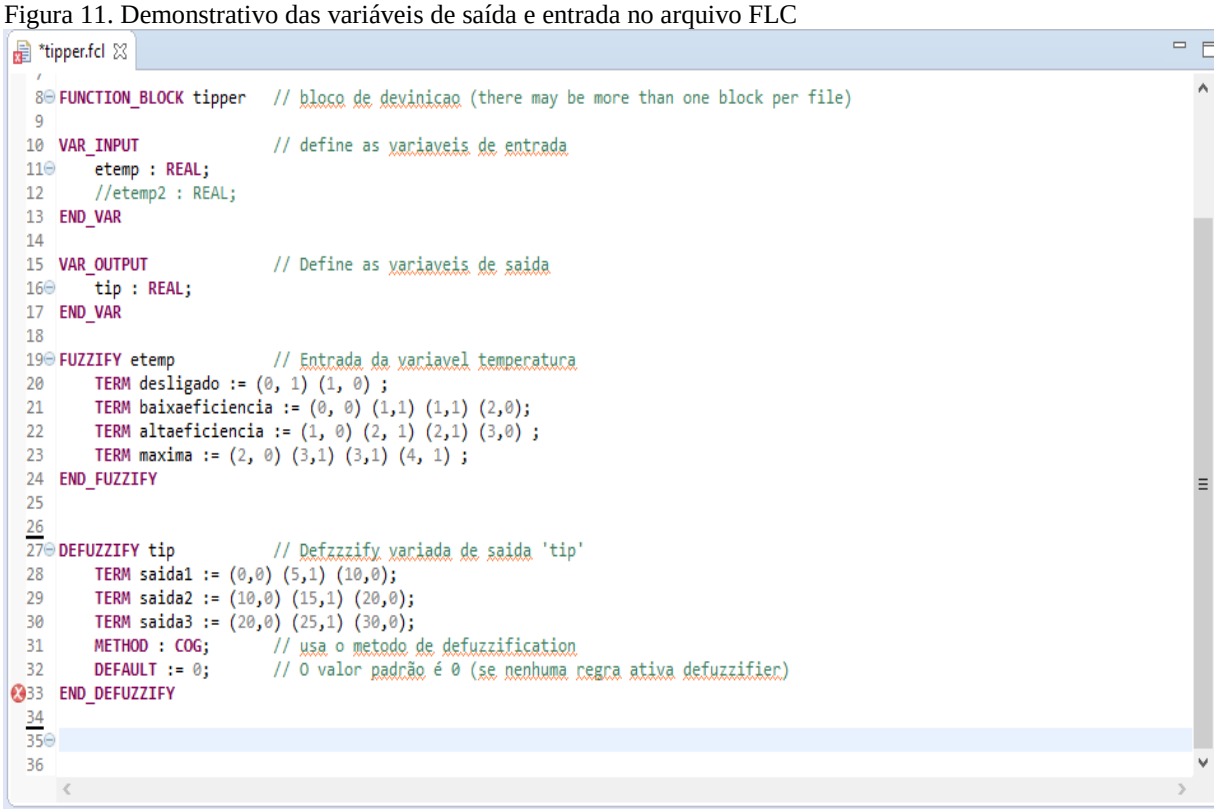
Figura 10. Formato geral de um arquivo do tipo FLC

```
FUNCTION_BLOCK simulador    // Início do bloco de definições

      VAR_INPUT               // Definição das variáveis de entrada
          nome_variavel_entrada: REAL;
      END_VAR

      VAR_OUTPUT              // Definição das variáveis de saída
          nome_variavel_saida: REAL;
      END_VAR

      FUZZIFY nome_variavel_entrada
          // definição das partições fuzzy e seus intervalos para cada variável de entrada
          TERM PARTICAO_X := (0.0, 0) (10.0, 1) (20.0, 0)
      END_FUZZIFY

      DEFUZZIFY nome_variavel_saida
          // definição das partições fuzzy e seus intervalos para cada variável de saída

          TERM PARTICAO_Y := (0.0, 0) (50.0, 1) (100.0, 0) ;
          METHOD : COG;  // Método de defuzzificação (Padrão é o Centro de Gravidade)
          DEFAULT := 0;  // Valor default caso nenhuma regra seja ativada
      END_DEFUZZIFY

      RULEBLOCK No1
          // Definição do conjunto de regras para o controlador Fuzzy. Este bloco irá descrever
          // as correlações entre as partições da variável de entrada com uma partição da variáve
          // de saída

          AND : MIN;  // Método MIN utilizado no processamento do operador lógico AND
          ACT : MIN;  // Método de ativação
          ACCU : MAX; // método de acumulação

          // Início da descrição de cada regra
          // RULE 1 : IF variavel_entrada1 IS PARTICAO1 AND variavel_entrada1 IS particao2  THEN

      END_RULEBLOCK

  END_FUNCTION_BLOCK
```

Fonte: O Autor.

No arquivo, FCL é definida como uma linguagem de controle de modo que o conceito principal é um bloco de controle, que tem algumas variáveis de entrada (VAR_INPUT) e saída (VAR_OUTPUT) de acordo com a Figura 11. O texto FCL é mais simplificado que outras linguagens de programação tradicional, não há nenhuma ordem de execução implícita, em teoria, todo o bloco é executado em paralelo (CINGOLANI, 2012).

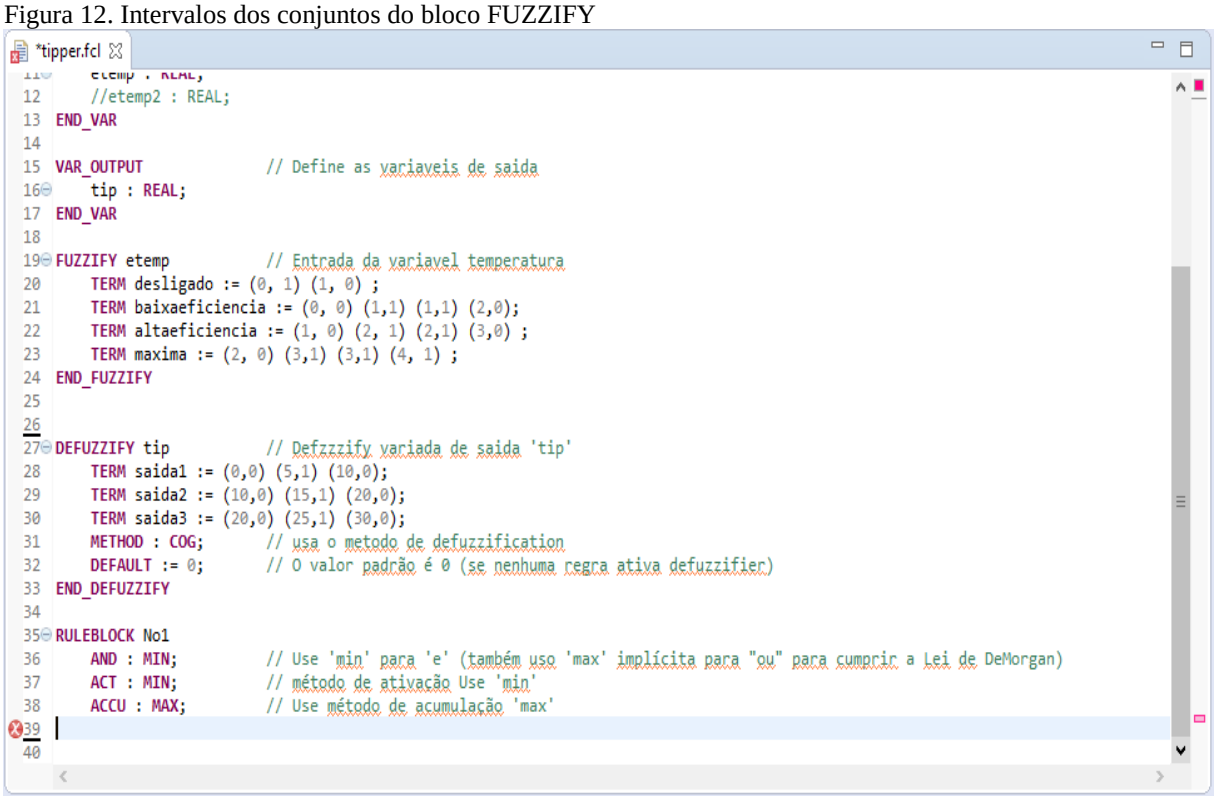
Figura 11. Demonstrativo das variáveis de saída e entrada no arquivo FLC



Fonte: O autor.

No caso do protético do controlador de temperatura, as variáveis de entrada são valores de temperatura e a variável de saída é o estado da máquina. Em seguida, definimos como cada variável de entrada será fuzzificada na definição do bloco *FUZZIFY*. Em cada bloco definimos um ou mais termos também chamados *LinguisticTerms* ou termos linguísticos. Cada termo é composto por um nome e uma função de membro. As declarações que são realizadas a partir das definições das partes para cada conjunto *Fuzzy* de cada variável são demonstradas na Figura 12 (CINGOLANI, 2012).

Figura 12. Intervalos dos conjuntos do bloco FUZZIFY



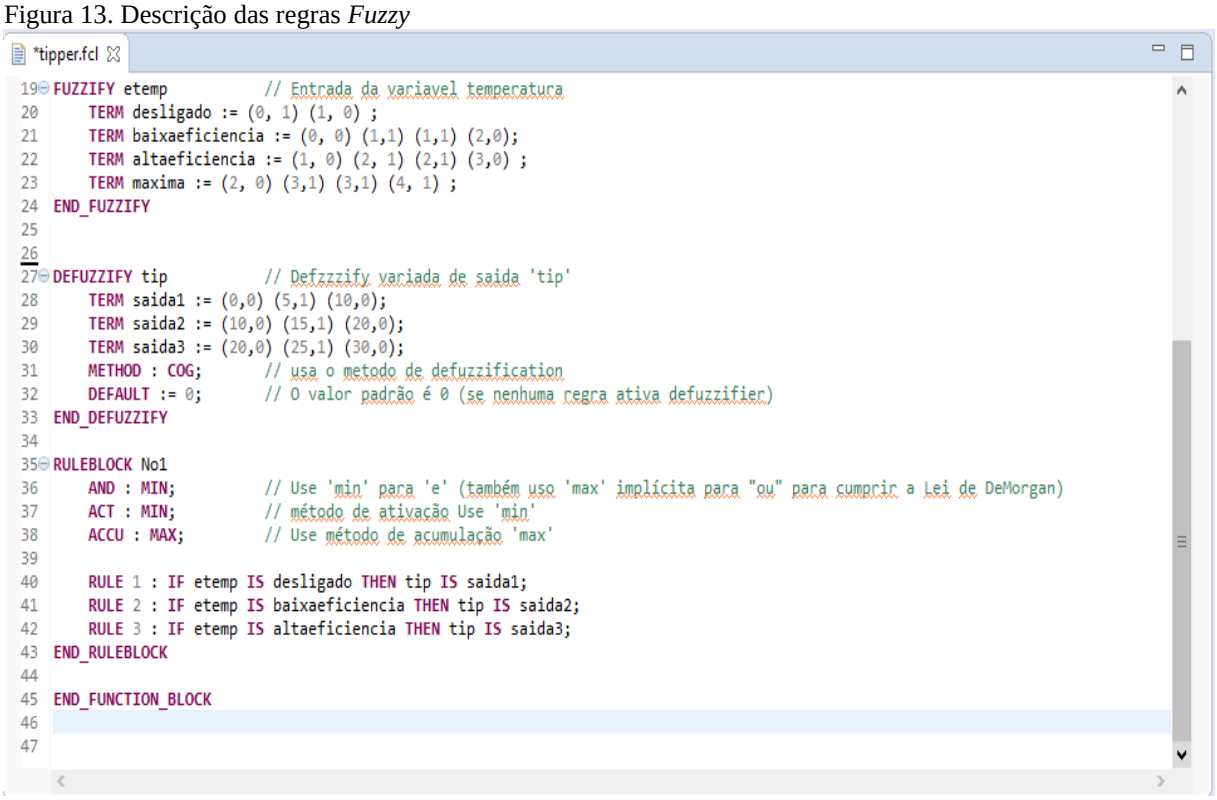
Fonte: O autor.

No mesmo bloco FUZZIFY se aplica o comando COG – *Center of Gravity* ou centro de gravidade, como método defuzzificação que está sendo aplicado. Existem outros métodos diferentes do COG para defuzzificação, mas a vantagem do COG em relação aos outros e que ele atende às necessidades da maioria dos casos.

Na declaração das regras *Fuzzy* são descritas situações a serem analisadas especificamente, onde acontece a análise da variável de saída e entrada com o conjunto do bloco *FUZZIFY*. No bloco *FUZZIFY* é declarado um valor para ser utilizado caso nenhuma regra seja aplicada. Normalmente adota-se o valor zero no comando *DEFAULT*, que ocorre quando nenhuma regra é aplicada (CINGOLANI, 2012).

A necessidade de se especificar o método que será utilizado para o processamento das regras como MIN para ativação que define a variável de entrada afetando a saída e MAX para acumulação que define quanto de fator de ponderação será aplicada. Na declaração MIN e MAX utilizam-se operadores lógicos AND para MIN e OR para MAX como demonstrado nas regras na Figura 13 (CINGOLANI, 2012).

Figura 13. Descrição das regras *Fuzzy*



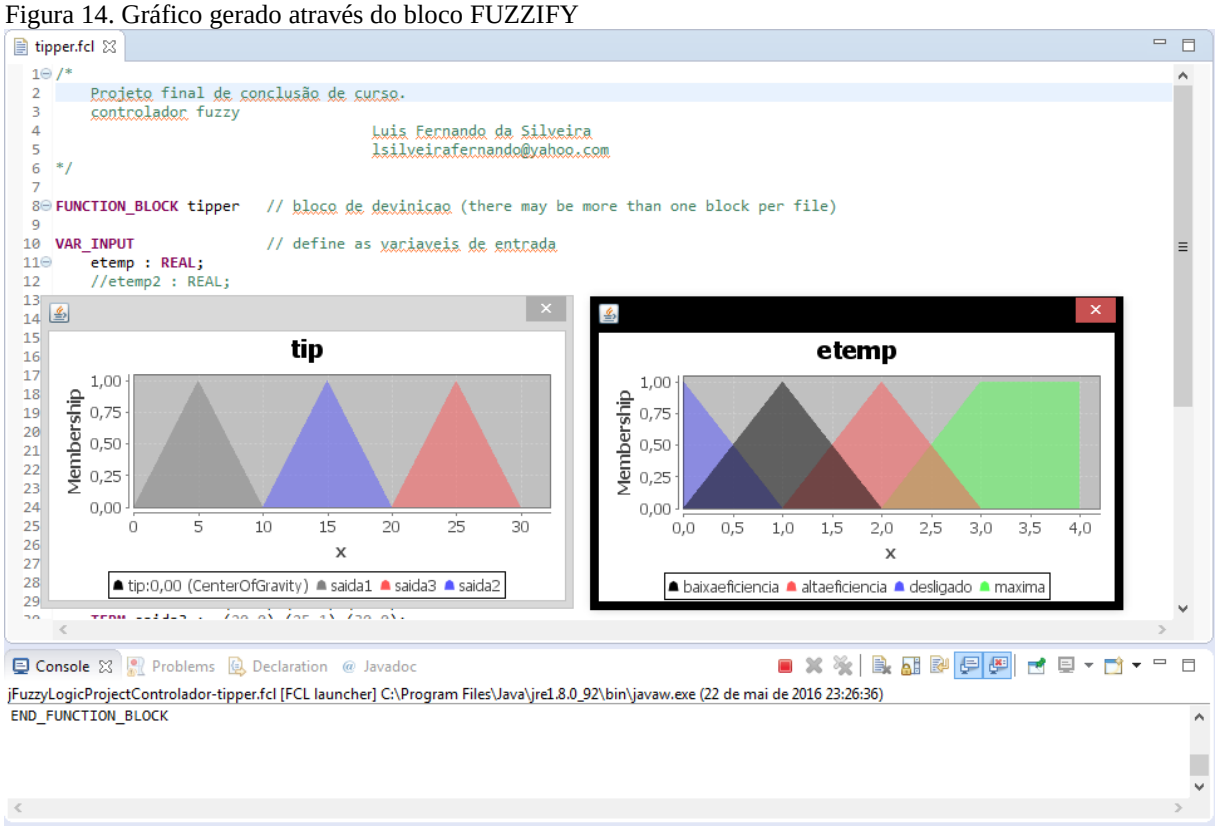
```
*tipper.fcl
19 FUZZIFY etemp          // Entrada da variavel temperatura
20     TERM desligado := (0, 1) (1, 0) ;
21     TERM baixaeficiencia := (0, 0) (1,1) (1,1) (2,0);
22     TERM altaeficiencia := (1, 0) (2, 1) (2,1) (3,0) ;
23     TERM maxima := (2, 0) (3,1) (3,1) (4, 1) ;
24 END_FUZZIFY
25
26
27 DEFUZZIFY tip          // Defzzzify variada de saida 'tip'
28     TERM saida1 := (0,0) (5,1) (10,0);
29     TERM saida2 := (10,0) (15,1) (20,0);
30     TERM saida3 := (20,0) (25,1) (30,0);
31     METHOD : COG;       // usa o metodo de defuzzification
32     DEFAULT := 0;       // O valor padrão é 0 (se nenhuma regra ativa defuzzifier)
33 END_DEFUZZIFY
34
35 RULEBLOCK No1
36     AND : MIN;          // Use 'min' para 'e' (também uso 'max' implícita para "ou" para cumprir a Lei de DeMorgan)
37     ACT : MIN;          // método de ativação Use 'min'
38     ACCU : MAX;         // Use método de acumulação 'max'
39
40     RULE 1 : IF etemp IS desligado THEN tip IS saida1;
41     RULE 2 : IF etemp IS baixaeficiencia THEN tip IS saida2;
42     RULE 3 : IF etemp IS altaeficiencia THEN tip IS saida3;
43 END_RULEBLOCK
44
45 END_FUNCTION_BLOCK
46
47
```

Fonte: O autor.

O arquivo tipper.fcl permite utilizar o comando java –jar jFuzzyLogic.jar tipper.fcl. Este comando irá analisar e carregar o código FCL e mostrar as funções de pertinência. Se três tipos são utilizados, *food, service e tip*, são gerados três gráficos automaticamente que facilitam a interpretação do comportamento definindo qual variável serviço será fuzzidificada.

O gráfico gerado *service* define como a variável serviço será fuzzidificada, onde os valores são obtidos através dos conjuntos do bloco *FUZZIFY*, demonstrado na Figura14, a fim de deixar mais fácil a visualização foram modificados os valores para números inteiros (CINGOLANI, 2012).

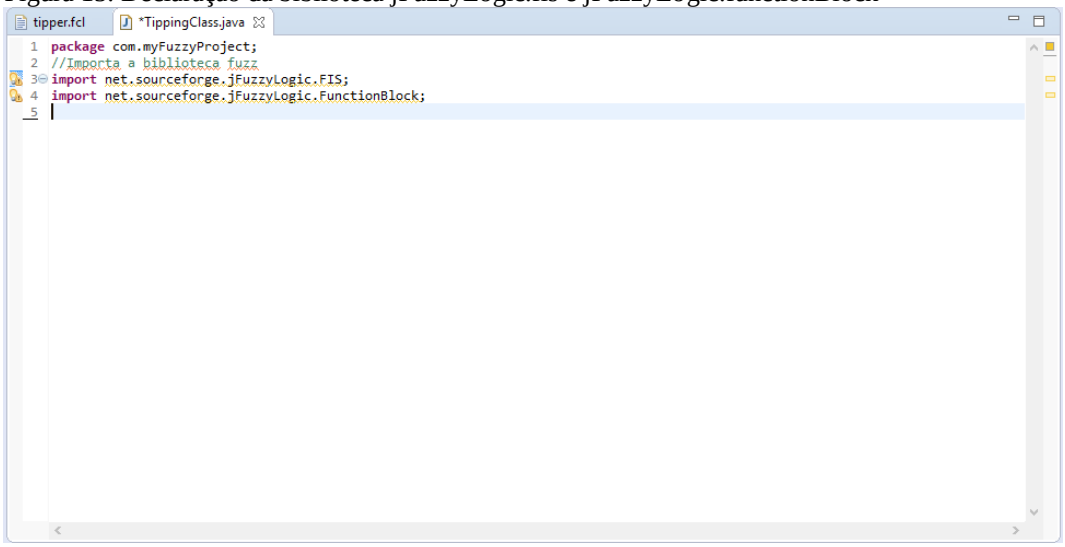
Figura 14. Gráfico gerado através do bloco FUZZIFY



Fonte: O autor.

A classe *Java* permite que o sistema *Fuzzy* seja rapidamente projetado, importando as bibliotecas que utilizadas conforme pode ser observado na Figura 15. A biblioteca possui uma interface de fácil utilização, através da função *import*.

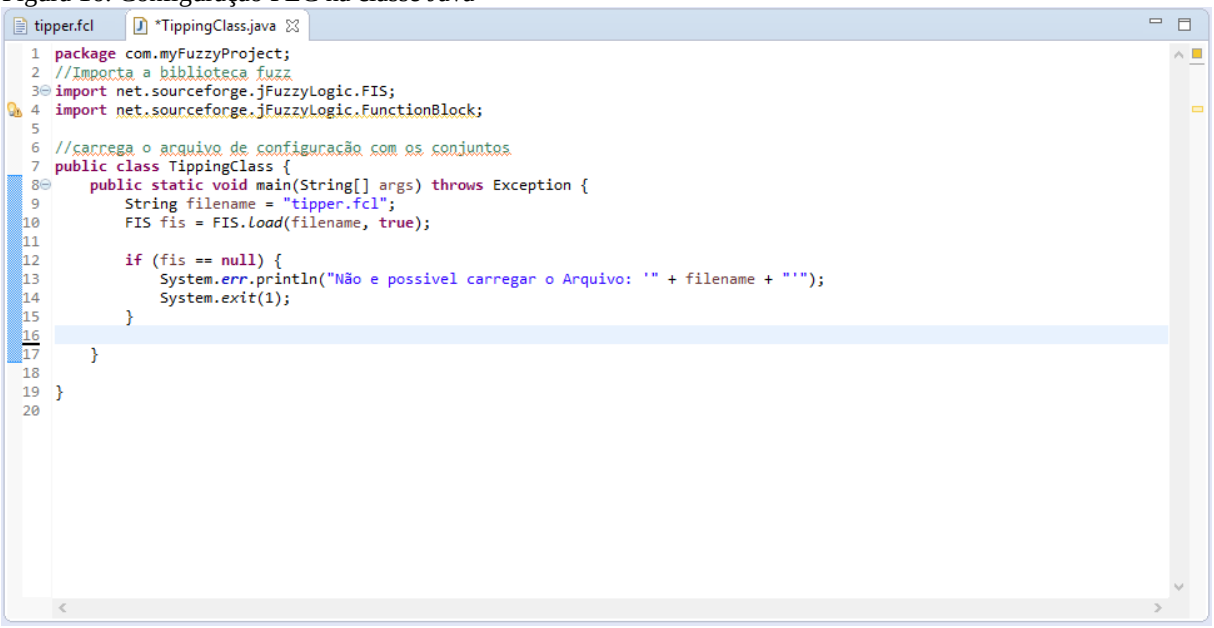
Figura 15. Declaração da biblioteca jFuzzyLogic.fis e jFuzzyLogic.functionBlock



Fonte: O autor.

Carregamos o arquivo de configuração FLC com as configurações da lógica *Fuzzy*, fazendo atratativo de um possível erro senão carregado o arquivo demonstrado na Figura 16.

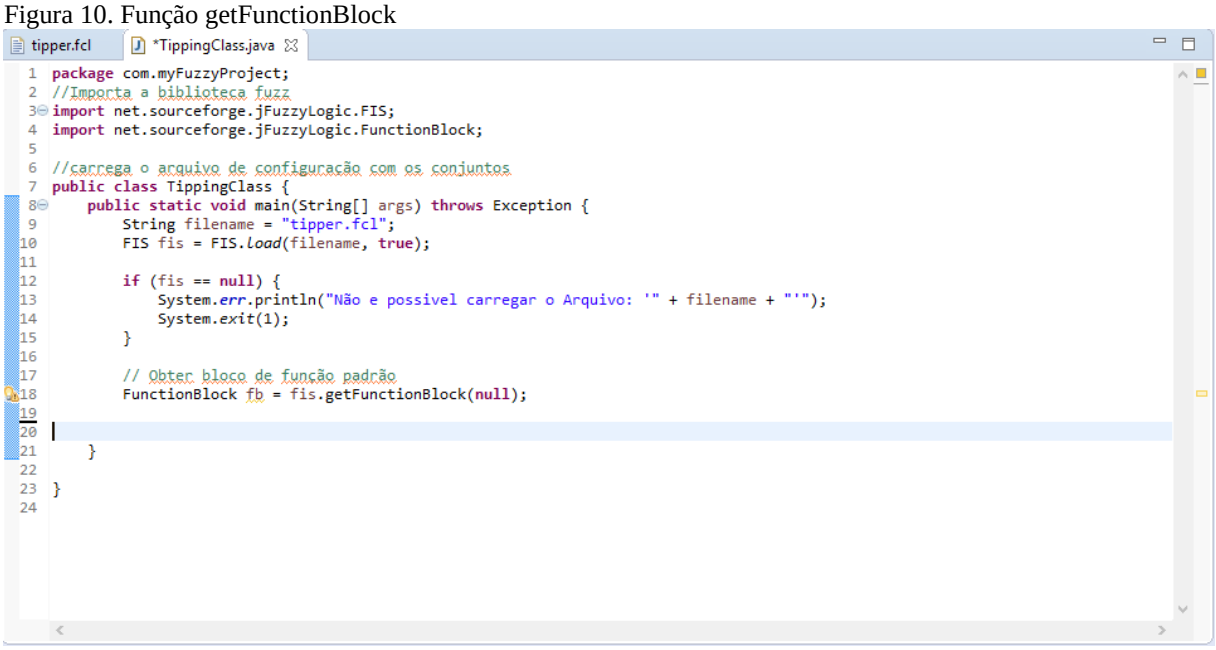
Figura 16. Configuração FLC na classe Java



Fonte: O autor.

Carregamos o bloco de função do arquivo de configuração com a função *getFunctionBlock* na Figura 17 para o cálculo das inferências com o conhecimento que foi alimentado no arquivo definido FLC (CINGOLANI, 2012).

Figura 10. Função getFunctionBlock



Fonte: O autor.

A declaração do código *fb.setvarible* permite que o valor da variável de entrada que vai interagir com o arquivo FLC seja inserido no cálculo, a fim de teste. A variável que irá receber corresponde a temperatura do controlador para cálculo a saída. Utilizamos a função *evolute* disponível na biblioteca *Fuzzy* para avaliar o valor da variável de entrada em relação aos conjuntos definidos no arquivo de configuração FLC demonstrado na Figura 18 (CINGOLANI, 2012).

Figura 18. Declaração do código *fb.setvarible* e *fb.evaluate*

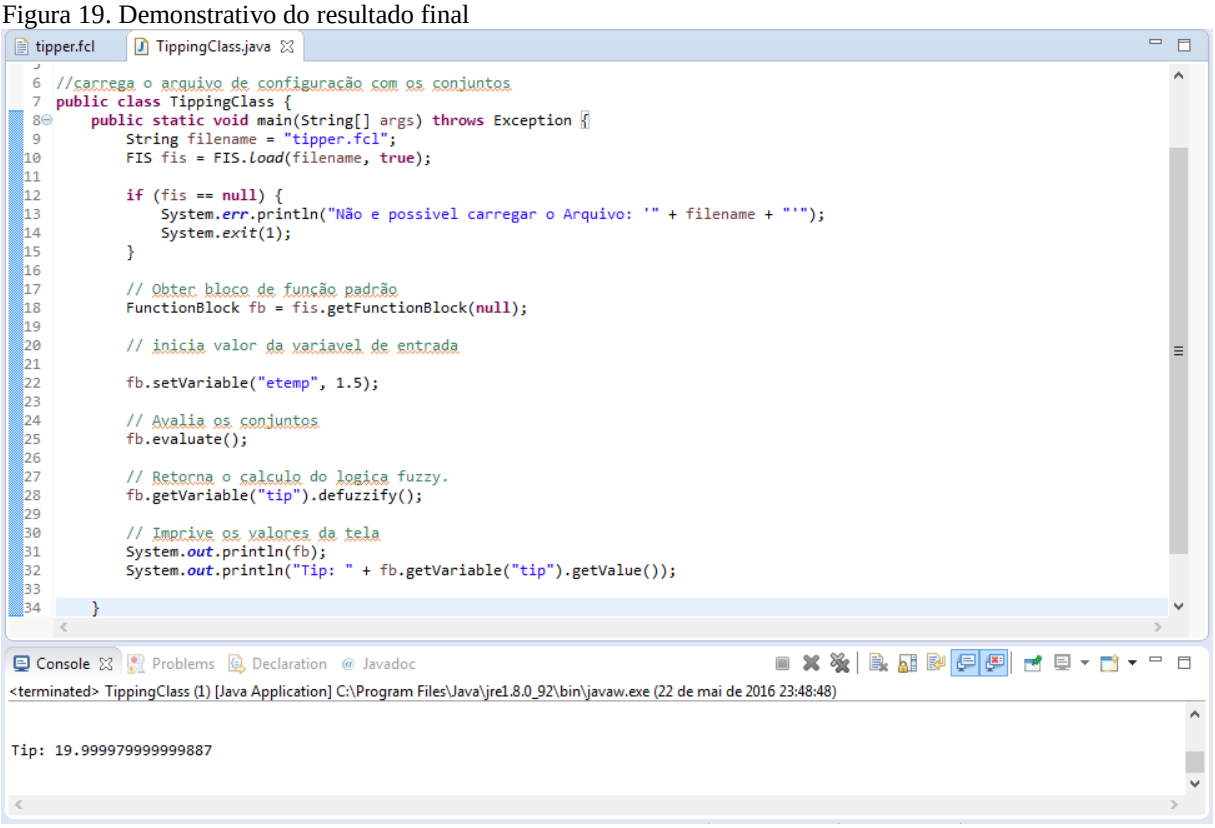
tipper.fcl | *TippingClass.java

```
import net.sourceforge.jFuzzyLogic.FIS;
import net.sourceforge.jFuzzyLogic.FunctionBlock;

//carrega o arquivo de configuração com os conjuntos
public class TippingClass {
    public static void main(String[] args) throws Exception {
        String filename = "tipper.fcl";
        FIS fis = FIS.load(filename, true);

        if (fis == null) {
            System.err.println("Não e possivel carregar o Arquivo: '" + filename + "'");
            System.exit(1);
        }

        // Obter bloco de função padrão
        FunctionBlock fb = fis.getFunctionBlock(null);

        // inicia valor da variavel de entrada

        fb.setVariable("etemp", 1.5);

        // Avalia os conjuntos
        fb.evaluate();


    }

}
```

Fonte: O autor.

Utilizamos a fução *Defuzzify* para processar e retornar o valor final do cálculo de acordo com os conjustos pre definidos no arquivo de configuração FLC.Por fim, utilizamos o a função *java System.out.println* para exibir o valor final do cálculo pelo console *Java*na Figura19. Por fim o resultado final no console java do cálculo *Fuzzy*.

Figura 19. Demonstrativo do resultado final



Fonte: O autor.

Para fim de teste a biblioteca *JFuzzyLogic* simplifica-se os cálculos, o que é suficiente para definir o resultado final onde não foram adotados valores reais para os conjuntos de entrada e saída. A modelagem do algoritmo foi definida pelo arquivo FLC, a função apenas o código Java para fornecer e atualizar valores para o código do controlador (CINGOLANI, 2012).

Futuramente pode ser feito implementação do controlador utilizando uma placa Arduino, utilizando sensor para o valor de entrada do controlador determinando o valor de temperatura. O processo de teste é importante para não ter incerteza e excluir possíveis falhas, antes de incorporar um *hardware* ou *software* para uso físico.

7.2 Ambiente de teste

Para criar o ambiente de testes temos três critérios de cálculo que são possíveis para utilização da função de defuzzyficação. O primeiro é o Critério do Máximo, neste processo é escolhido o valor (Y) do conjunto *fuzzy* para o qual a função de pertinência (μ(y)) é máxima. O segundo critério de cálculo é a Média do Máximo, neste processo são escolhidos os valores (y) do conjunto *fuzzy* de saída para o qual a função de pertinência (μ(y)) é máxima, assim a saída

corresponde à média aritmética destes valores. Por último, têm-se o critério Centro de Área, este método determina o centro da gravidade (y), o centro de área do conjunto *fuzzy* de saída, sendo este valor utilizado como resultado final.

O centro de área pode ser definido pela função:

$$y = \frac{\sum_{i=1}^{n} \mu iui}{\sum_{i=1}^{n} \mu i}$$

Onde:

n: número de regras

μi: grau de pertinência da regra i

ui: ação de controle recomendada referente a regra i.

O valor de *ui* é considerado sendo o valor da projeção sobre a abscissa, correspondente ao valor máximo do conjunto (μi = 1).

A base de regras descreve o comportamento e a tomada de ação do controlador. A inferência manipula a base de regras que são utilizadas e interpretadas de modo que os conjuntos sejam mapeados no controlador. Os valores de pertinência são definidos no intervalo de 0 a 1.

Descrito pelas equações:

$$\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x) : max\{\mu_A(x), \mu_B(x)\}$$

Onde:

∨: operador *or*

$$\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) : min\{\mu_A(x), \mu_B(x)\}$$

Onde:

∧: operador *and*

$$\mu -_A (x) = 1 - \mu_A(x)$$

O complemento do conjunto *fuzzy*.

Todos os cálculos realizados foram feitos no arquivo FCL da biblioteca *JFuzzyLogic*.

Para o ambiente de teste foi aplicado o critério de área para o cálculo de saída do controlador, por ser o critério mais utilizado. As temperaturas de entrada utilizadas no teste foram definidas entre 0.2° C aos 3° C, para obtenção dos resultados para análise.

Foram realizados um total de seis testes no controlador para verificar o seu comportamento.

No primeiro teste o controlador foi submetido a temperatura de 0.2°C onde apresentou a saída de 7.72 enquadrando-se no conjunto de saída um entre 0 e 10.

No segundo teste o controlador foi submetido a temperatura de 0.5° C onde apresentou a saída de 10 enquadrando-se também no conjunto de saída um entre 0 a 10.

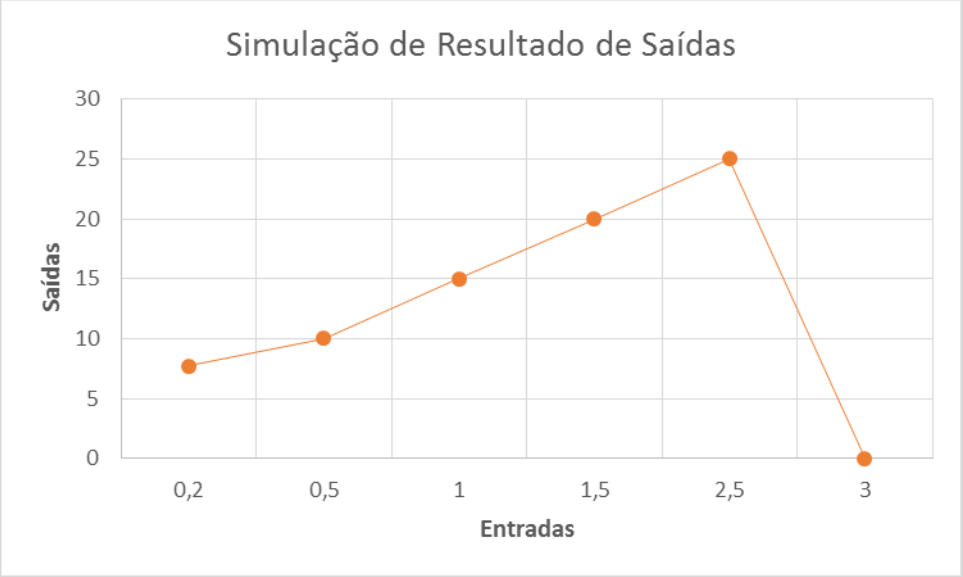
Já no terceiro teste o controlador foi submetido a temperatura de 1°C onde apresentou a saída de 15 enquadrando-se no conjunto de saída dois entre 10 a 20.

Na realização do quarto teste o controlador foi submetido a temperatura de 1.5°C onde apresentou a saída de 19.99 enquadrando-se no conjunto de saída dois entre 10 a 20.

Durante o quinto teste o controlador foi submetido a temperatura de 2.5°C onde apresentou a saída de 24.99 enquadrando-se no conjunto de saída três de 20 a 30.

No sexto e último teste o controlador foi submetido a temperatura de 3°C onde apresentou a saída de 0, devido ao fato do resultado obtido não se enquadrar em nenhum conjunto ele passa a ser tratado como regra definida para passar a valer 0 conforme pode ser demonstrado no Gráfico 5.

Gráfico 5: resultados gerados para saídas.



Fonte: O autor

Definimos como regra para o estouro de temperatura o valor 0se não contiver nos grupos. Vale ressaltar que os grupos de saída são relacionados ao comportamento do estágio do equipamento.

Assim, as funções e conjuntos estão relacionados normalmente ao objetivo de funcionamento do equipamento. Uma grande vantagem em usar uma lógica como controle em relação às outras formas de controle é que estas apresentam um alto desempenho computacional e são desenvolvidas com técnicas mais complexas. Portanto, quando houver a necessidade de manter o sistema de controle operando dentro dos limites operacionais, é adequado a utilização dessa forma lógica de controle.

8 CONCLUSÃO

Dentre os diversos meios de correção de problemas em sistemas de pressão, este estudo focou na utilização de métodos de programação matemática, o que melhora o tratamento dos problemas de otimização quanto ao funcionamento do sistema.

Para o sistema da empresa situada no município de São Gonçalo do Sapucaí/MG, têm-se a necessidade de manter um sistema de controle operando de forma a respeitar os limites sugeridos ao equipamento e com segurança.

Para alcançar os objetivos básicos de controle: otimização e segurança, empregou-se a lógica *Fuzzy* como recurso para os cálculos matemáticos complexos. Através das definições de conjuntos relacionados ao funcionamento do controlador em estudo. Desta forma foi possível a implementação básica do sistema para diagnosticar o comportamento e classificar os distúrbios de temperatura.

A utilização da biblioteca *JFuzzLogic* em conjunto com a linguagem de programação *Java* permitiram o desenvolvimento mais amigável do controlador. Possibilitando a utilização de cálculos matemáticos complexos de forma a interpretar as entradas do controlador.

Em implementação básica em teste conseguimos obter o resultado esperado de saída através da logica *Fuzzy*, definindo os parâmetros de entrada da temperatura. Aplicando os métodos e funções disponíveis na biblioteca de estudo, e estruturando as regras de decisões conseguimos um resultado estável final.

Os resultados obtidos abrem uma gama de possibilidades de aplicações do sistema desenvolvido para aplicações diversas, dada a flexibilidade das ferramentas utilizadas do desenvolvimento no controle de temperaturas, podendo ampliar a portabilidade do controle a distância.

REFERÊNCIAS

ARENY, P. R. **Sensores y Acondicionadores de señal.** 4 . ed. Barcelona:MercomboBoixareu Editores, 2003.

AZEVEDO, F. **Otimização de Rede de Distribuição de Energia Elétrica Subterrânea Reticulada através de Algoritmos Genéticos.** 140 f. Dissertação Mestrado, Curitiba: Universidade Federal do Paraná, 2010.

BATISTA, O. E. et al. **Abordagem possibilísticaFuzzy para localização de faltas em alimentadores com geração distribuída. ln**: Simpósio Brasileiro de Automação InteligenteXII SBAI. Fortaleza, 2013.

CAMBOIM, W.L.L, **Aplicação de técnicas Fuzzy no controle de pressão em sistemas de abastecimento de água**. Dissertação de Mestrado UFPB, João Pessoa: 2008.

CASTRO, R. E. **Otimização de estruturas com multi-objetivos via algoritmos genéticos.** 206 f. Doutorado em ciências em engenharia civil, Universidade federal do rio de janeiro. Agosto, 2011.

CINGOLANI, P. **jFuzzyLogic: A Robust and Flexible Fuzzy-Logic Inference System Language Implementation.** WCCI 2012 IEEE World Congresson Computational Intelligence. Australia, 2012.

GENIÊR, F. S. **Ciclos de refrigeração: conceitos e estudos de eficiência.** Alegre: Universidade Federal do Espiríto Santo, 2013.

LOPES, W. A.; JAFELICE, R. S. da M**. Modelagem Fuzzy de Diagnóstico Médico e Monitoramento do Tratamento da Pneumonia.** 2005. 175f. Disponível em: <http://www.ime.unicamp.br/~biomat/bio15final.pdf#page=82> Acesso em: 11outubro 2015.

NOBREGA SOBRINHO, C. A. **Controlador neural aplicado a um sistema posicional acionado por motores de indução trifásicos**. 106 f. Dissertação de Mestrado UFPB, João pessoa: Universidade Federal da Paraíba, 2011.

OGATA, K. **Engenharia de controle moderno.** 5. ed. São Paulo: Companion Website, 2010.

OLIVEIRA, A. P. **Eficiência energética aplicadas a sistemas de refrigeração,** 2012. Disponível em: <http://www.joinville.udesc.br/portal/professores/sergiovgo/materiais/Adriano_Pires_Trab_ET3EFE_2012_1.pdf>. Acesso em: 22 Setembro 2015.

OMEGA. **Manual do fabricante**. Disponível em: <http://www.omega.com> Acesso em: 04 outubro de 2015.

PINTO, R. L**. Aplicação de um sistema especialista fuzzy para redução de manobras de dispositivos shunts chaveados automaticamente por um 93 compensador estático**. 2010. 116f. Dissertação Mestrado. Universidade Federal do Rio de

Janeiro, 2010.Disponível em: <http://objdig.ufrj.br/60/teses/coppe_m/RaphaelLemosPinto.pdf> Acesso em: 05 outubro. 2015.

RICH, E.; KNIGHT, K.. **Inteligência artificial.** 2. ed. São Paulo: Makron Books, 1993.

RUBENS A. D. **Impacto da substituição de equipamentos na conservação de energia,** 1999. Disponível em: <http://www.abcm.org.br/app/webroot/anais/cobem/1999/pdf/AAAAID.pdf>.1999 Acesso em: 05 outubro. 2015.

SALVADOR, Francisco. **Projeto de um sistema de refrigeração industrial com "set-point" variável.** São Paulo: Universidade de São Paulo, 1999.

SANTOS, Germano J. C. **Lógica Fuzzy**. Ilhéus: Universidade Estadual de Santa Cruz, 2003.

SILBERSCHATZ, A.; BAER, P.G.; GAGNE, G. **Sistemas Operacionais com java**. 2. ed. São Paulo: Elsevier, 2008.

SILVA, M. N. da. **Eficiência energética em sistema de refrigeração industrial e comercial.** 2005, Disponível em: <http://www.mariloureiro.net/tecnica/refrigeracao/EficiEnergSist.Refri.IndustrialCom.pdf >. Acesso em: 25 Setembro 2015.

TONELLI NETO, M. de S. **Formulação do controle preventivo em sistemas de distribuição de energia elétrica baseada na lógica fuzzy e redes neurais**. 137 f. Solteira, 2012. Disponível em: <http://base.repositorio.unesp.br/bitstream/handle/11449/87143/tonellineto_ms_me_ilha.pdf?sequence=1&isAllowed=y> Acesso em: 10 janeiro. 2016.