

Antônio Augusto Cavalcante Leite
Ênio Salgado Pereira

Controle de Processo Térmico Multivariável visando a Racionalização de Energia

Trabalho de graduação apresentado à disciplina Projeto Final 2, como requisito parcial à conclusão do curso de Engenharia Mecatrônica. Laboratório de Visão, Automação e Sistemas Inteligentes, Departamento de Engenharia Elétrica, Faculdade de Tecnologia, Universidade de Brasília.

Orientador: Prof. Dr. Adolfo Bauchspiess

Brasília 2003

Agradecimentos

À minha mãe, Ana Júlia, por seu amor incondicional e apoio em todas as fases de minha vida.

Ao meu pai e irmãos pela amizade, apoio e carinho

Ao professor orientador Adolfo Bauchspiess, pela disponibilidade, paciência e compreensão dispensados.

Ao colegas, Ênio Salgado, Alexandre Silva e Leandro Matos, pela união na busca do sucesso deste projeto

À toda minha família, em especial a minha mãe, Dona Gelsa, por todo apoio ao longo dessa grande jornada, e de toda minha vida.

Aos colegas, Antônio Augusto, Leandro e Alexandre, sempre presentes.

Ao professor Adolfo Bauchspiess, por todo conhecimento passado nesse ano de trabalho.

A Carol por todo seu amor, carinho e paciência,

E a todos que contribuíram direta ou indiretamente para que eu chegasse onde estou hoje

Resumo

Existe uma grande preocupação no que diz respeito ao consumo excessivo de energia elétrica. Dentro deste contexto, estudos têm sido feitos no intuito de tornar mais eficiente o uso deste tipo de energia. Geralmente, sistemas térmicos são os que representam a parcela maior do consumo energético. Condicionadores e aquecedores de ar, chuveiros entre outros aparelhos elétricos/eletrônicos são responsáveis por grande parte do consumo de energia elétrica em edifícios, empresas e residências.

Neste trabalho, estuda-se uma forma de otimizar o desempenho de energia em sistemas térmicos. Este estudo é parte de um objetivo maior, tornar mais eficiente o consumo de energia elétrica em sistemas condicionadores de ar, sem desconsiderar o conforto. Este objetivo deverá ser alcançado em médio prazo, pois antes de tudo é necessário conhecer melhor o funcionamento dos próprios sistemas térmicos envolvidos no processo.

A fase atual deste estudo está focada no consumo de energia de um sistema térmico que simula um sistema de ar condicionado e no acesso remoto do experimento via Internet.

Para representar um sistema de ar condicionado, uma maquete, representando um escritório típico, é confeccionada. Neste sistema, o processo inverso do resfriamento de ar é analisado e secadores de cabelo realizam a função de fontes de calor.

O controle é feito pelo uso de microcontroladores PIC 16F877A da MICROCHIPTM e a medida da temperatura é tomada por sensores do tipo LM35 e com o auxílio do LabVIEW.

Para os próximos alunos ficará a missão de conhecer a fundo os aparelhos de ar condicionado, e de implementar controles para os aparelhos. O presente trabalho servirá como base e referência para os próximos trabalhos. Assim não será necessário repetir todo este estudo inicial, possibilitando um desenvolvimento mais rápido e objetivo do sistema final.

O produto final do presente trabalho e dos subseqüentes é desenvolver um equipamento *plug-and-play* que possa ser ligado, externamente, no ar condicionado e, assim, possibilitar otimização e racionalização no consumo de energia, unindo economia a conforto. A esta técnica dá-se o nome de *retrofitting*.

Existem também perspectivas no sentido de utilizar o sistema em estudo como laboratório remoto para auxiliar no ensino em disciplinas voltadas à área de controle de processos.

Abstract

Lately, many firms are concerned with the excessive demanded of electrical energy. Air conditioning systems are responsible for a great amount of this demand. A way of optimization of the air conditioning system work is studied in this project, considering also comfort. First, it is necessary to study the behavior of thermal systems.

The present phase of this study is focused on electrical energy demand by a thermal system, which represents an air conditioning system. In this case, experimental simulations with different types of control are done. As a result, a verification of which type of control is more efficient in terms of economy and simplicity can be verified.

In order to represent an air conditioning system correctly, a mockup, which represents a typical office, is made. In this system, the inverse of the process of cooling is analyzed and hair dryers are used as heat sources.

Microcontrollers PIC 16F877 of MICROCHIPTM are used to implement the control logic, LM35 sensors and LabVIEW are used to measure the temperature of the system.

Hereafter, studies of the air conditioner device itself can be made by other students in order to project a control system for it. This project can be used as a reference for next studies. Then, it will not be necessary all this initial study of the thermal system.

The result of this project and future ones is to develop a plug-and-play device which can be turned on the outside of the air conditioning device and then optimize the electrical energy expenditure, considering economy and comfort. This technique is called retrofitting. Studies are being held in order to use the system as a remote laboratory, to help the instruction of control theory to control students.

Índice

Agradecimentos	2
Resumo	3
1 Introdução	9
1.1 Métodos para modelagem de sistemas físicos	9
1.2 Justificativa.....	10
2 Fundamentos Teóricos	12
2.1 Modelamento matemático de sistemas térmicos	12
2.1.1 Transferência e fluxo de calor	13
2.1.2 Variáveis Térmicas	14
2.2 Distribuição de Ar no interior de uma sala.....	15
2.4 Sistemas Inteligentes	17
2.4.1 Lógica Fuzzy	18
2.5 Controle por PWM	20
2.6 Microcontroladores.....	21
2.6.1 Família PIC (Programmable Integrated Controller)	21
2.7 Optoacopladores	22
3 Metodologia	23
3.1 Sistema Térmico em Estudo – Maquete.....	23
3.2 Modelamento Matemático do Sistema	27
3.2.1 Parâmetros Concentrados	27
3.3 Microcontrolador Utilizado	29
3.3.1 PIC16F877A.....	29
3.4 Procedimentos para o uso do PIC16F877A e MPLAB	31
3.5 Circuito para gravação do PIC	31
3.6 Medição da Temperatura.....	32
3.6.1 Sensor Utilizado	32
3.6.2 Leitura da Temperatura pelo Microcontrolador	33
3.6.3 Ajustes do parâmetros do PIC para medição.....	35
3.7 O LabVIEW (Laboratory Virtual Instrument Engineering Workbench)	36
3.8 Driver de potência projetado	38
3.9 Programa para teste do Módulo PWM do PIC e driver de Potência	39
3.10 Algoritmo do controlador Fuzzy	40
• Funções de interface de fuzzificação.....	40
• Base de Conhecimento e Lógica de Tomada de Decisões	42
• Funções de interface de defuzzificação	43
3.11 Representação da variável de saída: ciclo de PWM	43
4 Resultados Obtidos	46
4.1 Parâmetros para Modelamento do Sistema	46
4.2 Projeto do Controlador PI.....	49
4.3 Projeto do Controlador Fuzzy	51
4.4 Resultados Experimentais do sistema com controlador PI.....	57
4.5 Resultados Experimentais dos Sistemas de Controle <i>fuzzy</i>	58
5 Conclusões	63
6 Referência Bibliográfica	66
APÊNDICE A – ALGORITMO FUZZY	67

APÊNDICE B – PROCEDIMENTOS PARA O USO DO MPLAB	69
APÊNDICE C – CÓDIGO FUZZY EXEMPLO : 2 REGRAS	70
APÊNDICE D – TESTE DO DRIVER COM PWM.....	85

Índice de figuras e tabelas

FIGURA 1 – JATO ISOTÉRMICO CENTRALIZADO.....	15
FIGURA 2 - JATO DE AR ISOTÉRMICO PRÓXIMO AO TETO.....	16
FIGURA 3 - JATO DE AR QUENTE PRÓXIMO AO CHÃO	16
FIGURA 4 – OPTOACOPLADOR	22
FIGURA 5 – MAQUETE DE PAPELÃO –PLANTA BAIXA	23
FIGURA 6 – MAQUETE DE PAPELÃO	24
FIGURA 7 – NOVO MODELO DE MAQUETE – PLANTA BAIXA.....	25
FIGURA 8 – NOVA MAQUETE	25
FIGURA 9 – CIRCUITO TÉRMICO.....	28
FIGURA 11 – CIRCUITO PARA GRAVAÇÃO DO PIC	31
TABELA 2 – O SENSOR LM35[11]	32
FIGURA 12 CONFIGURAÇÕES BÁSICA(A) E “FULL-RANGE”(B).....	32
FIGURA 13 – DIAGRAMA DO VI DE AQUISIÇÃO DE DADOS	37
FIGURA 14 – INTERFACE DO VI DE AQUISIÇÃO DE DADOS	37
FIGURA 15 <i>DRIVER</i> DE POTÊNCIA PROJETADO.....	38
TABELA 2 – COMPONENTES UTILIZADOS NO <i>DRIVER</i>	39
FIGURA 16 – FUNÇÃO DE PERTINÊNCIA TRAPEZOIDAL	41
TABELA 3 – FUNÇÕES DE PERTINÊNCIA NO ALGORITMO	42
FIGURA 16 – RESPOSTA AO DEGRAU DA SALA 1	46
FIGURA 18 – RESPOSTA AO DEGRAU DA SALA 2	47
FIGURA 19 – RESPOSTA AO DEGRAU DA SALA 3	47
FIGURA 20 – RESPOSTA AO DEGRAU DA SALA 4	48
FIGURA 21 – RESPOSTA AO DEGRAU DA SALA 5	48
FIGURA 22 -FUNÇÕES DE PERTINÊNCIA DE ENTRADA TEMPERATURA.....	52
FIGURA 23 - FUNÇÕES DE PERTINÊNCIA DE SAÍDA CICLO DE TRABALHO DO PWM	52
FIGURA 24 – SUPERFÍCIE DE CONTROLE DO CONTROLADOR 1.....	55
FIGURA 25 – SUPERFÍCIE DE CONTROLE DO CONTROLADOR 2	55
TABELA 4 –CARACTERÍSTICAS DO CONTROLADOR <i>FUZZY</i> -	56
FIGURA 26 – RESPOSTA CONTROLE PI.....	57
TABELA 5 – OSCIOSCÓPIO UTILIZADO.....	58
FIGURA 27 – RESPOSTA SISTEMA - REFERÊNCIA 30°C (CONTROLADOR 1).....	59
FIGURA28 - RESPOSTA À ONDA QU7ADRADA (40°C) – CONTROLADOR 1	60
FIGURA 29 – RESPOSTA DO SISTEMA – CONTROLADOR 2	61
TABELA 5 –COMPORTAMENTO DO SISTEMA – CONTROLADOR 1.....	61
TABELA 6 –COMPORTAMENTO DO SISTEMA – CONTROLADOR 2.....	62
FIGURA 30 – REGRAS <i>FUZZY</i>	68

1 Introdução

1.1 Métodos para modelagem de sistemas físicos

Um problema fundamental na ciência e na engenharia é prever o efeito de uma ação particular sobre um sistema físico. Existem basicamente três métodos de modelar um sistema: experimental, matemático e heurístico.

O método experimental necessita da aquisição de dados de entrada e de saída do sistema a ser analisado por parte de um experimentalista. Esses dados são pontos discretos que auxiliam na representação do comportamento do sistema. Uma curva característica é gerada a partir desses pontos e representa uma aproximação do comportamento do sistema. A maior dificuldade encontrada neste método diz respeito aos procedimentos e à instrumentação utilizados para a efetiva execução do experimento.

O método matemático descreve a dinâmica do sistema analisado por meio de equações que, em conjunto, modelam o comportamento do processo. Neste contexto, faz-se necessária determinação de parâmetros que quantificarão o modelo do sistema. Diz-se que o sistema possui parâmetros concentrados quando estes não possuem relação com o distanciamento ou localização, ou seja, pode-se tratar o parâmetro de uma maneira pontual. Em um sistema a parâmetros distribuídos, por sua vez, existe uma dependência entre as variáveis do sistema e a localização, ou seja, não se pode fazer uma análise pontual do parâmetro como um todo. Um exemplo seria o transporte de calor por condução em uma barra metálica onde a fonte de calor encontra-se em uma de suas pontas. A temperatura, neste caso, dependerá do ponto na barra onde ocorrerá a medição.

O equacionamento de sistemas reais é geralmente bastante complicado visto que praticamente todos os sistemas reais possuem não-linearidades. Desta forma, a fim de se obter um modelo idealizado do sistema, são necessárias algumas simplificações a fim de tornar o sistema analisado linear. Sistemas lineares podem ser representados por equações diferenciais ordinárias lineares que geralmente podem ser facilmente resolvidas por diversas técnicas como, por exemplo, transformada de Laplace.

Com a finalidade de simplificar o processo, pode-se assumir que o sistema é linear em um determinado ponto de operação. Considerações podem ser feitas no que diz respeito

à variação das variáveis do sistema com relação ao tempo. Alguns parâmetros do sistema podem ser considerados invariantes no tempo.

O chamado Método Heurístico utiliza-se de um conhecimento prévio do sistema a ser analisado. Por meio desta experiência prévia, pode-se determinar um conjunto de regras práticas que estão associadas ao comportamento do sistema. Parte-se, então, do particular para o geral, do concreto para o abstrato, para se atingir um resultado rigoroso, com definições e demonstrações de lógica.

1.2 Justificativa

Recentemente, crises no setor energético têm chamado atenção de cientistas e governantes no mundo inteiro. Pesquisas têm sido feitas no sentido de tornar mais eficiente o consumo de energia elétrica e assim diminuir gastos. Estudos na área de conservação energética em edificações mostram que alguns projetos chegam a promover reduções de 50 a 75% no consumo de energia em edificações projetadas[9]

No Brasil, a crise energética ameaçou a economia e a tranquilidade do País. Os reservatórios das usinas hidrelétricas do Sudeste, região que mais consome energia no País, ficaram abaixo de seu limite e, aliado a este fator, foram feitos poucos investimentos em geração alternativa. Nesta época, a falta de investimentos no setor foi apontada por especialistas como a principal responsável pela crise.

Em outros países, os investimentos em energia são feitos em usinas de diversos tipos, justamente com o intuito de evitar crises quando um modelo tem problemas. No Brasil, ao contrário, cerca de 87% da eletricidade é de origem hidroelétrica. O resto é produzido pelas centrais termoelétricas (10%) e pelos reatores das centrais nucleares de Angra dos Reis (2%)[10].

Observando o problema em um nível mais voltado ao consumidor, o uso eficiente da energia possui um caráter educacional, à medida que o consumidor utiliza energia elétrica de uma forma mais consciente, e um caráter relacionado ao avanço tecnológico, visto à necessidade de projetos de aparelhos domésticos com maior eficiência energética.

O aparelho de ar condicionado é, dentre os aparelhos domésticos, o que apresenta a maior parcela no consumo de energia elétrica.

Tabela 1 - Consumo de energia em empresas do DF – Fonte: Metroquattro Engenharia

USOS FINAIS	APROPRIAÇÃO DO CONSUMO DE ENERGIA ELÉTRICA – VALORES MÉDIOS		
	Pequeno Porte	Médio Porte	Grande Porte
Ar Condicionado	62,9 %	56,0%	48,6%
Iluminação	28,8 %	32,0%	18,5%
Elevadores		3,5%	10,4%
Bombeamento	-	0,3%	2,5%
CPD's, terminais e computadores.	7,0 %	8,0%	16,0%
Outros	1,3 %	0,2%	4,0%
TOTAIS	100,0 %	100,0%	100,0%

Os dados acima apresentados mostram que o consumo de ar condicionado é bem maior se comparado a iluminação, bombeamento e outros equipamentos elétrico/eletrônicos. Como ilustração, considere que no Laboratório de Automação, Visão e Sistemas Inteligentes, LAVSI, na UnB, um ar condicionado de 12.000 BTUs e 1400 W permaneça ligado durante 8 horas por dia ao longo de mês. A energia consumida é de 336 kWh. Ao comparar com o consumo das lâmpadas do laboratório, a saber, 18 lâmpadas fluorescente de 32 W, ligadas no mesmo intervalo de tempo em um mês, a energia consumida é de 138,24 kWh, aproximadamente 2,5 vezes menor que o consumo do ar condicionado.

O consumo de energia certamente seria minimizado se o uso do aparelho de ar condicionado fosse feito de uma maneira mais racional por parte dos usuários. Muitas vezes, por exemplo, deparamos-nos com situações onde os ocupantes de uma sala, ao deixá-la, esquecem o aparelho ligado. Se o aparelho pudesse tomar algumas decisões independente do usuário, evitar-se-ia situações deste tipo. Neste sentido, o presente trabalho tem como motivação a implementação de uma solução para problemas que envolvem controle de sistemas térmicos. O objetivo principal é o estudo deste tipo de sistema visando a eficiência energética. Para tanto faz-se uso de uma maquete, de forma a obter-se um estudo preliminar do comportamento do sistema. Este projeto tem em vista dar embasamento a um projeto maior onde, usando técnicas de *retrofitting* pretende-se desenvolver um dispositivo controlador que, ao ser conectado ao condicionador de ar, promova o funcionamento de forma otimizada.

2 Fundamentos Teóricos

2.1 Modelamento matemático de sistemas térmicos

Sistemas térmicos são sistemas onde as variáveis envolvidas estão relacionadas com o armazenamento e o transporte (fluxo) de calor. Nestes sistemas, a transferência de calor pode ocorrer em três formas distintas: condução, convecção e radiação[5]. Rigorosamente, as três formas de transferência ocorrem simultaneamente em sistemas. Porém, ao tratar-se o sistema na prática, geralmente uma das três formas é predominante, ou melhor, duas destas formas têm predominância sobre a terceira, o que é mais comum em sistemas reais.

A transferência de calor é a energia em trânsito devido a uma diferença nos níveis de temperatura, seja essa diferença em sólidos ou fluidos. A condução é a forma de transferência de calor em que o meio é estacionário e existe um gradiente de temperatura. Tal transferência ocorre por causa da movimentação aleatória dos átomos ou moléculas envolvidos e ocorre através do meio, seja este fluido ou sólido. Em contraste, a transferência de calor por convecção está relacionada à transferência de calor quando um fluido em movimento encontra-se com uma superfície de temperatura diferente. A convecção envolve dois mecanismos básicos: a movimentação molecular aleatória (difusão) e a movimentação global (macroscópica) do fluido. Já na radiação, a energia térmica é emitida por toda a matéria que possui uma temperatura não-nula.

2.1.1 Transferência e fluxo de calor

Os processos de transferência de calor são quantificados em termos de equações de taxas de transferência de calor apropriadas para cada caso. Tais equações são utilizadas para calcular a quantidade de energia que é transferida por unidade de tempo. No caso da condução de calor, a equação que representa a taxa de transferência de calor é também conhecida com Lei de Fourier[5].

$$q(t) = k.A.\frac{T_1(t) - T_2(t)}{d} \quad (2.1)$$

O fluxo de calor q (Kcal/s), para os casos onde a transferência de calor é feita por condução ou convecção, é representado pela seguinte equação:

$$q = K\Delta T \quad (2.2)$$

onde ΔT representa a diferença de temperatura em Kelvins [K] e K é o coeficiente de proporcionalidade em Kcal/s K, dado por:

$$K = \frac{kA}{\Delta X} \quad \text{na condução} \quad (2.3)$$

e

$$K = hA \quad \text{na convecção} \quad (2.4)$$

onde k é condutividade térmica, em kcal/m.K.s, A é a área normal ao fluxo de calor, em m^2 , ΔX representa a espessura do condutor, em m, e h é o coeficiente de transferência de calor por convecção, em kcal/ m^2 .K.s,[5]

Na radiação, o fluxo é representado por:

$$q = K_r (T_1^4 - T_2^4) \quad (2.5)$$

onde K_r é o coeficiente de proporcionalidade que depende da emissividade, configuração e tamanho da superfície e T_1 e T_2 representam as temperaturas do emissor e receptor, respectivamente.

2.1.2 Variáveis Térmicas

Geralmente, as variáveis térmicas que são utilizadas para o modelamento matemático de sistemas térmicos são:

T, temperatura (em Kelvins, [K])

q, fluxo de calor (em Kcal/s)

Sistemas térmicos são inerentemente sistemas a parâmetros distribuídos, ou seja, suas variáveis variam com a localização ou distância. Em sistemas a parâmetros distribuídos, o modelamento matemático é feito por meio de equações diferenciais parciais. Entretanto, em muitos casos, para simplificar o problema a ser analisado, admite-se que o sistema térmico possa ser representado a parâmetros concentrados, onde as substâncias são representadas por resistências e capacitâncias térmicas. Neste tipo de análise, considera-se que as substâncias que são representadas por resistências térmicas possuem capacitâncias térmicas desprezíveis, e vice-versa. Como conseqüências destas simplificações, o modelamento matemático é representado, neste momento, por equações diferenciais ordinárias.

2.2 Distribuição de Ar no interior de uma sala

O lançamento do ar a partir de saídas arredondadas, painéis perfurados, cortinas de ferro ou outras formas de saída (escape ou fonte) está intimamente relacionada com a velocidade média na área transversal ou face da abertura de onde o ar origina-se.[3]

Algumas considerações devem ser levadas em conta ao tratar-se do posicionamento dos jatos de ar em uma sala.

Para jatos não isotérmicos, o jato de ar tende a subir quando a temperatura do ar oriundo desse jato é maior que a temperatura do ambiente, e tende a descer quando a temperatura do ar no jato é menor que a do ambiente, o que resulta em um alcance menor por parte do jato de ar.

Outro aspecto interessante: quando um jato de ar é projetado paralelamente ou ao alcance de uma superfície qualquer, uma região de baixa pressão é criada entre a superfície e o jato de ar, fazendo o jato de ar atrelar-se à própria superfície. Esse fenômeno resulta em uma mudança no comportamento do jato de ar, aumentando o alcance do jato (ou arremesso) e diminuindo a descida do ar, se o mesmo for frio, ou a subida, se quente, para jatos horizontais quando comparados a condições onde o espaço é aberto.

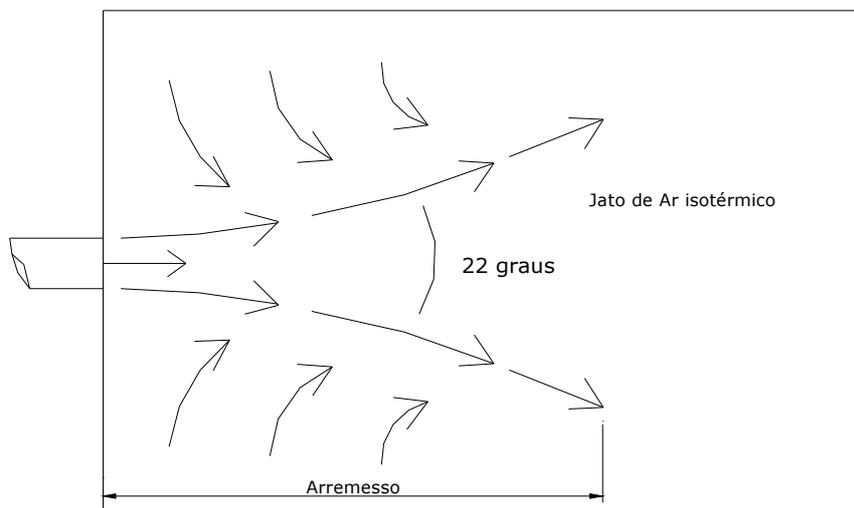


FIGURA 1 – JATO ISOTÉRMICO CENTRALIZADO

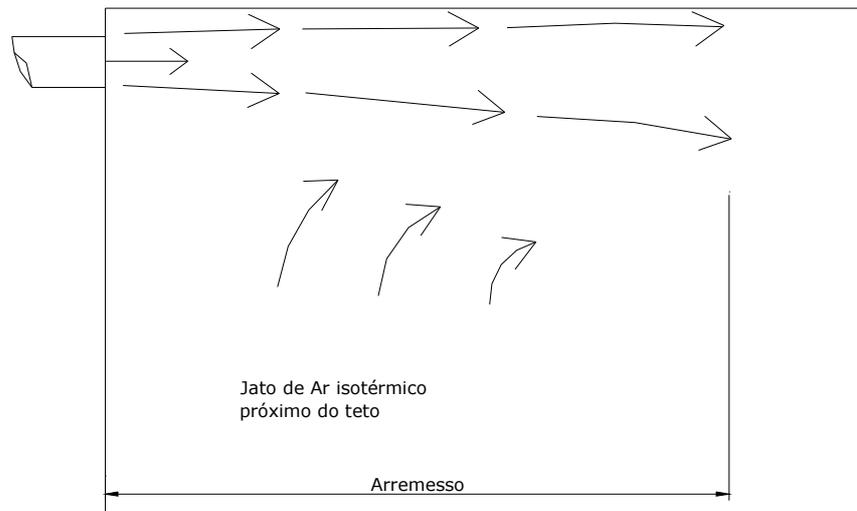


FIGURA 2 - JATO DE AR ISOTÉRMICO PRÓXIMO AO TETO

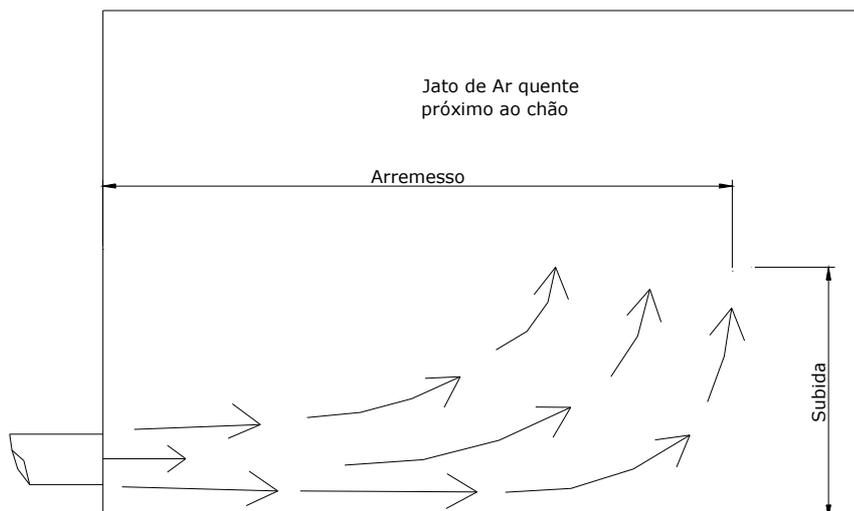


FIGURA 3 - JATO DE AR QUENTE PRÓXIMO AO CHÃO

Pode-se aumentar tal efeito de superfície a fim de se obter um alcance maior. Isso pode ser feito distanciando a saída de ar da superfície de forma que o jato desloca-se o máximo possível antes que tal jato sofra o impacto com a superfície.

Ao espalhar uma corrente de ar em uma sala, os efeitos de alcance e a descida (ou subida) do ar são reduzidos. A descida ou a subida do ar, a priori, é mais dependente da quantidade de ar que do tamanho da fonte ou da velocidade do ar. Isso significa que o uso de um número maior de saídas (fontes) com uma menor taxa de transporte de ar reduz o

efeito de queda, ou descida do ar (para o caso em que o jato é mais frio que o ambiente) e/ou subida (para o caso em que o jato é mais quente que o ambiente).

De uma forma geral, o objetivo da distribuição de ar no interior de uma sala, seja em aquecimento, condicionamento do ar ou ventilação, é proporcionar temperatura, umidade, movimentação e qualidade apropriadas para o ar. Para obter condições confortáveis em um espaço qualquer, limites aceitáveis na variação das variáveis envolvidas devem ser estabelecidos [3]. Esses limites englobam a própria temperatura do ar, a movimentação, a umidade relativa e principalmente os efeitos fisiológicos no corpo humano. Desta forma, qualquer variação maior ou menor que os limites aceitáveis pode causar desconforto nos ocupantes da sala. Além disso, o desconforto pode ser causado, também, pela ausência de condições uniformes ou por excessivas flutuações nas condições na sala ou em parte dela.

2.4 Sistemas Inteligentes

Os chamados sistemas inteligentes são aqueles que solucionam problemas por meio de respostas apropriadas às situações específicas dos problemas em questão. Tais respostas podem ser novas ou até mesmo inesperadas [1]. Neste contexto, redes neurais artificiais, lógica fuzzy, algoritmos genéticos e sistemas especialistas fazem parte do estudo de um ramo da ciência chamado de inteligência artificial. A principal característica de um sistema inteligente é sua capacidade de aprendizado e adaptação a novos problemas ou situações[1].

A inteligência artificial está relacionada ao estudo do comportamento humano bem como às características fisiológicas do corpo humano, uma vez que seu objetivo é estudar as formas que pessoas resolvem problemas para que seja possível fazer uma posterior emulação deste comportamento em máquinas, principalmente computadores.

2.4.1 Lógica Fuzzy

A lógica fuzzy está relacionada ao projeto de controladores onde é incorporada a forma humana de analisar e resolver problemas. Na lógica clássica aristotélica, as expressões ou regras são bivalentes, reconhecendo apenas dois valores: falso ou verdadeiro. Porém, no raciocínio humano, as expressões são geralmente qualitativas e imprecisas. Existem, portanto, “graus” de verdades ou meias-verdades ao se tratar de comunicação e raciocínio humanos. Neste sentido, a lógica fuzzy traduz a linguagem humana, imprecisa e qualitativa, em valores que podem ser compreensíveis em máquinas de aplicações diversas (geralmente computadores), sendo portanto uma lógica multivalente, onde as implicações lógicas estão ligadas a regras de raciocínio e possuem graus de verdade dentro do intervalo $[0 \ 1]$ [1].

A lógica fuzzy permite a utilização de variáveis lingüísticas, típicas da comunicação humana, e regras para a criação de sistemas especialistas. Estas variáveis lingüísticas estão relacionadas tanto ao raciocínio indutivo, ou seja, o aprendizado e a generalização são feitos por meio de exemplos particulares observados de um sistema, quanto ao raciocínio dedutivo, onde a ação e a inferência são provenientes de conclusões baseadas em informações previamente conhecidas.

Para a criação de controladores fuzzy, é necessário possuir conhecimento do sistema que se deseja controlar. Esse conhecimento, proveniente de um especialista, é reunido em um banco de regras. O usuário apresenta ao sistema alguns fatos, e por meio de inferência, ações são tomadas. Esta inferência é realizada pela máquina de inferência, que compara os fatos com as premissas das regras provenientes da base de regras.

Um sistema fuzzy divide-se em quatro partes distintas:

- *Fuzzificador*;
- Banco de regras ou base de conhecimento;
- Inferência, ou lógica para tomada de decisões;
- *Defuzzificador*.

As variáveis utilizadas em aplicações reais são geralmente variáveis contínuas representadas por números reais ou complexos (corrente elétrica, temperatura, umidade,

força, torque etc), em contrapartida, a lógica fuzzy utiliza variáveis lingüísticas. Desta forma, para que exista uma interface entre o controle ou máquina, que utiliza lógica fuzzy, e seu exterior (homem, sensores, sistema,etc) faz-se necessária a conversão dos valores exatos em valores lingüísticos e vice-versa [6]

O fuzzificador realiza a conversão das variáveis exatas em variáveis lingüísticas por meio de um mapeamento das entradas numéricas em conjuntos fuzzy.

A inferência é realizada utilizando-se regras que têm como origem um banco de regras. Valores lingüísticos na entrada são então mapeados em valores lingüísticos na saída. Ao conjunto de condições de entradas da máquina de inferência dá-se o nome de antecedentes e ao conjunto de saídas, responsável pela ação de controle, dá-se o nome de conseqüente.

Existem, basicamente, dois tipos de implicações ou regras de inferência *fuzzy*: o *modus ponens* (modo afirmativo) e o *modus tollens* (modo negativo). Uma regra *fuzzy* ou inferência *fuzzy* relaciona conjuntos usando o *modus ponens* da seguinte forma :

SE <condições> **ENTÃO** < conclusão> ou

SE <antecedentes> **ENTÃO** <conseqüentes>

O defuzzificador realiza a tarefa contrária do fuzzificador, ou seja, converte uma variável lingüística em variável exata. Quando a saída do sistema pode ser interpretada de maneira qualitativa, não se faz necessário o uso desta função. Diferentes métodos de defuzzificação podem ser utilizados dependendo da aplicação. Os mais utilizados são[1]:

7

- Centro da Área (C – o – A) – Enfoque baseado no centróide da área composta pela saída da função de pertinência resultante. Necessita de uma demanda computacional maior, devido ao cálculo de áreas.
- Centro do Máximo (C – o – M) – Os picos da função de pertinência representados no universo de discurso da variável de saída são usados [1] Este método também é chamado de defuzzificação pelas alturas. A saída discreta é calculada pela média ponderada dos máximos, onde os pesos são representados pelos graus de pertinência.

- Média do Máximo (M – o – M) – Calcula-se a média dos valores com máximo grau de pertinência da variável de saída.

2.5 Controle por PWM

Dada uma onda retangular de frequência conhecida e período “T”, cujo intervalo de tempo em nível alto, “V”, possa ser controlado, sendo denominado “T_{on}”. O valor médio desta onda é dado por:

$$V_{DC} = \frac{1}{T} \left(\int_0^T v(t) dt \right) = \frac{1}{T} \left(\int_0^{T_{on}} v(t) dt + \int_{T_{on}}^T v(t) dt \right) = \frac{V}{T} \left(\int_0^{T_{on}} dt \right) = V \frac{T_{on}}{T} \quad (2.6)$$

Esta equação mostra que o valor médio da tensão é diretamente proporcional à quantidade de tempo onde a onda permanece em nível alto em relação ao período da onda. Quando variamos o valor de T_{on} estamos aplicando o controle PWM, que fornece uma relação linear entre a onda quadrada e o seu valor médio.

Uma forma de implementar o controle PWM é utilizando transistores para chaveamento. Um controlador digital pode mandar sinais correspondentes ao *duty cycle* desejado e este chavear um transistor que seria a ligação entre o micro controlador e o circuito a ser controlado.

O chaveamento feito por um transistor bipolar, por exemplo, ocorre entre os modos de operação de corte e saturação. No corte não há nenhuma dissipação de potência e na saturação, devido à tensão na junção coletor-emissor, de aproximadamente 0,2 V, há uma pequena dissipação.

Entretanto, a dissipação acaba sendo um pouco maior pelo fato dos transistores não serem ideais, ou seja, não serem capazes de abrir e fechar o circuito em um tempo ínfimo. Eles precisam de um intervalo de tempo para mudar de estado, aí então, ocorre mais uma pequena perda.

2.6 Microcontroladores

Em poucas palavras, poderíamos definir o microcontrolador como um “pequeno” componente eletrônico, dotado de uma “inteligência” programável, utilizado no controle de processos lógicos[7].

O microcontrolador é programável, pois toda a lógica de operação é estruturada em forma de programa e gravada dentro do componente. Depois disso, toda a vez que o microcontrolador for alimentado o programa interno será executado.

O microcontrolador é bem versátil, pois em uma única pastilha de silício, temos todos os componentes necessários para o controle de um processo, ou seja, ele está provido internamente de memória de programa, memória de dados, portas de entrada e/ou saída paralela, timer's, contadores, comunicação serial, PWM's, conversores A/D etc.[7]

2.6.1 Família PIC (Programmable Integrated Controller)

Os microcontroladores PIC apresentam uma estrutura interna do tipo Harvard, enquanto a maioria dos outros microcontroladores apresenta uma arquitetura do tipo Von-Neumann. A diferença está na forma como os dados e o programa são processados pelo microcontrolador. Na arquitetura Von-Neumann, existe apenas um barramento interno (geralmente de 8 bits) por onde passam as instruções e os dados. Já na arquitetura Harvard existem dois barramentos internos, sendo um para dados e outro para instruções. No caso do PIC, o barramento de dados é sempre de 8 bits e o de instruções pode ser de 12, 14 ou 16 bits, dependendo do PIC.

Todos os microcontroladores PIC seguem a filosofia RISC, tendo como principal vantagem a facilidade de aprendizado. Porém, um número reduzido de instruções faz com que seja necessário o desenvolvimento de várias funções para implementar uma tarefa que seria feita por apenas uma instrução CISC.

A memória de programa dos microcontroladores PIC pode ser de 12, 14 ou 16 bits. O tamanho dessa memória varia muito com o modelo do PIC. Na maioria dos modelos a memória é do tipo EPROM, que só pode ser gravada uma vez para PIC's mais comuns, ou

gravada varias vezes no caso de PIC's com janelas para ultravioleta. Podemos ainda ter memórias EEPROM e FLASH, o que os torna mais caros.

2.7 Optoacopladores

A separação de potenciais exige uma transformação de sinais elétricos em sinais com outras características. No caso do optoacoplador, o sinal elétrico da entrada é transformado em luz. Utiliza-se para tal tarefa dispositivos optoeletrônicos (semicondutores fotosensíveis). Nesses dispositivos optoacopladores, a transmissão do sinal de um circuito para outro, faz-se pela emissão luminosa de um díodo de arseneto de gálio, emissor de infra-vermelhos. Por outro lado, o fototransistor inserido no outro circuito entra em comutação em função da luz recebida.

A figura a seguir representa o Optoacoplador:

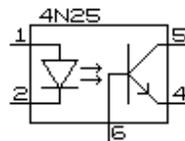


FIGURA 4 – OPTOACOPLADOR

Um optoacoplador pode ser usado para separar o sinal de saída de um microcontrolador de um dispositivo de saída. Isto pode ser necessário para isolar o circuito de uma tensão alta ou para amplificar a corrente.

3 Metodologia

3.1 Sistema Térmico em Estudo – Maquete

Para se estudar um sistema de controle de ar condicionado é necessário possuir espaço disponível para realizar os experimentos, ou seja, deveríamos ter, pelo menos, uma sala onde seriam feitas as medições e onde o ar condicionado iria atuar. Se pensarmos num sistema maior, com mais aparelhos de ar condicionado e mais salas teríamos um alto custo de manter este número de salas inativas, utilizando um espaço que a universidade não tem disponível.

A saída utilizada foi reduzir o sistema, de forma que fosse possível tê-lo em cima de uma mesa. Para tanto, uma maquete foi desenhada e encomendada, imitando um escritório, com diversas salas e dois secadores de cabelo de 1kW de potência simulando os condicionadores de ar. Nota-se que o sistema tratado estará simulando o caso contrário do condicionamento de ar, ou seja, o aquecimento do ar. Porém, para a análise que se deseja fazer (fluxo de calor), tratar de aquecimento ou resfriamento é praticamente o mesmo problema em termos de equacionamento matemático.

Inicialmente foi utilizada uma maquete de papelão da figura 5 representando um escritório típico com oito salas.

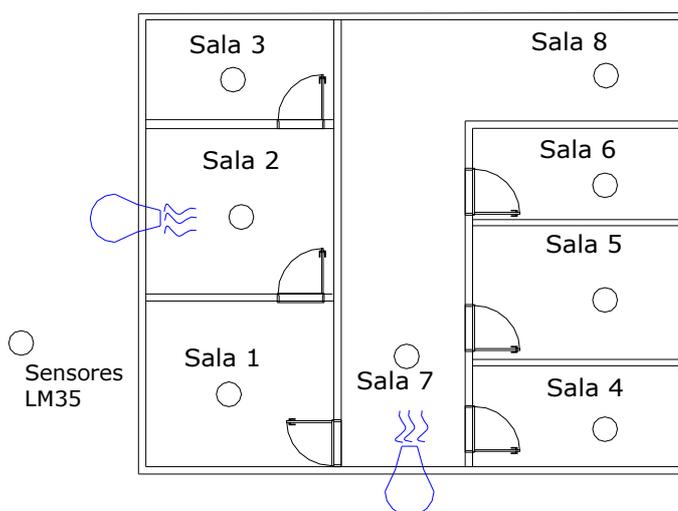


FIGURA 5 – MAQUETE DE PAPELÃO –PLANTA BAIXA



FIGURA 6 – MAQUETE DE PAPELÃO

Em cada uma das salas, sensores de temperatura do tipo LM35 foram utilizados para monitorar a temperatura das mesmas, bem como auxiliar no processo de controle. Porém, tal maquete apresentou alguns problemas quando da simulação do processo. Os sensores das salas que possuíam secador ficavam muito próximos ao mesmo, fazendo com que a temperatura nesses pontos aumentasse rapidamente, o que não corresponde à realidade do sistema térmico que a maquete deve representar. Tal fato dificulta significativamente não só a análise do sistema em si, mas também o controle do processo. Outro problema era o elevado número de salas. Sabe-se que com duas fontes de calor apenas a temperatura em duas salas da maquete pode ser controlada, portanto um número grande de salas complica a análise do processo e também dificulta o modelamento matemático do sistema.

Frente a esses problemas, chegou-se a conclusão de que a confecção de uma segunda maquete era necessária. Este novo sistema mostrado nas figuras 7 e 8 possui um menor número de salas e considera parâmetros importantes que não foram tratados no caso anterior, como a influência do sol na temperatura do sistema.

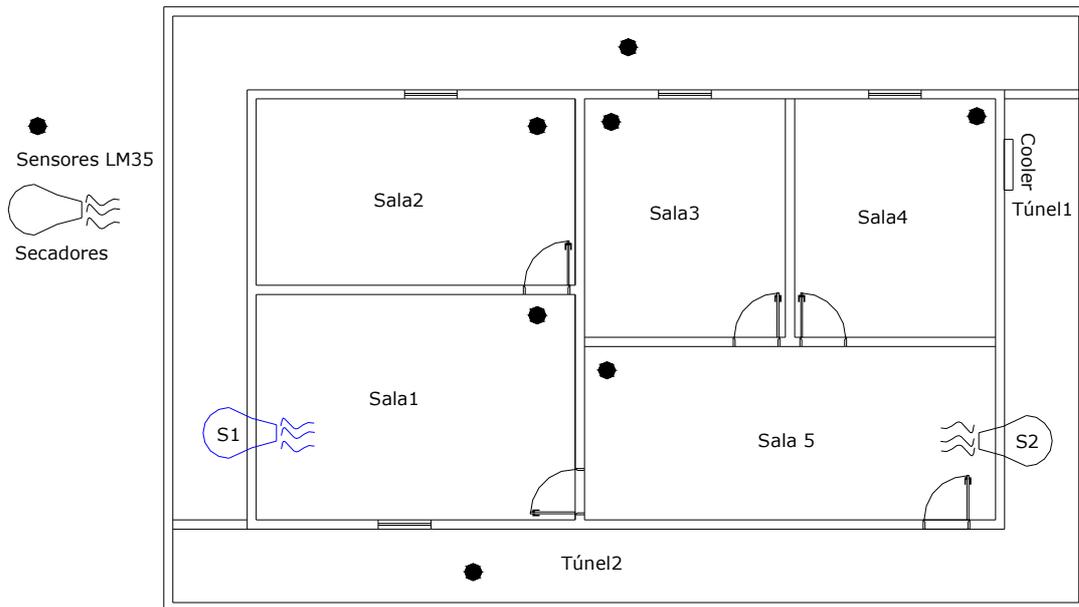


FIGURA 7 – NOVO MODELO DE MAQUETE – PLANTA BAIXA



FIGURA 8 – NOVA MAQUETE

A nova maquete foi construída em madeira (compensado) e possui paredes de espessura igual a 1cm.

A maquete compreende três grandes áreas. A primeira é representada por duas salas adjacentes (salas 1 e 2, área de 1902 cm²). Sendo esta área a menor da maquete, pode-se verificar a condição de um condicionador de ar de grande potência em relação ao seu tamanho, e assim implementar um controle de uma situação crítica. A maior das salas (sala 1 – 1156 cm²) contém a fonte de calor (secador). A segunda área (de 2451 cm²) é representada por duas salas à meia altura (salas 3 e 4 – 1204 cm², somadas) e um sala maior (sala 5 – 1247 cm²), onde se encontra a segunda fonte de calor (secador de cabelo). Em ambos os casos, procurou-se localizar os sensores de temperatura de forma que o fluxo de calor não fosse direcionado sobre eles, ou seja, procurou-se evitar a proximidade dos sensores às fontes de calor. A adição de salas com paredes à meia altura (salas 3 e 4) foi feita no intuito de simular paredes que podem ser representadas por resistências em paralelo. Essas duas áreas representam a maquete principal, ou seja, o escritório. A terceira área é representada por dois túneis a fim de simular o efeito do sol. O ar que circula por estes túneis pode circular sobre bolsas de gelo, ou seja, o ar é resfriado. Como o referido sistema simula o aquecimento do ar, o ar frio que circula no exterior da maquete principal, ou seja, no túnel simula o efeito do calor irradiado pelo sol.

Os sensores de temperatura estão localizados em cada uma das salas (total de cinco sensores) e nos túneis para a medição da temperatura externa ao escritório (dois sensores).

Os secadores foram posicionados de forma a garantir um maior arremesso do ar e assim uma melhor distribuição do mesmo, conforme explicado no item 2.2. Para tanto, os secadores estão localizados próximos ao chão da maquete. Além disso, de forma a facilitar a execução de diversos experimentos, um “cooler” foi colocado em posição estratégica(sala 4) a fim de que, quando finalizado um experimento, o resfriamento do ar no interior da maquete fosse feito de maneira mais rápida e eficaz. Para permitir a visualização, a tampa da maquete é de vidro.

3.2 Modelamento Matemático do Sistema

3.2.1 Parâmetros Concentrados

Para efetuar o modelamento matemático do sistema térmico, neste caso representado pela maquete, uma análise a parâmetros concentrados é realizada. Considerando que não existe um gradiente de temperatura muito alto ao longo das salas e que a temperatura nas salas está sendo medida de maneira pontual, ou seja, não são feitas diferentes medidas de temperatura ao longo das salas, pode-se então associar uma capacitância térmica a cada sala, que representa o calor armazenado em seu interior, e uma resistência térmica a cada parede de uma sala, que representa a restrição por parte das paredes ao fluxo de calor. Cabe lembrar que o uso de parâmetros concentrados também pode ser justificado pelo fato de que uma estimativa do comportamento dinâmico do sistema está sendo procurada.

Pode-se, então, fazer uma analogia do “circuito térmico”, que representa a maquete, a circuitos elétricos formados por capacitores e resistores. Além disso, deve-se determinar quem são as variáveis de fluxo e de esforço do sistema citado. O fluxo de calor é uma variável de fluxo do sistema, bem como a corrente elétrica em um sistema térmico, já a temperatura é uma variável de esforço, que em um circuito elétrico é representada pela tensão elétrica. Pode-se então representar a temperatura em cada uma das salas (medida pelo sensor) analogamente a uma tensão sobre um capacitor. As fonte de calor, ou seja, os secadores, podem ser representados como fontes de fluxo de calor, o que seria o análogo a fontes de corrente em circuitos elétricos. Assim, pode-se montar o “circuito” térmico como representado na figura 9 a seguir.

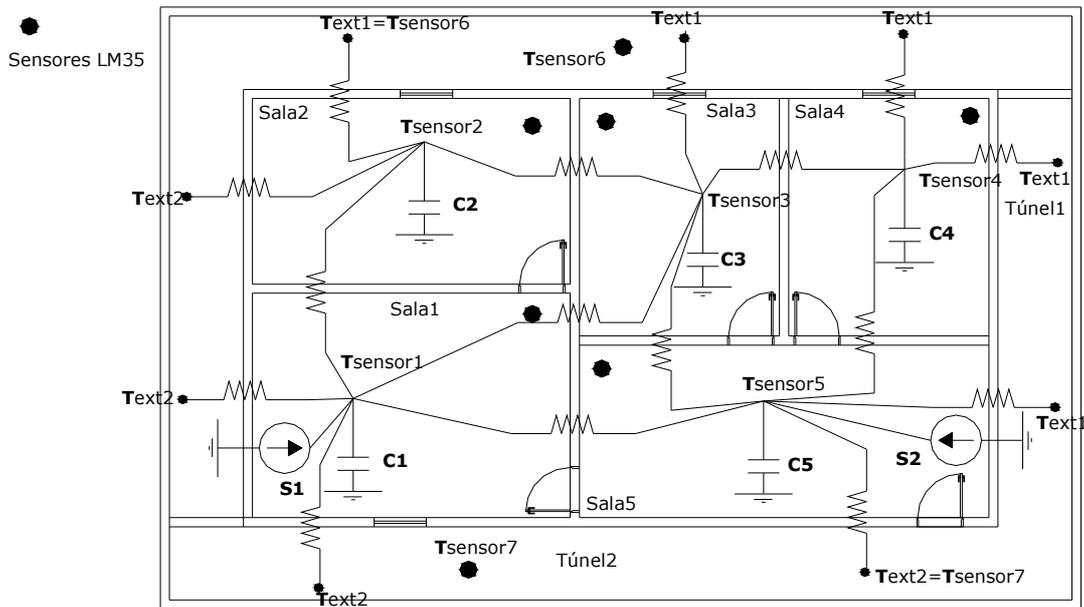


FIGURA 9 – CIRCUITO TÉRMICO

A constante de tempo de cada uma das salas está intimamente ligada ao valor das resistências térmicas (R) e capacitâncias térmicas (C_i), onde i representa o número da sala. Em um sistema envolvendo uma capacitância e uma resistência em série, tal constante de tempo é igual a RC , por se tratar de um sistema de primeira ordem. Porém, devido ao fato do sistema tratado não ser de primeira ordem, a constante de tempo dependerá de qual sala estará envolvida no processo e das capacitâncias e resistências das mesmas. Tais constantes de tempo podem ser encontradas de maneira experimental isolando as salas envolvidas nas quais se deseja obter tal constante, considerando desta forma um sistema menor. Por exemplo, no isolamento de duas salas adjacentes, sendo que uma possui uma fonte de calor (sala A), e a outra apenas um sensor de temperatura (sala B), o sistema é representado por uma resistência (parede que separa as salas), duas capacitâncias (representando cada sala) e uma fonte de “fluxo” de calor. Se considerada uma sala isolada, o modelo pode ser representado por um sistema de primeira ordem. Nestes sistemas, pode-se obter a constante de tempo experimentalmente marcando-se o tempo de subida. A constante de tempo será o intervalo de tempo em que o sistema leva para alcançar 63,7% do valor final. Para o sistema citado acima, por exemplo, poder-se-ia marcar o intervalo de tempo em que a temperatura da sala 1 (possui a fonte de calor) leva para estabilizar-se, considerando que o atuador está sendo ligado no máximo da potência e uma entrada degrau.

3.3 Microcontrolador Utilizado

3.3.1 PIC16F877A

Este modelo foi escolhido por ter memória suficiente para o controle do sistema em estudo, além de possuir 2 módulos que servem para captura, comparação e geração de sinal PWM.

O PIC possui as seguintes características principais:

- CPU RISC;
- Microcontrolador de 40 pinos, o que possibilita a montagem de um hardware complexo e capaz de interagir com diversos recursos e funções ao mesmo tempo;
- Via de programação com 14 bits e 35 instruções;
- 33 portas configuráveis como entrada ou saída;
- Memória de programação E²PROM FLASH
- Memória de programa com 8kwords, com capacidade de escrita e leitura pelo próprio código interno;
- Memória E²PROM (não volátil) interna com 256 bits;
- Comunicações seriais: SPI, I²C e USART;
- Conversores analógicos de 10 bits (8x);
- Dois módulos de comparação CCP: Capture, Compare e PWM;
- Velocidade de operação de até 20 MHz;
- Quatro opções de oscilador: ressoador, cristal, RC ou circuitos híbridos;

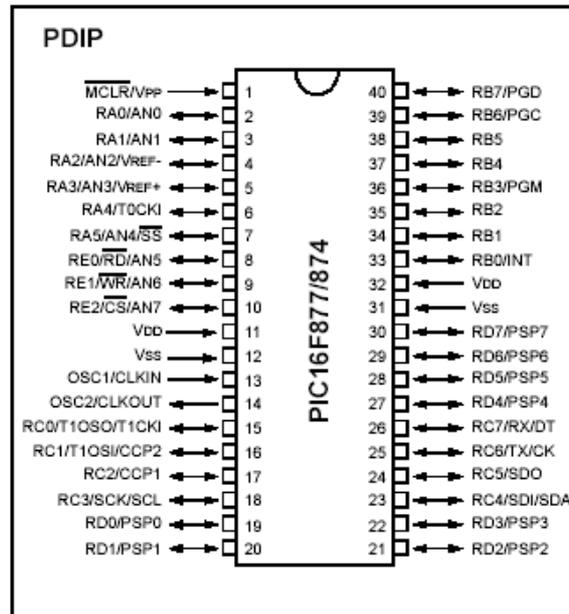


Figura 10 – Pinagem do PIC16F877A, encapsulamento PDIP

O modelo de PIC utilizado da série 16F877A, atualmente só permite que sua programação seja feita em *Assembly*, não há disponível um compilador em C ou em BASIC como no modelo 16F877. Isso impõe uma maior dificuldade de programação mas também permite que os programas sejam mais eficientes e se utilize melhor a memória disponível.

Uma característica desse microcontrolador é que seu ciclo de máquina interno corresponde ao externo dividido por 4. Isso ocorre pela forma que o PIC trabalha internamente. Para a execução de uma única instrução pelo processador várias operações precisam ser realizadas. Como não existe capacidade de processamento paralelo, estas operações são executadas em sub-ciclos do ciclo de máquina, originadas pela divisão do clock externo, esses sub-ciclos são quatro.

Os registradores do PIC podem ser divididos em dois grupos, Registradores especiais e os de uso geral. Os de uso geral são para o armazenamento de variáveis definidas pelo usuário para serem escritas e lidas pelo programa.

Os registradores especiais são utilizados pelo microcontrolador para execução do programa e processamentos de ULA. Esses registradores podem ser escritos e lidos tanto pelo usuário quanto pelo hardware, e servem também para configuração de diversas funções e para utilização de todos os periféricos.

3.4 Procedimentos para o uso do PIC16F877A e MPLAB

O desenvolvimento do programa é realizado por meio de um pacote de programas chamado MPLAB. O MPLAB é um ambiente de desenvolvimento para a linguagem de programação standard de microcontroladores destinado a computadores pessoais.

A linguagem de programação utilizada para o desenvolvimento é o assembly do da PIC16F87X. O MPLAB possui bibliotecas padrão para diversos modelos de PIC. Para que os registradores do PIC possam ser utilizados, faz-se necessário o uso dessas bibliotecas. Para o caso do PIC16F877A, o arquivo da biblioteca chama-se P16F877A.inc. As principais etapas para o desenvolvimento de um programa no MPLAB encontram-se no apêndice B.

3.5 Circuito para gravação do PIC

Para realizar a gravação do PIC16F7877A, utilizou-se o circuito de gravação uCFlash. Este gravador opera dentro do MPLAB e grava toda a linha Flash da *Microship*, famílias 12F, 16F e 18F de 8 ou 40 pinos. O circuito é constituído por uma placa com soquete para gravação, um cabo de interligação e um cabo serial DB – 9. No mercado, tal circuito custa cerca de \$34,00.

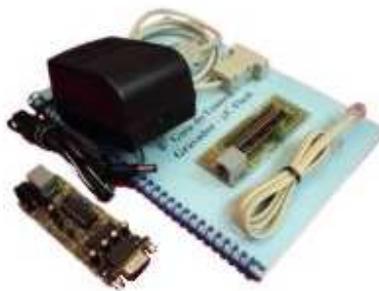


FIGURA 11 – CIRCUITO PARA GRAVAÇÃO DO PIC

3.6 Medição da Temperatura

3.6.1 Sensor Utilizado

Os sensores LM35 utilizados no projeto são circuitos integrados e possuem tensão de saída proporcional a temperatura local em graus Celsius. Eles utilizam as propriedades dos semicondutores para a medição da temperatura.

Uma grande vantagem deste tipo sensor é a facilidade de uso. Uma precisão de 0,5°C é fornecida quando em 25 °C. Drena baixa corrente, em torno de 60 µA, impedindo que ocorra grande dissipação de energia resultando num aumento insignificante da temperatura[11]. Segue uma tabela com um resumo das especificações do LM35.

TABELA 2 – O SENSOR LM35[11]

Características	Valor
Fator de Escala	10 mV/°C
Faixa de Operação	-55 °C a 150 °C
Precisão	0,5 °C (a 25 °C)
Consumo de Corrente	Menor que 60 µA
Alimentação	De 4 a 20 V

Existem algumas configurações para conectar o LM35 ao circuito. A configuração básica permite obter temperaturas de 2 °C a 150 °C, já a configuração “Full-Range” permite que o sensor atue em toda sua faixa de operação. A seguir são mostradas as duas configurações[11].

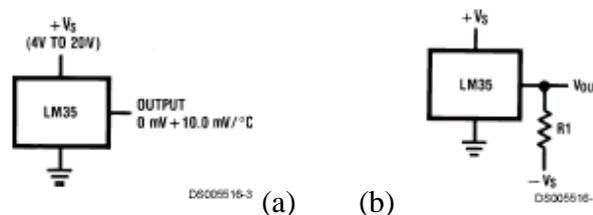


FIGURA 12 CONFIGURAÇÕES BÁSICA(A) E “FULL-RANGE”(B)

Neste projeto, o sensor foi utilizado em sua configuração básica e alimentado com uma tensão de 5V.

3.6.2 Leitura da Temperatura pelo Microcontrolador

A leitura do valor da tensão de saída do sensor LM35 é realizada pelo microcontrolador PIC16F877A por meio de seu conversor A/D interno. Este conversor possui as seguintes características:

- Representação da conversão em 10 bits, totalizando 1024 pontos;
- Máximo de 8 canais de conversão, com configurações entre analógicos e digitais;
- 4 tipos de referência: Vdd(interna), Vss(interna), Vref+(externa) e Vref-(externa);
- Frequência de conversão baseado no clock da máquina ou circuito RC externo;
- 3 ajustes de frequência e 2 tipos de justificação do resultado: direita ou esquerda;
- 1 interrupção para término de conversão.

O sistema de conversão utilizado pelo PIC é do tipo aproximação sucessiva. Neste sistema, a conversão é realizada do bits mais significativo(MSB) ao menos significativo.

O resultado da conversão é armazenado em dois registradores: ADRESH e ADRESL. Justificando-se o resultado da conversão pela esquerda, os 8 bits mais significativos são armazenados em ADRESH e os dois menos significativos em ADRESL.

Pode-se perceber de forma empírica que o sistema térmico em estudo não ultrapassa o valor de temperatura de 70 °C. Considerando que 1024 valores de temperatura podem ser representados com 10 bits, a menor variação de temperatura medida é igual a :

$$T_{med} = \frac{T_{max}}{\# pontos} = \frac{70}{1024} \cong 0.068^{\circ} C$$

Observa-se que este valor é menor que a precisão do sensor LM35 (0,5 °C), portanto não justifica a utilização de 10 bits para conversão A/D. Além disso , existem problemas relacionados ao uso desta quantidade de bits para a conversão, tais como a dificuldade computacional ao tratar apenas dois bits no registrador ADRESL e possíveis oscilações no

valor final da conversão [8]. Desta forma, fez-se a opção de se utilizar apenas os 8 bits mais significativos da conversão, armazenados no registrador ADRESH.

Considerando que a menor variação do valor de temperatura convertido seja igual à precisão do sensor, temos que o valor máximo de temperatura medido é:

$$T_{med} = \frac{T_{max}}{255} = 0,5^{\circ} C \Rightarrow T_{max} = 127,5^{\circ} C$$

Outro aspecto de relevância diz respeito à tensão de referência. A tensão de referência escolhida é representada pela referência interna (5V) e, desta forma, a conversão resultará no valor máximo (255 em ADRESH) quando o valor de entrada do conversor A/D for igual a este valor de referência. A tabela a seguir apresenta a relação existente entre tensão de entrada do conversor, temperatura e valor convertido digital do conversor.

Tabela 2 –Fatores de escala da Conversão AD

Sinal de entrada (V)	Temperatura(°C)	Sinal Convertido Digital
5	127,5	255

A fim de utilizar a capacidade de representação do conversor A/D em sua totalidade, o sinal de saída do sensor LM35 precisa ser amplificado e calibrado para que a relação descrita na tabela acima seja válida.

Sabe-se que o valor de temperatura 100 °C resulta em um sinal de 1V na saída do sensor. De acordo com a tabela 2, este valor de temperatura deve representar um sinal de saída no amplificador dado por:

$$\frac{V_{out}}{100^{\circ} C} = \frac{5V}{127,5^{\circ} C} \Rightarrow V_{out} = 3,9215V$$

Logo, o sinal amplificado deve ter um ganho de 3,9215.

Utilizando um amplificador operacional 741 em sua configuração não – inversora, temos:

$$\frac{V_{out}}{V_{sen sor}} = 1 + \frac{R_2}{R_1} = 3,9215V \Rightarrow \frac{R_2}{R_1} = 2,9215$$

3.6.3 Ajustes dos parâmetros do PIC para medição

O primeiro aspecto que deve ser observado é que apesar de o microcontrolador possuir 8 canais analógicos, existe apenas um sistema de conversão internamente. Desta forma, apenas um canal pode ser usado por vez. O PIC16F877A permite que estes canais (pinos) sejam configurados também como I/Os digitais. Isso é feito alterando valores dos bits no registrador $ADCON1\langle PCFG3:PCFG0 \rangle$. O posicionamento do bit NSP é feito por meio da alteração do bit ADFM deste registrador. Para o caso da justificação à esquerda, o valor deste bit é zero. São utilizados no sistema sete canais analógicos. Além disso, faz-se necessário, também, ajustar corretamente os registradores TRISA e TRISE para que tais canais sejam ajustados como entrada.

O próximo ponto a ser tratado se refere à velocidade e ao tempo de amostragem da conversão A/D. Com o intuito de evitar ruídos e variações na entrada, o PIC utiliza-se do processo *Sample and Hold* (S/H). Portanto deve ser levado em consideração o tempo de carregamento do capacitor (120pF) interno do PIC ligado ao canal analógico utilizado. O tempo padrão necessário para garantir a carga do capacitor deve ser de pelo menos 40µs. Para o funcionamento correto do sistema de conversão, um clock de período T_{AD} é aplicado ao mesmo. A cada período T_{AD} é convertido um bit. Como o conversor é de 10 bits, o tempo total para conversão é de 10 T_{AD} . Além disso, acrescentam-se a este tempo 2 T_{AD} referentes à adequação e ao início da conversão, 2 T_{AD} referentes à atualização dos registradores do resultado e 40µs que representam a nova adequação do capacitor (troca de canal).

A frequência do conversor pode ser configurada por meio de 2 bits do registrador $ADCON0\langle ADCS1:ADCS0 \rangle$. Dependendo do ajuste do valor desse registrador, a frequência de trabalho pode ser a frequência do oscilador externo do PIC (F_{osc}) dividida por 2, 8 ou 32. Para o caso deste projeto, optou-se por dividir a frequência por 32, então:

$$f_{conv} = \frac{1}{T_{AD}} \Rightarrow T_{AD} = \frac{32}{4MHz} = 8\mu s$$

No caso do PIC16F877A, o valor mínimo aceitável de T_{AD} é igual a 1,6µs. Valores muito altos também não ocasionam uma resposta adequada, portanto aconselha-se que o valor de T_{AD} não pode ultrapasse de 20µs [8].

O tempo para conversão do sinal do sensor é, portanto, igual a:

$$T_{conv} = 40\mu s + 10T_{AD} + 2T_{AD} + 2T_{AD} = 152\mu s$$

São realizadas, em cada ciclo do programa de controle, sete medições de temperatura, totalizando 1,064 ms para conversão do sinal.

3.7 O LabVIEW (Laboratory Virtual Instrument Engineering Workbench)

O LabVIEW da *National Instruments*® é um sistema de programação gráfica para aquisição, análise, controle e apresentação de dados, que utiliza uma metodologia de programação onde o usuário, graficamente, cria seus programas chamados instrumentos virtuais.

Nele os programas são criados no formato de diagrama de blocos, um modelo de programação dataflow (as instruções são executadas apenas quando todas as entradas estão disponíveis), que difere da arquitetura linear das linguagens baseadas em texto. Ou seja, a ordem de execução é determinada pelo fluxo de dados entre blocos lógicos e não por seqüências de linhas de texto. Possuindo uma estrutura modular e hierárquica que permite projetar e modificar sistemas com facilidade e num curto intervalo de tempo além de não requerer conhecimentos prévios de programação já que utiliza uma metodologia de programação gráfica alternativa.

O programa possui inúmeras bibliotecas. Dentre elas bibliotecas para aquisição de dados, controle de instrumentos via porta de comunicação serial, GPIB (*General Purpose Interface* bus, barramento padrão para controle de instrumentos eletrônicos utilizando computador, também conhecido como IEEE 488 bus), além de bibliotecas de funções para as mais variadas aplicações.

O instrumento virtual desenvolvido para a aquisição das temperaturas nas salas é mostrado na figura 13. Para cada sala se tem um bloco formado por dois VI's, um "AI Sample Channel", que é responsável pela aquisição em si, e um "Write to Spreadsheet File" que grava os valores das amostras num arquivo texto. Os demais blocos e comparadores são utilizados para controle de parada e desenho dos gráficos no próprio LabView. Na figura 14 é mostrada a interface criada para visualização das temperaturas instantâneas.

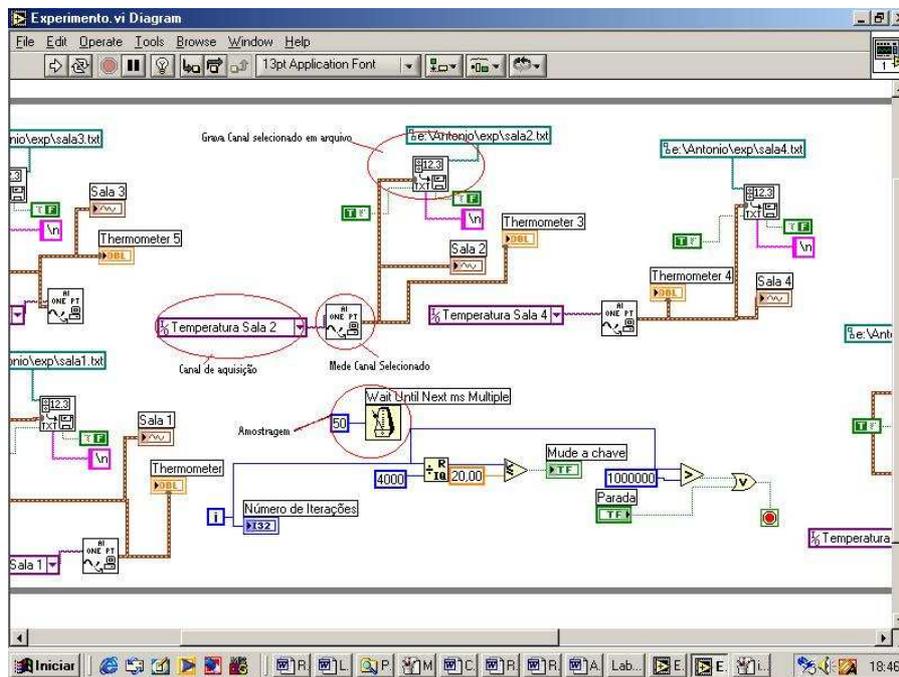


FIGURA 13 – DIAGRAMA DO VI DE AQUISIÇÃO DE DADOS

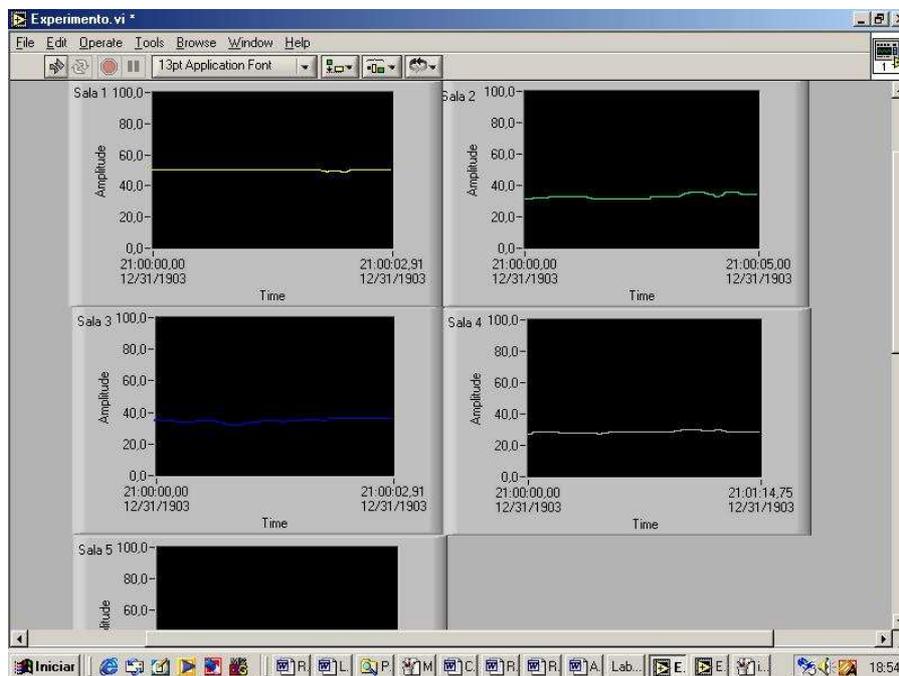


FIGURA 14 – INTERFACE DO VI DE AQUISIÇÃO DE DADOS

TABELA 2 – COMPONENTES UTILIZADOS NO DRIVER

Capacitor	220 μ F e 200 V
Transistor de Potência	BU 426 ^A
Transistor de Sinal	2N2222
Diodo	D1N4004
Ponte de Diodos	PBL 406
Optoacoplador	4N35

3.9 Programa para teste do Módulo PWM do PIC e driver de Potência

Um programa para testes do driver de potência foi desenvolvido para verificar sua resposta de acordo com o valor do ciclo de PWM de saída do PIC16F877A . Este programa mostrou-se bastante útil para o teste de diversos tipos de transistores no driver. A lógica do algoritmo é bastante simples: a cada 5s é somado 25% no valor do ciclo de trabalho do PWM. Ao final do *loop* do programa, o ciclo de trabalho retorna ao valor 0%. O algoritmo encontra-se no apêndice D.

3.10 Algoritmo do controlador Fuzzy

- **Funções de interface de fuzzificação**

Os valores da variável de entrada são obtidos por meio do sinal de tensão analógico do sensor LM35. A taxa de amostragem do sistema está também relacionada com o número de instruções no código do programa. Cada instrução leva 4 ciclos de clock para ser executada, portanto o tempo gasto por instrução é igual a:

$$T_{instrução} = \frac{4}{4MHz} = 1\mu s$$

A conversão do sinal do sensor é realizada em 152 μs por meio do conversor A/D interno do PIC, conforme explicado no item 3.6. O código do programa possui cerca de 400 instruções. De acordo com simulações no programa MPLAB, a taxa de amostragem (conversão do sinal de temperatura e algoritmo) é de 4,065 ms.

O universo de discurso para a variável de entrada “temperatura” é representado pelo intervalo [0°C, 127,5°C]. Internamente no PIC, este conjunto é representado por 255 entradas discretas, as quais são membros do conjunto definido pelo intervalo [0, 255]. Logo, tem-se um fator de escala para a variável de entrada “temperatura” igual a 2.

A interface de fuzzificação utiliza funções de pertinência contidas na base de conhecimento, convertendo o sinal de entrada em valores chamados de graus de pertinência, dentro de um intervalo [0,1]. De forma a aproveitar-se a resolução máxima fornecida com o uso de um byte, este intervalo foi mapeado em [0, 255]. A menor variação representada de um grau de pertinência é dada por:

$$\mu(x)_{\min} = \frac{1}{255} = 3.92 \times 10^{-3}$$

Devido à facilidade de implementação, funções de pertinência trapezoidais são utilizadas para a representação das variáveis lingüísticas de entrada e de saída. No programa do controlador *fuzzy* implementado, o valor do grau de pertinência (DOM – *degree of membership*) é calculado por meio da função F_TRAPEZOIDAL. São

parâmetros de entrada desta função o valor PONTO, que representa o valor de entrada da função trapezoidal, M1 e M2, representando inclinações de retas e mais quatro variáveis, a saber, BESQ (baixo esquerda), BDIR (baixo direita), AESQ (alto esquerda), ADIR (alto direita), representando os dois pontos da base maior (abaixo) e os dois da base menor (acima) do trapézio, respectivamente. A figura a seguir ilustra a posição destes pontos.

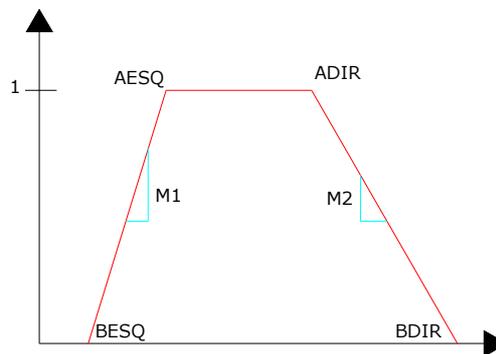


FIGURA 16 – FUNÇÃO DE PERTINÊNCIA TRAPEZOIDAL

As constantes M1 e M2 podem facilmente ser calculadas dentro do próprio algoritmo. Porém este cálculo exige operações de divisão, que representam aumento significativo de tempo de execução. Optou-se, então, em considerar tais constantes como parâmetros da função.

A função F_TRAPEZOIDAL retorna o valor DOM, equivalente ao grau de pertinência referente ao valor armazenado em PONTO. Para cada variável lingüística, os valores de entrada de F_TRAPEZOIDAL devem ser inicializados de forma que a função obtenha o formato desejado.

Aspectos importantes devem ser observados de forma que o controlador fuzzy seja preciso, tais como o número de funções de pertinência utilizado e o grau de superposição entre tais funções.

- **Base de Conhecimento e Lógica de Tomada de Decisões**

O modelo do sistema a ser controlado é representado pela base de conhecimento. Esta base consiste de uma base de dados, responsável pelas definições numéricas referentes às funções de pertinência, e uma base de regras, que representa as ações e a estratégia que devem ser tomadas para o devido controle do sistema. Essas regras geralmente são definidas por especialistas. Por meio de experimentos, pode-se conhecer melhor a dinâmica do sistema térmico e elaborar de forma criteriosa o conjunto de regras que representam sua base. À medida que o número de regras utilizado aumenta, maior é a demanda computacional exigida por parte do algoritmo de controle.

Para facilitar a definição destas regras no algoritmo, foram criadas funções que representam a condição de cada variável lingüística. Todas as funções implementadas são apresentadas na tabela a seguir:

TABELA 3 – FUNÇÕES DE PERTINÊNCIA NO ALGORITMO

Funções	Descrição
TEMPERATURA_FRIO, TEMPERATURA_AGRADAVEL, TEMPERATURA_MORNO, TEMPERATURA_QUENTE, TEMPERATURA_MUITO_QUENTE	Funções de pertinência de entrada para variável temperatura.
PWM_DESLIGA, PWM_BAIXO, PWM_BAIXO_MEDIO, PWM_MEDIO, PWM_MEDIO_ALTO, PWM_ALTO, PWM_MAX	Funções de pertinência de saída para variável ciclo de PWM.

Dentro de cada uma das funções, são especificados os valores dos parâmetros M1, M2, BESQ, AESQ, BDIR e ADIR referentes a sua respectiva função de pertinência .

A lógica de inferência que representa o sistema térmico em estudo é chamada de inferência máx-mín. Primeiramente, as variáveis de entrada *fuzzy* (antecedentes) são combinadas utilizando o operador mín, correspondente ao conectivo **E** (regras de intersecção fuzzy). A esta operação dá-se o nome de agregação. A segunda etapa, chamada de composição *fuzzy*, combina os resultados das operações anteriores por meio da operação máx, correspondente ao conectivo **OU**. O conectivo **OU** cria um envelope comum entre os conjuntos dos consequentes *fuzzy*. Isto é realizado no programa por meio da função F_OR.

Todo o procedimento para a implementação da lógica de inferência no algoritmo está apresentado no apêndice A.

- **Funções de interface de defuzzificação**

Na defuzzificação, o valor da variável linguística de saída inferida, resultado da composição *fuzzy* (OU), é traduzido em um único valor discreto. Este valor é utilizado para gerar a ação de controle. O cálculo deste valor discreto é obtido por meio do método de defuzzificação média do máximo (M - o - M), que calcula a média dos valores no universo de discurso com máximo grau de pertinência. Este método foi escolhido devido ao seu pequeno esforço computacional, fator relevante ao se tratar de aplicações em microcontroladores.

O método de defuzzificação é realizado pela função F_MEDIA_MAX que calcula a média dos valores no universo de discurso referentes aos graus de pertinência máximos.

3.11 Representação da variável de saída: ciclo de PWM

O microcontrolador PIC16F877A possui dois canais de PWMs (CCP1 E CCP2), sendo que cada um possui uma resolução de 10 bits. Portanto, o ciclo de trabalho do PWM pode ser regulado de 0 a 100% com uma resolução máxima de 1024 pontos.

Pode-se determinar o período do sinal PWM diretamente ajustando-se o Timer 2, por meio do registrador PR2. O período e a frequência do PWM se relacionam com o valor de PR2 da seguinte forma [8].

$$T_{pwm} = [(PR2 + 1)] \times 4 \times T_{osc} \times (\text{Pr escaleTMR2})$$
$$f_{pwm} = 1/T_{pwm}$$

O tempo que o sinal PWM permanece em nível alto (largura de pulso) é ajustado diretamente por meio dos registradores CCPRaL (8bits) e CCPaCON<CCPaX:CCPaY>(2 bits), de acordo com a seguinte relação.

$$t_{alto} = CCPRaL : CCPaCON < CCPaX : CCPaY > \times T_{osc} \times (\text{Pr escaleTMR2})$$

Portanto, o ciclo de trabalho PWM (CT) em porcentagem é dado por:

$$\frac{t_{alto}}{T_{pwm}} = \frac{CCPRaL : CCPaCON < CCPaX : CCPaY > \times T_{osc} \times (\text{Pr escaleTMR2})}{[(PR2 + 1)] \times 4 \times T_{osc} \times (\text{Pr escaleTMR2})} \times 100$$

Onde a pode ser 1 ou 2, dependendo do canal PWM utilizado.

Nota-se que o ciclo de trabalho não depende do valor do Prescale do TMR2 e do período do oscilador.

$$CT(\%) = \frac{t_{alto}}{T_{pwm}} = \frac{CCPRaL : CCPaCON < CCPaX : CCPaY >}{[(PR2 + 1)] \times 4} \times 100$$

O universo de discurso para a variável de saída “ciclo de PWM” é representado pelo intervalo [0%,100%]. Observa-se que o ciclo de PWM no PIC é controlado diretamente por 10 bits armazenados de forma que os 8 bits mais significativos encontram-se no registrador CCPRaL e os 2 menos significativos em CCPaCON<CCPaX:CCPaY>. O número de pontos de ajuste do ciclo de PWM é quatro vezes maior que o valor ajustado em (PR2+1). Em termos de bits, significa dizer que o ciclo de trabalho é dois bits maior que o número de bits que formam PR2. A manipulação desses dois bits menos significativos requer um esforço computacional maior por parte do algoritmo de controle, visto que as operações deverão ser tratadas em nível de 2 bytes, mesmo que apenas dois bits do registrador CCPaCON sejam utilizados. De forma a padronizar o algoritmo de controle, optou-se por utilizar apenas 8 bits para a representação do universo de discurso da variável de saída.

Portanto, internamente no PIC, o ciclo de PWM é representado por 255 saídas discretas, membros do conjunto definido pelo intervalo [0,255]. Estes valores de saída são diretamente armazenados em CCPRaL. Os bits CCPaX e CCPaY do registrador CCPaCON são setados sempre em zero.

Devido a tais características, o fator de escala do universo de discurso da variável “ciclo de PWM “ é igual a:

$$\frac{CT(\%)}{CCPRaL} = \frac{4}{[(PR2 + 1)] \times 4} \times 100 = \frac{100}{PR2 + 1}$$

Para o projeto, optou-se em trabalhar com um valor em PR2 igual a 255 e prescale do Timer 2 em 16. Portanto a frequência do PWM, considerando que o oscilador fornece uma frequência de 4MHz, é igual a:

$$T_{pwm} = [(254 + 1)] \times 4 \times 250 \times 10^{-9} \times (16) = 4,08ms$$

$$f_{pwm} = 245,09Hz$$

4 Resultados Obtidos

4.1 Parâmetros para Modelamento do Sistema

Considerando um sub-sistema representado por uma sala, a função de transferência que descreve o comportamento do mesmo, sendo a entrada o ciclo de trabalho do PWM é:

$$\frac{\text{Temperatura}}{\text{Duty_Cicle}} = \frac{Kp}{\tau s + 1}$$

As seguintes curvas representam a resposta do sistema para uma entrada degrau, considerando o máximo de potência fornecida ao secador (ciclo de trabalho do PWM de 100%). Ao mesmo tempo em que foram medidas as constantes de tempo para determinada sala, foi mensurada também a influência desta entrada sobre as outras salas, resultando também em curvas. Para a sala 1, foram medidos 71715 pontos, com um intervalo de tempo de 50 ms entre cada medida, resultando num experimento com quase uma hora de duração. A sala 1 tem o comportamento mostrado no gráfico abaixo.

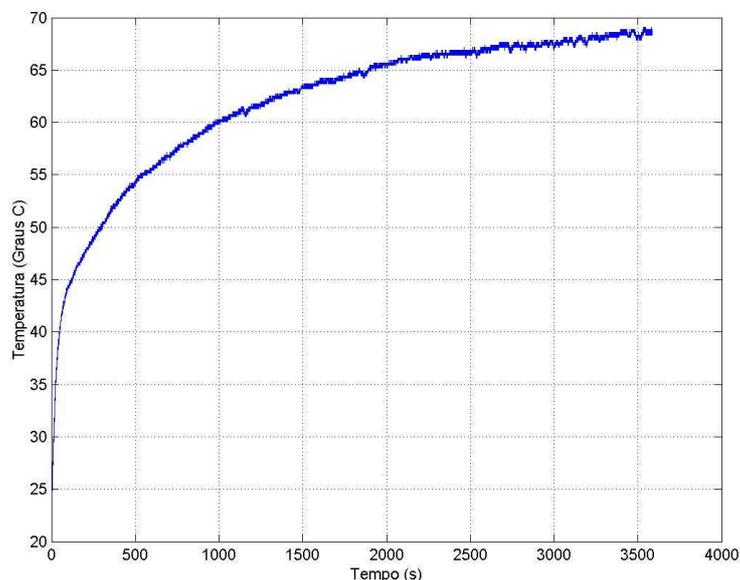


FIGURA 16 – RESPOSTA AO DEGRAU DA SALA 1

Com base na curva da resposta ao degrau obteve-se a constante de tempo da sala 1. A temperatura inicial era de 24,6 °C e quando o sistema se torna estável foi de 68,6 °C,

então a variação foi de 44 °C. 63,7 % deste valor corresponde à aproximadamente 28,0 °C. Para o sistema chegar a temperatura de 52,6 °C levamos 405,7 segundos.

A seguir será mostrada a evolução das temperaturas das outras 4 salas no mesmo período.

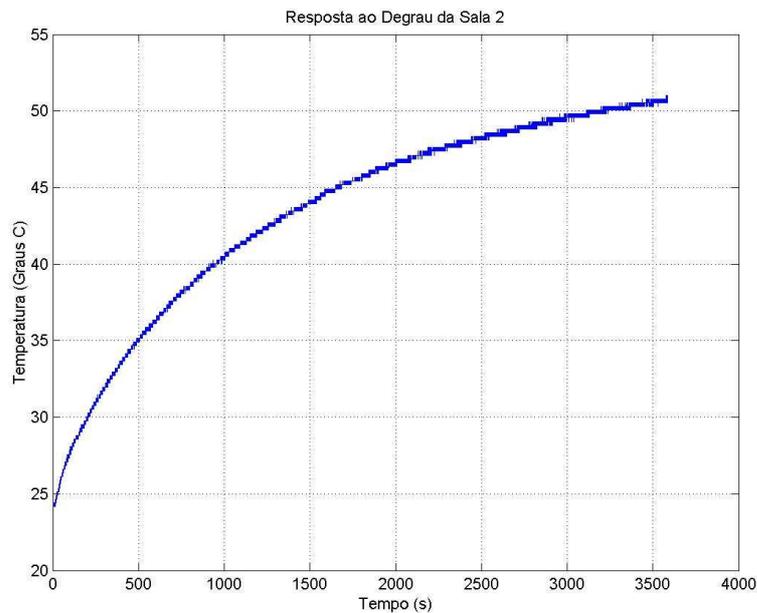


FIGURA 18 – RESPOSTA AO DEGRAU DA SALA 2

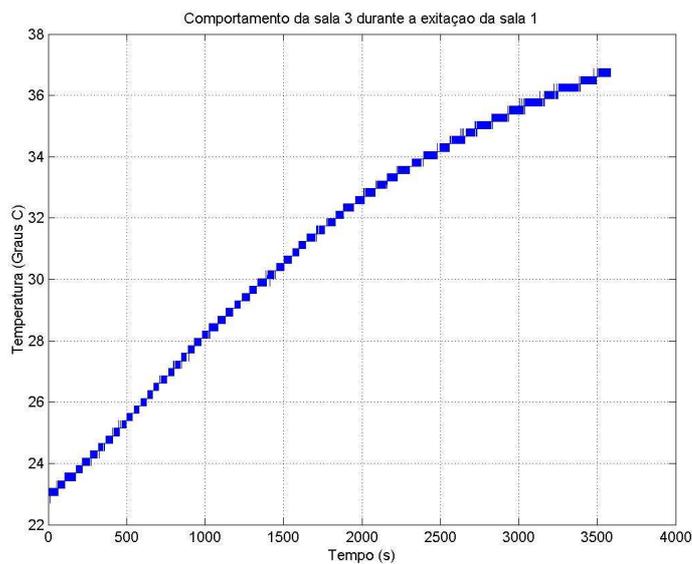


FIGURA 19 – RESPOSTA AO DEGRAU DA SALA 3

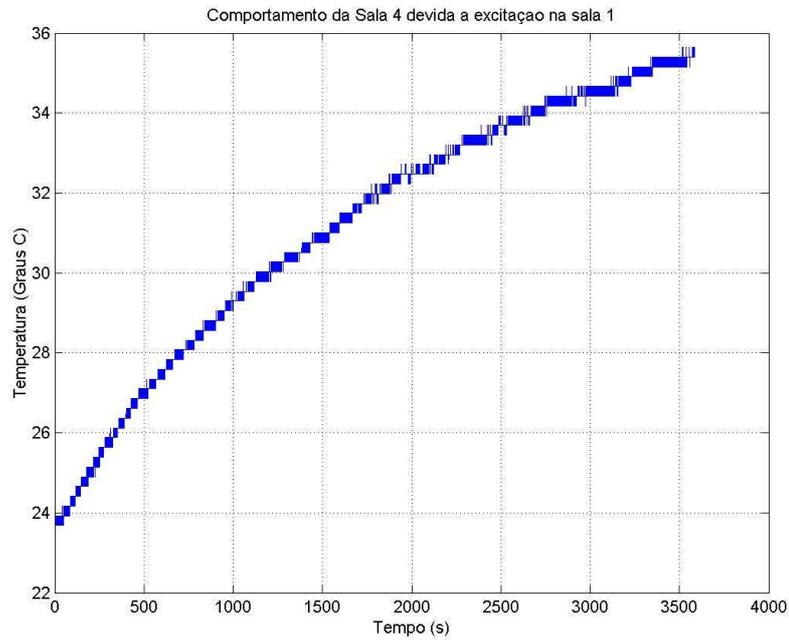


FIGURA 20 – RESPOSTA AO DEGRAU DA SALA 4

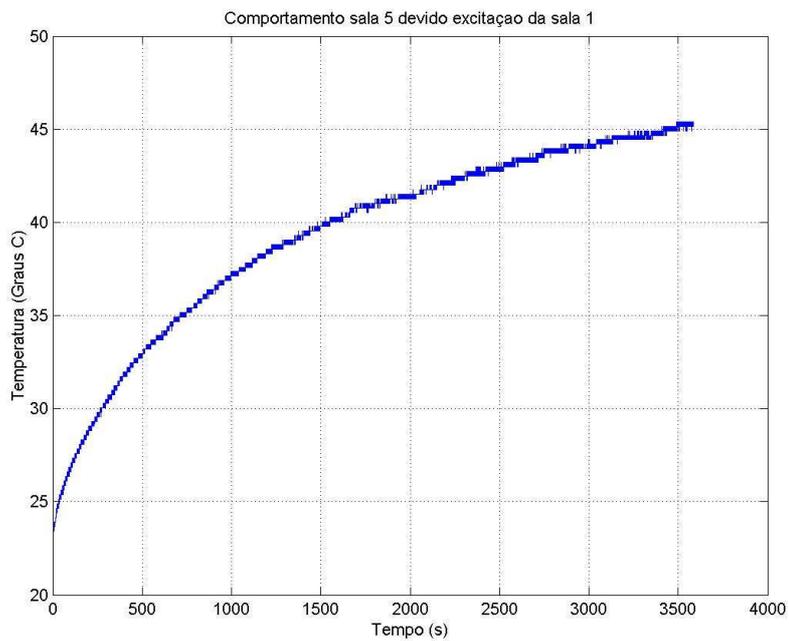


FIGURA 21 – RESPOSTA AO DEGRAU DA SALA 5

A interferência da temperatura de uma sala sobre a outra pode se tornar um problema quando se controla 2 salas, pois as duas entradas irão interferir em todas as salas,

provocando assim uma perturbação que deve ser levada em conta na hora de se projetar os controladores.

Com base nos dados colhidos podemos obter a função transferência para a sala 1.

$$FT_{Sala1} = \frac{0.686}{405.7s + 1}$$

Para as outras salas também é possível obter uma função transferência considerando que a entrada da sala 1 também atua sobre elas. Para as outras salas obtivemos:

	Constante de Tempo(s)	Ganho (°C / %PWM)
Sala 1	405,7	0,686
Sala2	1110,7	0,508
Sala 3	1731,0	0,368
Sala 4	1568,2	0,354
Sala 5	1053,0	0,454

4.2 Projeto do Controlador PI

O controlador PI é um compensador por atraso de fase. Ele possui um zero em $s = -1/T$ e um polo em $S = 0$. Assim, a característica do controlador PI é um ganho infinito na frequência zero. Isso melhora as características de regime estacionário. Contudo a inclusão de um controle PI ao sistema aumenta em uma unidade o tipo do sistema compensado e isto faz com que o sistema compensado seja menos estável, e eventualmente se torne instável.

Devido a isso os valores de K_p e de T_i devem ser escolhidos cuidadosamente de modo a assegurar uma resposta transitória conveniente.

Por este ser um sistema lento um menor tempo de subida seria interessante. Os parâmetros escolhidos foram:

- Sobresinal de 2 %;
- Tempo de Subida de 250 segundos.

Para esses parâmetros devemos ter:

- ξ (Amortecimento) = 0,8482;
- $\omega_n = 0,0195$.

Com isso os pólos dominantes são:

$$P = 0,00827 \pm i0,01761$$

A função transferência em malha fechada do sistema controlado é:

$$FT(s) = \frac{KpK}{Ti\tau} \frac{(Tis + 1)}{s^2 + \left(\frac{1 + KpK}{\tau}\right)s + \frac{KpK}{Ti\tau}}$$

Com isso temos:

$$T_I = 37,00$$

$$K_I = 8,30$$

O controlador aumenta o tempo de subida, mas em contra partida provoca um sobresinal acentuado.

O microcontrolador realiza uma instrução em 1 μ s. A função que implementa o PID (e o PI com a derivativa igual a zero) foi toda programada em *assembly*. E como o PIC é um controlador de arquitetura RISC todas suas instruções tem o mesmo “tamanho”.

Observando o código, podemos ver que o número de instruções chega a pouco menos de 100. Aproximando para 100 instruções temos 100 μ s, que em comparação com constante de tempo na ordem de minutos se torna desprezível.

4.3 Projeto do Controlador Fuzzy

No projeto de um controlador *fuzzy*, passos distintos devem ser devidamente seguidos obedecendo critérios a fim de tornar o controlador mais eficiente. Deve-se, portanto:

- Definir as funções de pertinência de entrada e de saída;
- Definir a base de regras adotada;
- Definir os métodos de implicação, agregação e defuzzificação.

Como tratar cada uma dessas etapas dentro do programa desenvolvido está sucintamente descrito no apêndice A. A análise, neste momento, está relacionada ao projeto do controlador em si, escolha de suas variáveis e funções.

As variáveis de entrada escolhidas são a **temperatura de referência da região** (Região 1 ou 2 da maquete), as **temperaturas das salas**, e as **temperaturas externas**. São variáveis de saída do controlador *fuzzy*, o **ciclo de trabalho do PWM** no atuador 1 e 2.

A cada uma das variáveis descritas anteriormente, são associadas variáveis lingüísticas por meio de funções de pertinência. Um número prático de funções de pertinência utilizado varia entre 2 e 7. Para o presente projeto, foram utilizadas 5 funções de pertinência para cada variável de entrada e 6 para a variável de saída. Procurou-se, também, obter um valor de cerca de 50% de superposição entre as funções de pertinência, o que, segundo experiências realizadas na área de inteligência artificial, é um número razoável para uma boa precisão do sistema [1].

As funções utilizadas, seus formatos e valores são ilustrados pelas figuras a seguir:

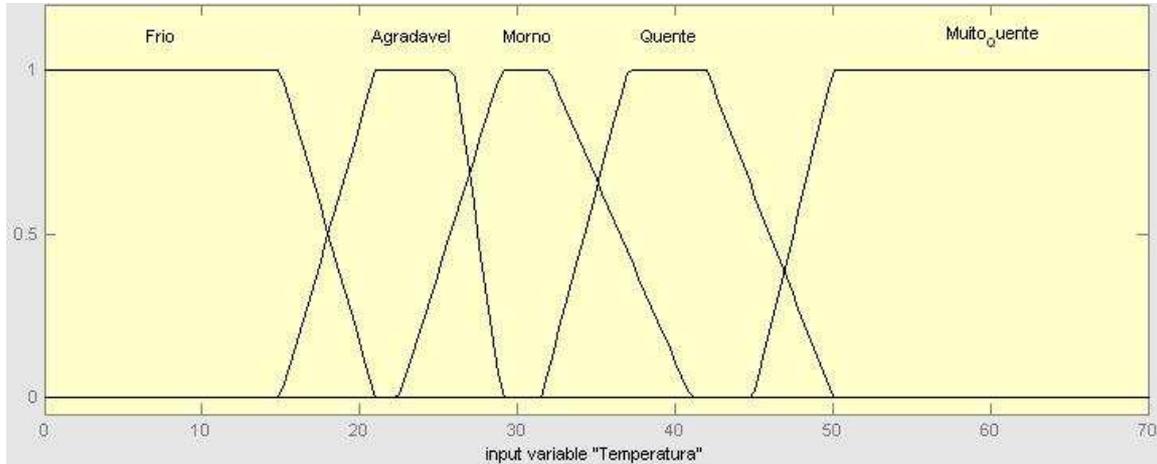


FIGURA 22 - FUNÇÕES DE PERTINÊNCIA DE ENTRADA TEMPERATURA

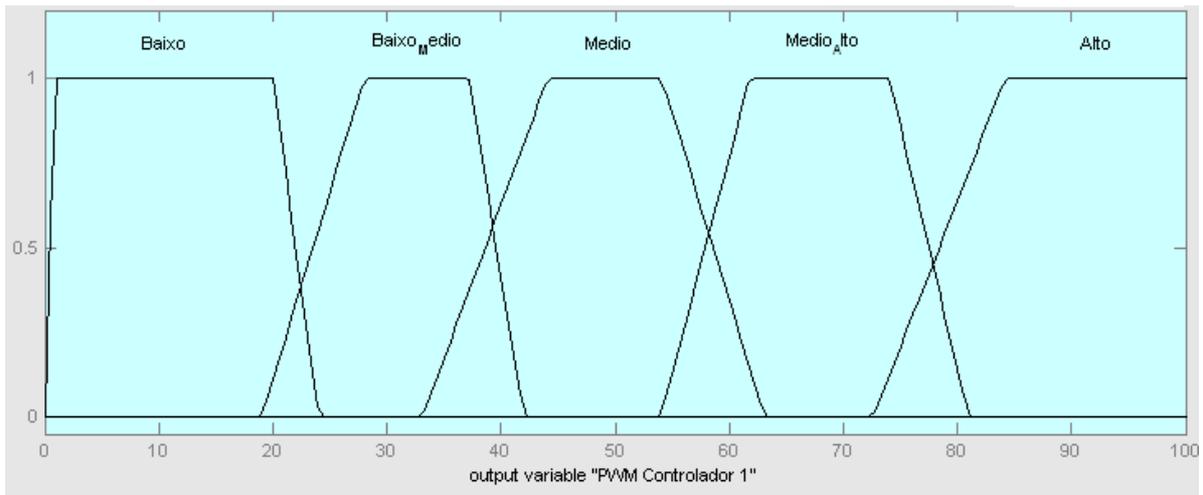


FIGURA 23 - FUNÇÕES DE PERTINÊNCIA DE SAÍDA CICLO DE TRABALHO DO PWM

A grandeza física temperatura foi tratada pelas mesmas funções de pertinência para todas as salas e regiões externas.

Experiências com o sistema mostraram que a temperatura de estabilização dificilmente ultrapassa 70°C. Como os atuadores são capazes apenas de fornecer calor ao sistema, a faixa de temperatura escolhida para o controle encontra-se entre 21°C e 60°C. Assim sendo, nesta faixa de temperatura, as funções de pertinência encontram-se mais densas, garantindo uma maior sensibilidade nesta região, enquanto ajustes mais grosseiros são permitidos fora da região de atuação do controle.

As mesmas funções de pertinência são consideradas para os dois processos, porém a base de regras para cada um destes é diferente. Isto implica em uma atuação diferente para cada região da maquete. O uso da linguagem *assembly* permitiu que um número significativo de regras fosse armazenado no microcontrolador PIC16F677A .

O conjunto de regras forma o banco de regras. Foram definidas 35 regras, tendo como princípio a redução do consumo de energia e a eficiência do controlador. O número de variáveis do sistema permite que seja implementado um banco de regras maior, porém procurou-se tratar apenas as regras mais relevantes de acordo com o comportamento do sistema real. A seguir, estas regras são apresentadas:

Controlador 1

1. **SE** <Referência Agradável > **E** <Sala1 Frio> **ENTÃO** <PWM médio>
2. **SE** <Referência Agradável > **E** <Sala1 Agradável> **ENTÃO** <PWM baixo>
3. **SE** <Referência Morno > **E** <Sala1 Frio> **ENTÃO** <PWM Médio Alto>
4. **SE** <Referência Morno > **E** <Sala1 Frio> **ENTÃO** <PWM médio alto>
5. **SE** <Referência Morno > **E** <Sala1 Agradável> **E** < Externa Agradável> **ENTÃO** <PWM baixo médio>
6. **SE** <Referência Morno > **E** <Sala1 Agradável> **E** < Externa Frio> **ENTÃO** <PWM médio>
7. **SE** <Referência Morno > **E** <Sala1 Morno> **ENTÃO** <PWM baixo>
8. **SE** <Referência Morno > **E** <Sala1 Quente> **ENTÃO** <PWM baixo>
9. **SE** <Referência Morno > **E** <Sala1 Muito Quente> **ENTÃO** <PWM baixo>
10. **SE** <Referência Quente > **E** <Sala1 Frio> **ENTÃO** <PWM alto>
11. **SE** <Referência Quente > **E** <Sala1 Agradável> **E** <Externa Agradável> **ENTÃO** <PWM médio>
12. **SE** <Referência Quente> **E** <Sala1 Agradável> **E** <Externa Frio> **ENTÃO** <PWM médio alto>
13. **SE** <Referência Quente> **E** <Sala1 Morno> **E** <Externa Frio> **ENTÃO** <PWM médio>
14. **SE** <Referência Quente > **E** <Sala1 Quente> **ENTÃO** <PWM baixo>
15. **SE** <Referência Quente > **E** <Sala1 Muito Quente> **ENTÃO** <PWM baixo>
16. **SE** <Referência Frio> <PWM baixo>
17. **SE** <Referência Quente > **E** <Sala1 Muito Quente> **ENTÃO** <PWM baixo>
18. **SE** <Referência Muito Quente > **E** <Sala1 Muito Quente> **ENTÃO** <PWM baixo>
19. **SE** <Referência Muito Quente > **E** <Sala1 Quente> **ENTÃO** <PWM médio baixo>
20. **SE** <Referência Muito Quente > **E** <Sala1 Morno> **ENTÃO** <PWM médio>
21. **SE** <Referência Muito Quente > **E** <Sala1 Agradável> **ENTÃO** <PWM alto médio>
22. **SE** <Referência Muito Quente > **E** <Sala1 Frio> **ENTÃO** <PWM alto>

Controlador 2

23. **SE** <Referência Agradável > **E** <Sala5 Frio> **ENTÃO** <PWM médio>
24. **SE** <Referência Agradável > **E** <Sala5 Agradável> **ENTÃO** <PWM baixo>
25. **SE** <Referência Morno > **E** <Sala5 Frio> **ENTÃO** <PWM Médio Alto>
26. **SE** <Referência Morno > **E** <Sala5 Frio> **ENTÃO** <PWM médio alto>
27. **SE** <Referência Morno > **E** <Sala5 Agradável> **E** < Sala3 Agradável> **ENTÃO** <PWM médio>
28. **SE** <Referência Morno > **E** <Sala5 Agradável> **E** < Sala 4 Agradável> **ENTÃO** <PWM baixo médio>
29. **SE** <Referência Morno > **E** <Sala5 Agradável> **E** < Externa Agradável> **ENTÃO** <PWM baixo médio>
30. **SE** <Referência Morno > **E** <Sala5 Agradável> **E** < Externa Frio> **ENTÃO** <PWM alto médio>
31. **SE** <Referência Quente> **E** <Sala3 Agradável> **E** < Externa Frio> **ENTÃO** <PWM alto médio>
32. **SE** <Referência Quente > **E** <Sala4 Agradável> **E** < Externa Frio> **ENTÃO** <PWM alto médio>
33. **SE** <Referência Morno > **E** <Sala5 Morno> **ENTÃO** <PWM baixo>
34. **SE** <Referência Morno > **E** <Sala5 Quente> **ENTÃO** <PWM baixo>
35. **SE** <Referência Morno > **E** <Sala5 Muito Quente> **ENTÃO** <PWM baixo>
36. **SE** <Referência Quente > **E** <Sala5 Frio> **ENTÃO** <PWM alto>
37. **SE** <Referência Quente > **E** <Sala5 Agradável> **E** <Externa Agradável> **ENTÃO** <PWM médio>
38. **SE** <Referência Quente> **E** <Sala5 Agradável> **E** <Externa Frio> **ENTÃO** <PWM médio alto>
39. **SE** <Referência Quente> **E** <Sala5 Morno> **E** <Externa Frio> **ENTÃO** <PWM médio>
40. **SE** <Referência Quente > **E** <Sala5 Quente> **ENTÃO** <PWM baixo>
41. **SE** <Referência Quente > **E** <Sala5 Muito Quente> **ENTÃO** <PWM baixo>
42. **SE** <Referência Frio> <PWM baixo>
43. **SE** <Referência Quente > **E** <Sala5 Muito Quente> **ENTÃO** <PWM baixo>
44. **SE** <Referência Muito Quente > **E** <Sala5 Muito Quente> **ENTÃO** <PWM baixo>
45. **SE** <Referência Muito Quente > **E** <Sala5 Quente> **ENTÃO** <PWM médio baixo>
46. **SE** <Referência Muito Quente > **E** <Sala5 Morno> **ENTÃO** <PWM médio>
47. **SE** <Referência Muito Quente > **E** <Sala5 Agradável> **ENTÃO** <PWM alto médio>
48. **SE** <Referência Muito Quente > **E** <Sala5 Frio> **ENTÃO** <PWM alto>

As figuras 24 e 25 foram obtidas por meio da *toolbox* de lógica *fuzzy* do MATLAB e ilustram as superfícies de controle para os dois controladores.

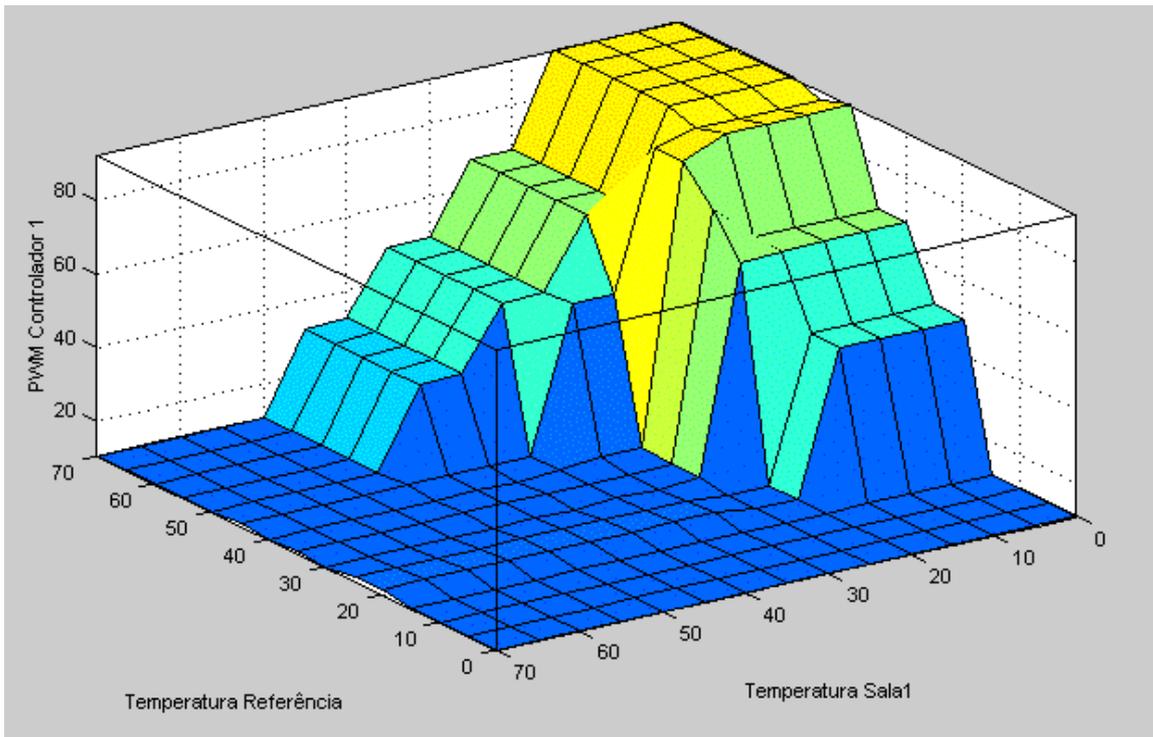


FIGURA 24 – SUPERFÍCIE DE CONTROLE DO CONTROLADOR 1

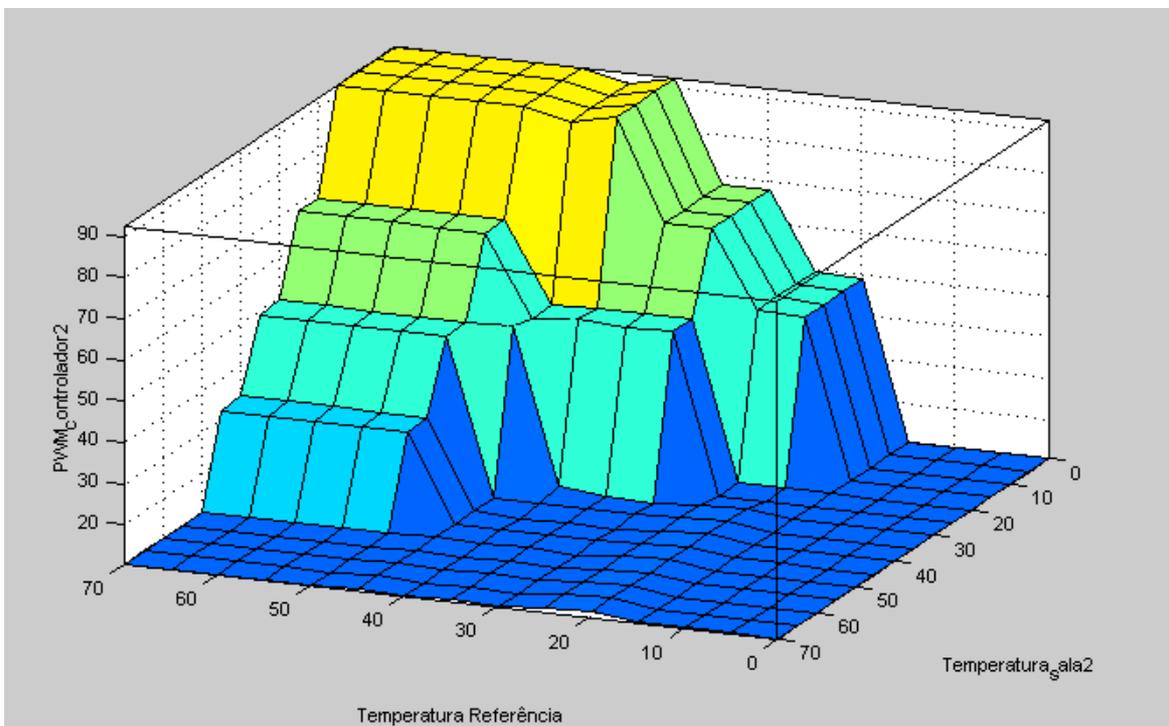


FIGURA 25 – SUPERFÍCIE DE CONTROLE DO CONTROLADOR 2

O ciclo de trabalho do PWM é nulo ou próximo de zero sempre que a temperatura da sala estiver próxima da temperatura de referência e aumenta à medida que a temperatura da sala se afasta da referência.

No algoritmo, testes foram feitos para cada uma das regras a fim de garantir o bom funcionamento do controlador. Uma definição errônea de uma regra, pode prejudicar o desempenho do sistema de controle.

O controlador *fuzzy* utiliza-se da inferência máx-mín. Isso é feito por meio das operações agregação (**E – fuzzy**) e composição (**OU – fuzzy**). O método de defuzzificação é o M- o –M (média do máximo). A tabela a seguir descreve o controlador fuzzy implementado.

TABELA 4 –CARACTERÍSTICAS DO CONTROLADOR *FUZZY*-

Funções de Pertinência	Trapezoidais
Inferência	Máx-mín
Agregação	Mínimo (E – fuzzy)
Composição	Máximo (OU – fuzzy)
Defuzzificação	M – o – M (Média do Máximo)

O esforço computacional que tais características representam foi aspecto relevante quando da escolha dos mesmos. Cálculo de Gaussianas (funções de pertinência) ou de Centróides (defuzzyficação) representam dificuldade na codificação (*assembly*) bem como a exigência de um tempo maior para execução.

4.4 Resultados Experimentais do sistema com controlador PI

Pôde-se fazer medidas com diferentes referências com o controlador PI projetado. Estas medidas foram feitas remotamente via *internet*. A figura a seguir ilustra o comportamento do sistema.

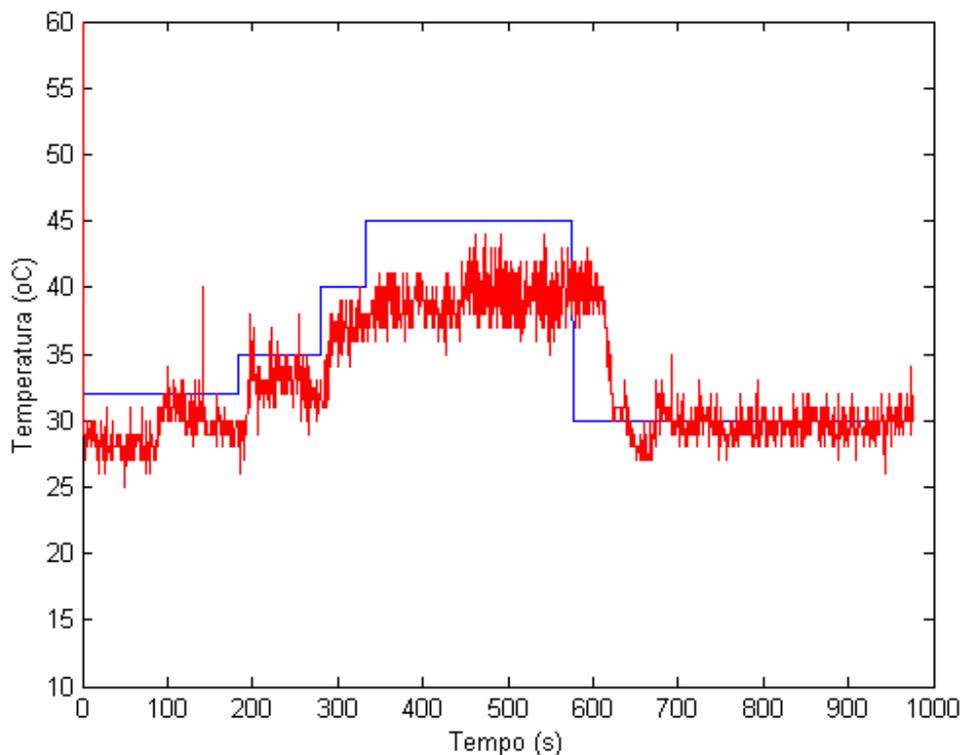


FIGURA 26 – RESPOSTA CONTROLE PI

Pode-se notar que o controlador implementado possui um erro em regime permanente relativamente alto em torno dos 45 °C. O sinal apresenta-se bastante ruidoso, o que dificulta uma análise mais precisa.

4.5 Resultados Experimentais dos Sistemas de Controle *fuzzy*

Devido a problemas na placa de aquisição de dados PCI 6024E, houve grande dificuldade para testes e análise do controlador atuando sobre o sistema. As medições puderam ser feitas por meio de um osciloscópio, gravando-se as medições (pontos) em disquete, porém com a restrição de 500s de duração para um experimento. A especificação do osciloscópio utilizado é mostrada na tabela a seguir:

TABELA 5 – OSCILOSCÓPIO UTILIZADO

Marca	Modelo	Número de Série	Alimentação
Agilent	5462A	MYA0000996	100V ~ 240V

O primeiro ensaio foi realizado com a região representada pelas salas 1 e 2, região externa e controlador 1. Verificou-se como o sistema se comportava ao se “setar” uma determinada temperatura de referência. Inicialmente realizou-se um teste com o controlador 1, temperatura de referência de 30 °C, considerando a região das salas 1 e 2 e temperatura externa. O sinal de referência é “setado” via código *assembly* por meio de um registrador chamado TEMP_REF. Verificou-se que o comportamento do atuador estava de acordo com as regras estabelecidas. O tempo de estabilização foi de aproximadamente 100s. À medida que se alcançava a temperatura de referência, o ciclo de trabalho do PWM diminuía, aproximando-se de 0%. Um baixo valor de ciclo de trabalho era mantido com o objetivo de manter a temperatura da região. A figura 27 ilustra o comportamento do sistema.

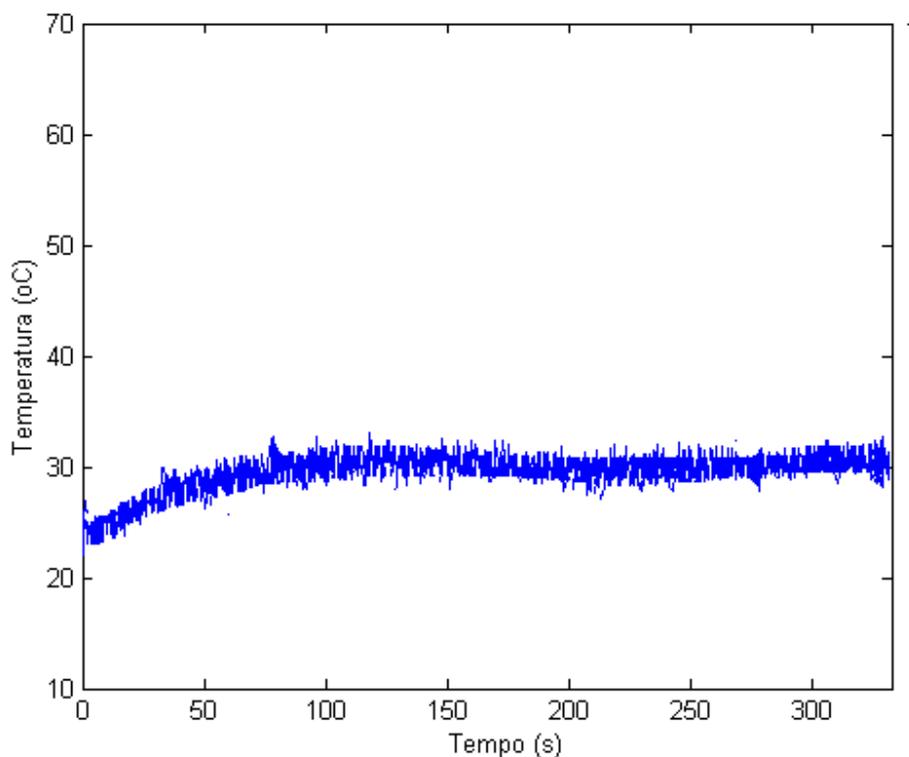


FIGURA 27 – RESPOSTA SISTEMA - REFERÊNCIA 30°C (CONTROLADOR 1)

Uma vez estabilizado o sistema representado pela região supra citada, perturbou-se o sistema, por meio da abertura de portas e janelas da maquete e ligando-se o *cooler*. Acompanhando a temperatura foi possível observar que um pequeno decréscimo de 1°C em relação à referência resultava em um aumento da potência do atuador, evitando assim que a temperatura ficasse abaixo da desejada por muito tempo. Mesmo não sendo possível realizar medidas precisas por meio do osciloscópio, a visualização pôde ser feita um multímetro digital e foram de grande valia.

Um outro ensaio foi realizado com a temperatura de referência em 39 °C. O tempo de estabilização foi cerca de 150 s. Depois de estabilizada a temperatura, foram feitos os mesmos procedimentos visando aplicar perturbações ao sistema . O sistema mostrou-se robusto a tais perturbações. Da mesma forma que no ensaio anterior , um decréscimo de temperatura em relação a referência resultou em um aumento da potência do atuador.

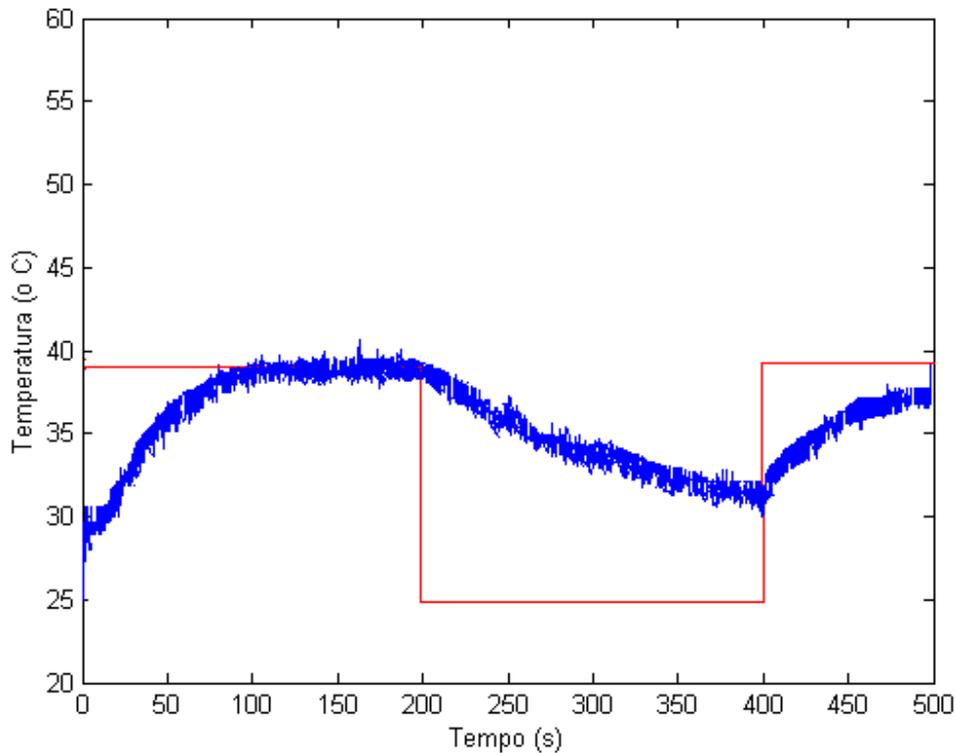


FIGURA28 - RESPOSTA À ONDA QUADRADA (40°C) – CONTROLADOR 1

Ensaaios com temperaturas de referência mais elevadas, a saber, 50°C, 55°C e 60°C resultaram em um tempo de estabilização muito alto. Isso se deve principalmente ao novo modelo de maquete, que possui salas maiores, e às perdas de calor através da tampa de vidro. Um valor de temperatura acima de 50°C é representado pela variável linguística “temperatura muito quente”. Uma aproximação da faixa de temperatura entre 50°C e 70°C foi implementada pois, por se tratar de um sistema com altas constantes de tempo, seria praticamente inviável realizar experimentos de controle remotamente via *Internet*, exigindo do usuário uma grande dedicação de tempo para a realização de um único experimento.

Procedimento similar ao anterior foi realizado para a região representada pelas salas 3,4 e 5. Observou-se que praticamente as mesmas considerações discutidas anteriormente são válidas para este caso, com a exceção, obviamente, do valor do tempo de estabilização. O gráfico da figura 29 ilustra este comportamento do sistema a uma onda quadrada (34°C, valor máximo).

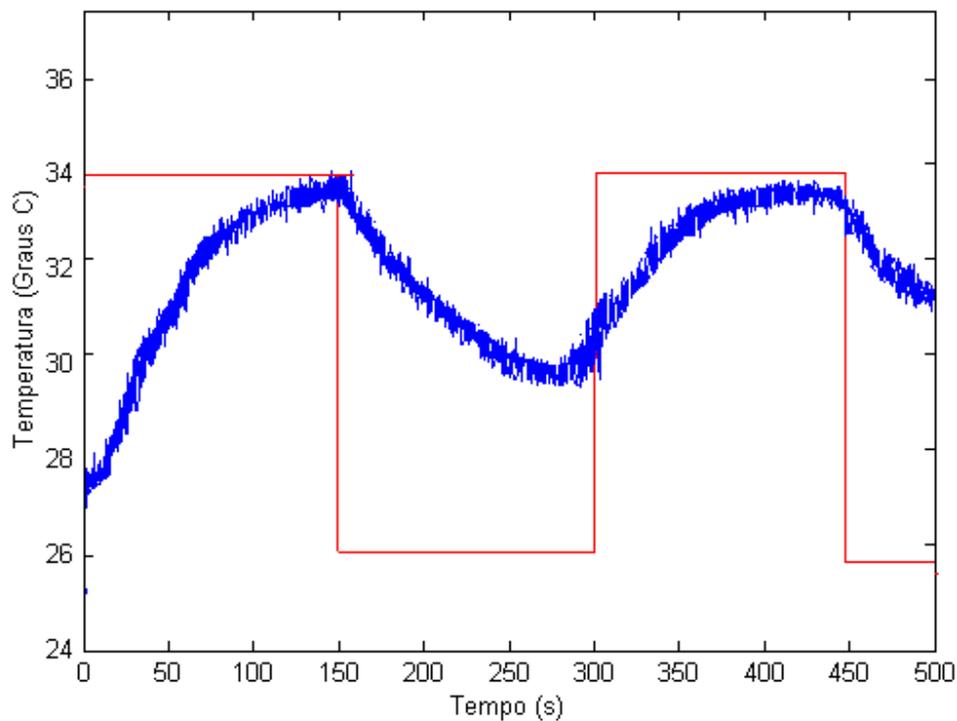


FIGURA 29 – RESPOSTA DO SISTEMA – CONTROLADOR 2

As tabelas a seguir apresentam o tempo de estabilização para os experimentos realizados .

TABELA 5 –COMPORTAMENTO DO SISTEMA – CONTROLADOR 1

Temperatura(°C)	Tempo de estabilização (s)	Comportamento Regime permanente
33	120	
40	840	
>50	>3000	Valor de regime de aprox. 50°C para qualquer referência

TABELA 6 – COMPORTAMENTO DO SISTEMA – CONTROLADOR 2

Temperatura(°C)	Tempo de estabilização (s)	Comportamento Regime permanente
34	150	
40	1700	
>50	>5500	Valor de regime de aprox. 50°C para qualquer referência

Para o sistema do controlador 2, não foi possível realizar medições com o osciloscópio para referências maiores que 37°C devido à constante de tempo.

5 Conclusões

Embora tenham sido feitas simplificações na análise do sistema, como o uso de parâmetros concentrados, o sistema térmico apresenta alguns problemas que dificultam o equacionamento do mesmo. Os parâmetros capacitância e resistência térmicas foram considerados apenas em duas dimensões, o que não ocorre na realidade do sistema. Existe uma perda de calor considerável através da tampa de vidro. Esta tampa representa também uma capacitância e resistência térmica o que resulta em uma dificuldade na estabilização do sistema. Ao se considerar uma região do sistema, como por exemplo, uma sala, este subsistema pode comportar-se como um sistema de primeira ordem, porém ao considerar o sistema como um todo, a ordem do mesmo depende do número de salas pois a cada uma delas está associada um determinado número de parâmetros e uma variável de estado, neste caso, a temperatura.

Outro aspecto a ser considerado é o fato de o sistema permitir diversas formas de perturbação. O ato de abrir e fechar uma janela ou porta, a perturbação de um atuador a uma região vizinha, a transferência de calor entre as paredes, o efeito da temperatura externa são apenas algumas das formas que o sistema pode ser perturbado.

Dentro deste contexto, a opção de um controlador *fuzzy* baseado em regras é bastante plausível. Um controlador *fuzzy* é inerentemente confiável, robusto e principalmente resistente a perturbações externas.

Além disso, o projeto deste controlador não exige um modelamento matemático do sistema. Isso pôde ser observado no decorrer do desenvolvimento do projeto, uma vez que o projeto do controlador *fuzzy* foi realizado paralelamente aos experimentos relacionados às constantes de tempo do sistema. Embora não fosse conhecido o modelo do sistema, uma observação experimental do mesmo possibilitou a especialização de seu comportamento, aspecto decisivo quando da realização de um projeto com lógica *fuzzy*.

Aplicações de controle de temperatura relacionadas a conforto térmico geralmente permitem uma maior mobilidade ao se tratar a variável. O homem normalmente refere-se a uma variação da temperatura de acordo com suas sensações. Se a temperatura em um dia de verão está em torno de 32,54°C, diz-se que o dia está “quente”, ou em um dia de inverno intenso, diz-se que o dia está “muito frio”. Uma faixa entre 22°C e 26°C pode ser

considerada uma faixa onde a temperatura é confortável, dependendo da pessoa. Portanto, para o tipo de aplicação estudada neste projeto, o uso da lógica *fuzzy*, onde as variáveis são tratadas de maneira lingüística, permite a criação regras *fuzzy* de fácil compreensão no sentido de simular o comportamento do sistema e a partir deste ponto tomar decisões de controle.

Neste sentido, o uso de sensores LM35 mostrou-se adequado para o objeto de estudo. Este sensor possui um custo baixo e sua operação é extremamente simples.

Infelizmente os experimentos realizados puderam ser apenas qualitativos e com diversas restrições devido a problemas com a placa de aquisição , mas por meio destes foi possível obter informações úteis a novos experimentos.

O sistema de controle é lento em uma faixa de temperatura maior que 40°C. Em relação ao controlador, isso é um aspecto positivo pois a taxa de amostragem do mesmo é pequena se comparada às constantes de tempo do sistema, o que possibilita um controle *quasi contínuo*. Porém, uma aplicação deste sistema como experimento remoto via Internet exige que a resposta não seja muito demorada, visto que pode ser dispendioso a um usuário com conexão *dial-up*.

As opções de perturbação são interessantes e possibilitam a criação de controladores PID para observação da parte derivativa na ação contra perturbações.

O PIC16F877A mostrou-se bastante eficiente. Neste *microchip*, encontram-se os principais módulos e funções utilizados em um sistema de controle, tais como conversor A/D, módulo PWM, interrupções entre outros. A programação em *Assembly* possibilitou uma economia de memória significativa. O PIC possui quatro bancos de memória e , embora a base de regras tenha sido representada por 18 regras (controlador 1) e 17 regras (controlador 2), um banco de memória não chegou a ser utilizado completamente. Neste sentido, regras podem ser criadas levando em consideração variáveis relacionadas a outras grandezas físicas, como tempo e umidade.

Para futuros trabalhos e implementações poder-se-ia fazer um estudo a respeito do comportamento do sistema quando lhe aplicada alguma perturbação. Verificar como o sistema funcionaria com o ar frio entrando pelas janelas, por exemplo. Outra possibilidade é o estudo do supervisório juntamente com o sistema de controle. Esses dois trabalhos foram feitos separadamente para depois trabalharem em conjunto.

Uma necessidade para verificar a possibilidade ou não do trabalho como um todo seria um profundo estudo dos condicionadores de ar, para saber que tipo de controle se pode fazer nele e como poder aproveitar o que já foi estudado. Isso marcaria uma saída do estudo do protótipo para um sistema real.

6 Referência Bibliográfica

- [1] SHAW, I. S., SIMÕES, M. G., “Controle e Modelagem *Fuzzy*”, Edgard Blücher, São Paulo, 1999.
- [2] OGATA, K., “Engenharia de Controle Moderno”, LTC, Rio de Janeiro, v.3, 1998
- [3] McQUISTON, F. C., PARKER, J. D., “Heating, Ventilating and Air Conditioning, analysis and design”, John Wiley & Sons, Canadá, 1988.
- [4] LOESCH, C., SARI, S. T., “Redes Neurais Artificiais – Fundamentos e Modelos”, Ed. Da FURB, Blumenau, 1996.
- [5] INCROPERA, F. P., DEWITT, D. P., “Fundamentos de transferência de calor e de massa”, LTC, 1998.
- [6] BAUCHSPIESS, A. . “Apostila do Mini-curso da 6ª semana de engenharia elétrica da UnB - Sistemas Inteligentes – Redes Neurais e Lógica Fuzzy”, UnB, Brasília, 2002.
- [7] SOUZA, D. J., “Desbravando o PIC”, Editora Érica Ltda, 2ª ed., São Paulo, 200p, 2000.
- [8] SOUZA, D. J., “Conectando o PIC – Explorando Recursos Avançados”, Editora Érica Ltda, 1ª ed., São Paulo, 200p, 2001.
- [9] XVII edição do Prêmio Jovem Cientista e Prêmio Jovem Cientista do Futuro: Energia Elétrica: geração, transmissão, distribuição e uso racional –Publicação Resumida dos trabalhos vencedores. Porto Alegre: Grupo Gerdau; Brasília: CNPq; Rio de Janeiro: Fundação Roberto Marinho, 2002
- [10] <http://www.criseenergetica.com.br/tudosobre/motivocrise.htm>
- [11] Data sheet do sensor de temperatura LM35.

APÊNDICE A – ALGORITMO FUZZY

A lógica de tomada de decisões usa implicações fuzzy para simular decisões humanas. Isto é feito por meio da estrutura de inferência da base de regras. Assim, ações de controle – consequentes – são geradas e inferidas a partir de um conjunto de decisões de entrada – antecedentes.

A lógica de inferência que representa o sistema térmico em estudo é chamada de inferência máx-mín. No algoritmo, algumas etapas precisam ser seguidas criteriosamente de forma a realizar a lógica de inferência corretamente. A fim de exemplificar como estas etapas são realizadas, considere que o controlador *fuzzy* contenha, dentre suas regras, as seguintes :

Regra 1:

SE Temperatura referência = quente **E** Temperatura sala 1= quente **ENTÃO** PWM = baixo

Regra 2:

SE Temperatura referência = quente **E** Temperatura sala 1= frio **ENTÃO** PWM = alto

Primeiramente, as variáveis de entrada *fuzzy* (antecedentes) são combinadas utilizando o operador mín, correspondente ao conectivo **E** (regras de intersecção fuzzy). Suponha que o valor medido pelo sensor de temperatura da sala 1 seja 65°C e a temperatura de referência seja , 25°C, o que corresponde, internamente no PIC, aos valores 130 e 50, respectivamente. Para a regra 1, o valor de entrada 130 corresponde , para o antecedente “temperatura da sala 1 = QUENTE “, a um grau de pertinência igual a 204 (equivalente a 0,8), enquanto o valor de entrada 50 , para o antecedente “temperatura referência = quente” , um grau de pertinência igual a 77(equivalente a 0,302). Usando o operador **E**, obtêm-se o menor valor entre os graus de pertinência previamente calculados e trunca-se o valor do grau de pertinência do consequente “PWM = baixo” neste valor. A esta operação dá-se o nome de agregação. O mesmo procedimento é realizado para a regra 2 e a todas as regras que porventura estiverem no banco de regras.

A operação realizada anteriormente é ilustrada pela figura a seguir:

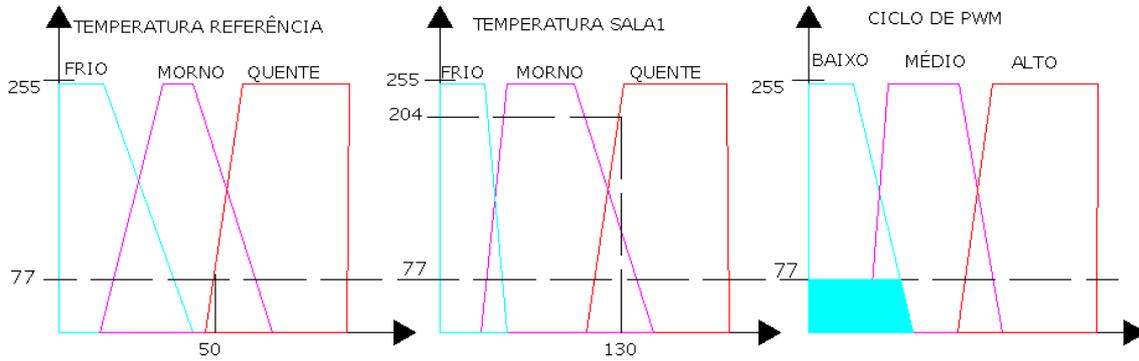
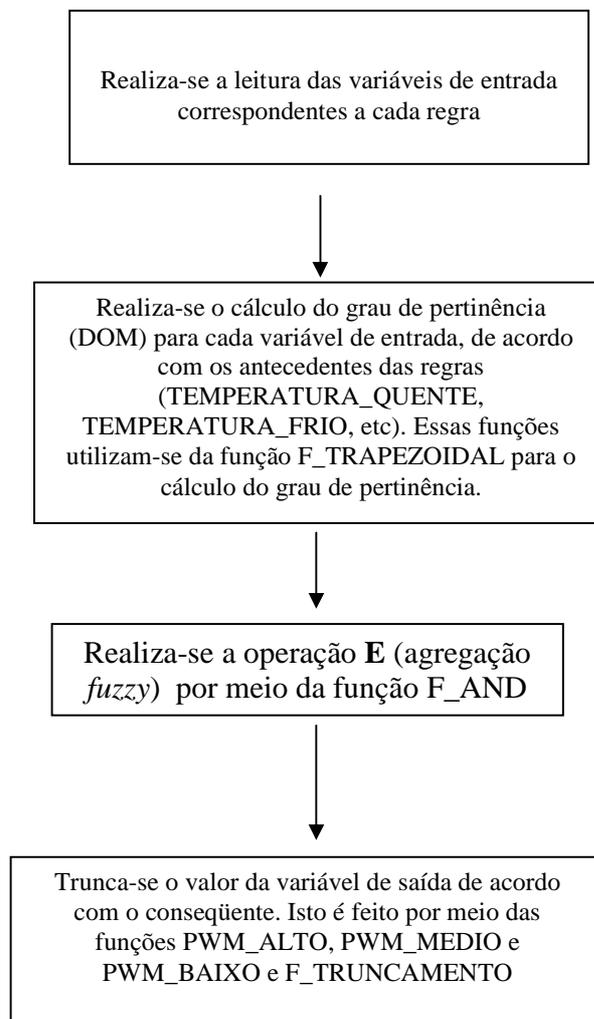


FIGURA 30 – REGRAS FUZZY

No algoritmo, esta etapa é realizada da seguinte forma:



APÊNDICE B – PROCEDIMENTOS PARA O USO DO MPLAB

As principais etapas para o desenvolvimento de um programa no MPLAB são:

- **Escolha de um modo de desenvolvimento:** Necessário para que o MPLAB possa saber quais ferramentas serão usadas na execução do programa. Neste ponto, é escolhido o simulador no qual o programa será testado. O simulador escolhido chama-se “MPLAB SIM Simulator” .
- **Criação de um projeto :** Proceder com a escolha do nome do projeto e associá-lo com um arquivo .asm (assembly). Pode-se também modificar o nome do arquivo .hex (hexadecimal) no qual o MPLAB construirá o código de máquina referente ao programa.
- **Escrever e traduzir o programa para linguagem de máquina (.hex):** Uma vez escrito o programa, é necessário traduzi-lo para a linguagem de máquina do PIC. Isso é feito por meio dos comandos *Project -> Build, Build All* ou *Make Project*.
- **Simulação do Programa:** Pode-se por meio dos comandos em *Debug* simular se o comportamento do programa está apropriado para que a gravação no microcontrolador possa ser efetuada. Nesta etapa, é importante fazer uso de *Watch Windows* para visualizar o conteúdo dos registradores. É possível, também, abrir janelas com o conteúdo dos registradores especiais (SFR – *Special Function Registers*), da memória de programa (ROM) e memória de dados (RAM).
- **Gravação do programa no PIC:** Uma vez implementado e testado, o programa precisa ser gravado no microcontrolador. Isso é feito por meio com o auxílio do uFlash .


```

X1      ;VARIÁVEIS AUXILIARES PARA OPERADORES FUZZY (F_AND)
X2
Z1      ;VARIÁVEIS AUXILIARES PARA OPERADORES FUZZY (F_OR)
Z2
SAIDA_F_AND  ;SAIDA DA FUNÇÃO AGREGAÇÃO FUZZY(AND)
SAIDA_F_OR   ;SAIDA DA FUNÇÃO COMPOSIÇÃO FUZZY(OR)

;*****
;VARIABLES PARA ROTINAS DE ADIÇÃO, MULTIPLICAÇÃO E DIVISÃO
;*****

R1L      ;ACUMULADOR R DE 16 BITS UTILIZADO
R1H      ;NA ROTINA DE ADIÇÃO
AD2L     ;ACUMULADOR AD DE 16 BITS UTILIZADO
AD2H     ;NA ROTINA DE ADIÇÃO
ACCaHI   ; ACUMULADOR a DE 16 BITS UTILIZADO
ACCaLO   ; NA ROTINA DE DIVISÃO
ACCbHI   ; ACUMULADOR b DE 16 BITS UTILIZADO
ACCbLO   ; NA ROTINA DE DIVISÃO
ACCaHI   ; ACUMULADOR c DE 16 BITS UTILIZADO
ACCaLO   ; NA ROTINA DE DIVISÃO
ACCaHI   ; ACUMULADOR d DE 16 BITS UTILIZADO
ACCaLO   ; NA ROTINA DE DIVISÃO
temp     ; CONTADOR TEMPORÁRIO UTILIZADO
         ; NA ROTINA DE DIVISÃO
RESULTHI ; ACUMULADOR DE 16 BITS UTILIZADO
RESULTLO ; P/ RETORNAR O VALOR DA ROTINA
         ; DE MULTIPLICAÇÃO

ENDC

;*****
;*      DEFINIÇÃO DAS VARIÁVEIS INTERNAS DO PIC      *
;*****
#include <P16F877A.INC>      ; MICROCONTROLADOR UTILIZADO
;*****
;*      DEFINIÇÃO DOS BANCOS DE RAM      *
;*****
; OS PSEUDOS-COMANDOS "BANK0" E "BANK1", AQUI DEFINIDOS, AJUDAM A COMUTAR
; ENTRE OS BANCOS DE MEMÓRIA.

#define BANK1 BSF STATUS,RP0      ; SELECIONA BANK1 DA MEMORIA RAM
#define BANK0 BCF STATUS,RP0      ; SELECIONA BANK0 DA MEMORIA RAM

;*****
;*      VETOR DE RESET DO MICROCONTROLADOR      *
;*****
; POSIÇÃO INICIAL PARA EXECUÇÃO DO PROGRAMA

ORG 0X0000      ; ENDEREÇO DO VETOR DE RESET
GOTO CONFIG     ; PULA PARA CONFIG DEVIDO A REGIÃO
               ; DESTINADA AS ROTINAS SEGUINTES

;*****
;LEITURA DO VALOR DE TEMPERATURA - NÃO ESQUECER DE CONFIGURAR O CONVERSOR A/D POR MEIO DE
;AJUSTES NO VALOR DO REGISTRADOR ADCON0
;*****

LEITURA_SENSORES_TEMPERATURA

LER_TEMP      MACRO
BANK0
MOVWF ADCON0
BSF ADCON0,GO      ;INICIA CONVERSÃO A/D
BTFSC ADCON0,GO    ;TERMINOU A CONVERSÃO?
GOTO $-1           ;NÃO - VOLTA 1 INSTRUÇÃO
                 ;SIM

MOVF ADRESH,W     ;MOVE O VALOR DA CONVERSÃO PARA WORK

```

```
ENDM

;LEITURA DA TEMPERATURA DA SALA 1
  MOVLW B'10000001'          ;VELOCIDADE CONVERSOR A/D
                              ;VELOCIDADE -> FOSC/32
                              ;CANAL0
                              ;MODULO LIGADO

  LER_TEMP
  MOVWF TEMP_SALA1          ;TEM-SE A TEMPERATURA DA SALA 1

;LEITURA DA TEMPERATURA DA SALA 2
  MOVLW B'10001001'          ;VELOCIDADE CONVERSOR A/D
                              ;VELOCIDADE -> FOSC/32
                              ;CANAL1
                              ;MODULO LIGADO

  LER_TEMP
  MOVWF TEMP_SALA2

;LEITURA DA TEMPERATURA DA SALA 3
  MOVLW B'10010001'          ;VELOCIDADE CONVERSOR A/D
                              ;VELOCIDADE -> FOSC/32
                              ;CANAL2
                              ;MODULO LIGADO

  LER_TEMP
  MOVWF TEMP_SALA3

;LEITURA DA TEMPERATURA DA SALA 4
  MOVLW B'10011001'          ;VELOCIDADE CONVERSOR A/D
                              ;VELOCIDADE -> FOSC/32
                              ;CANAL3
                              ;MODULO LIGADO

  LER_TEMP
  MOVWF TEMP_SALA4

;LEITURA DA TEMPERATURA DA SALA 5
  MOVLW B'10100001'          ;VELOCIDADE CONVERSOR A/D
                              ;VELOCIDADE -> FOSC/32
                              ;CANAL4
                              ;MODULO LIGADO

  LER_TEMP
  MOVWF TEMP_SALA5

;LEITURA DA TEMPERATURA EXTERNA 1
  MOVLW B'10101001'          ;VELOCIDADE CONVERSOR A/D
                              ;VELOCIDADE -> FOSC/32
                              ;CANAL5
                              ;MODULO LIGADO

  LER_TEMP
  MOVWF TEMP_EXT1

;LEITURA DA TEMPERATURA EXTERNA 2
  MOVLW B'10110001'          ;VELOCIDADE CONVERSOR A/D
                              ;VELOCIDADE -> FOSC/32
                              ;CANAL6
                              ;MODULO LIGADO

  LER_TEMP
  MOVWF TEMP_EXT2

RETURN
;*****
;FUNÇÃO SOMA - 16 BITS
;SOMA O VALOR PASSADO POR R1H,L COM AD2H,L . O RESULTADO É ARMAZENADO EM AD2H,L
;*****
```

SOMA16B

```

MOVFW R1L ;
ADDWF AD2L,F ;LS byte

MOVFW R1H ;MS byte
SKPNC
    INCFSZ R1H,W
    ADDWF AD2H,F ;
RETURN
    
```

```

;*****
;
;          ROTINA DE DIVISÃO
;*****
;
;          Double Precision Division
;*****
; Division : ACCb(16 bits) / ACCa(16 bits) -> ACCb(16 bits) with
;           Remainder in ACCc (16 bits)
; (a) Load the Denominator in location ACCaHI & ACCaLO ( 16 bits )
; (b) Load the Numerator in location ACCbHI & ACCbLO ( 16 bits )
; (c) CALL D_divF
; (d) The 16 bit result is in location ACCbHI & ACCbLO
; (e) The 16 bit Remainder is in locations ACCcHI & ACCcLO
;*****
    
```

D_divF

```

MOVLW .16
MOVWF temp ; CARREGA CONTADOR PARA DIVISÃO

MOVF ACCbHI,W
MOVWF ACCdHI
MOVF ACCbLO,W
MOVWF ACCdLO ; SALVA ACCb EM ACCd

CLRF ACCbHI
CLRF ACCbLO ; LIMPA ACCb

CLRF ACCcHI
CLRF ACCcLO ; LIMPA ACCc

DIV

BCF STATUS,C
RLF ACCdLO,F
RLF ACCdHI,F
RLF ACCcLO,F
RLF ACCcHI,F
MOVF ACCaHI,W
SUBWF ACCcHI,W ;check if a>c
BTFSZ STATUS,Z
GOTO NOCHK
MOVF ACCaLO,W
SUBWF ACCcLO,W ;if msb equal then check lsb

NOCHK

BTFSZ STATUS,C ;carry set if c>a
GOTO NOGO
MOVF ACCaLO,W ;c-a into c
SUBWF ACCcLO,F
BTFSZ STATUS,C
DECF ACCcHI,F
MOVF ACCaHI,W
SUBWF ACCcHI,F
BSF STATUS,C ;shift a 1 into b (result)

NOGO

RLF ACCbLO,F
    
```

```

RLF      ACCbHL,F

DECFSZ  temp,F          ; FIM DA DIVISÃO ?
GOTO    DIV            ; NÃO - VOLTA P/ DIV
                          ; SIM
RETURN                               ; RETORNA

;*****
; MULTIPLICAÇÃO 8 bit x 8 bit = 16 bit (unsigned)
; TEMPO DE EXECUÇÃO: 38 CICLOS
; TAMANHO DO CÓDIGO: 35 words
; MULTIPLICADOR: w
; MULTIPLICANDO: resultlo
; RESULTADO: resulthi:resultlo
;*****

MUL8X8          ;* 2 CICLOS PARA CHAMAR INSTRUÇÃO

MULT MACRO
BTFSZ STATUS,C
ADDWF RESULTHI,F
RRF RESULTHI,F
RRF RESULTLO,F
ENDM

CLRf RESULTHI    ;* 1 CICLO
RRF RESULTLO,F  ;* 1 CICLO

MULT             ;* 4 CICLOS

RETURN

;*****
;*                               FUNÇÕES FUZZY                               *
;*****

;*****
; FUNÇÃO TRAPEZOIDAL
;ESTA FUNÇÃO CALCULA UMA FUNÇÃO DE PERTINÊNCIA TRAPEZOIDAL. DEVEM SER PASSADOS COMO
;PARÂMETROS PONTOS BESQ,AESQ,ADIR,BDIR E M. A FUNÇÃO RETORNA O VALOR DA FUNÇÃO u(x)
;(SAIDA_TRAPEZOIDAL) CORRESPONDENTE AO PONTO PASSADO COMO PARÂMETRO.RETORNA O VALOR DO
;SAIDA DA DEGREE OF MEMBERSHIP (DOM)
;
;(255) 1|          AESQ ..... ADIR
;
;          |          .          .
;          |          .          .          PONTO (GENÉRICO NA FUNÇÃO)
;(DOM)  |          .  _|M1  M2|_
;          |          BESQ .          BDIR
;          |          .....          .....
;(0)    0|_____
;*****

F_TRAPEZOIDAL

CLRWDt
;*****
;* SE PONTO MENOR QUE BESQ - RETORNA 0 *
;*****
MOVf BESQ,W          ;MOVE O VALOR DO PONTO DESEJADO PARA W
SUBWF PONTO,W        ;

```

```

SKPNC                ;SE PONTO < BESQ
GOTO  ENDIF_T1      ;PONTO NÃO É MENOR, PULA
MOVLW 0X00
MOVWF  DOM
GOTO  FIM_TRAPEZOIDAL

;*****
;* ELSE - BESQ< PONTO <AESQ  *
;*****

ENDIF_T1
;SE NÃO O CASO ANTERIOR (PONTO<=BESQ)
MOVF  AESQ,W
SUBWF PONTO,W
SKPNC ;SE PONTO<=AESQ
GOTO  ENDIF_T2
MOVF  BESQ,W
SUBWF PONTO,W ;RESULTLO = PONTO - BESQ
MOVWF RESULTLO
MOVF  M1,W
CALL  MUL8X8 ; (PONTO - BESQ) *M1
MOVF  RESULTLO,W
MOVWF DOM ;JOGA O RESULTADO EM DOM
GOTO  FIM_TRAPEZOIDAL
ENDIF_T2

;*****
;* ELSE - AESQ =<= PONTO <= ADIR RETORNA 255 (1)*
;*****

;SE NÃO CASO ANTERIOR(PONTO<=AESQ)
MOVF  PONTO,W
SUBWF ADIR,W
SKPC  ;SE PONTO<ADIR
GOTO  ENDIF_T3
MOVLW 0XFF
MOVWF DOM ;RETORNA 255 EM DOM
GOTO  FIM_TRAPEZOIDAL
ENDIF_T3

;*****
;* ELSE - ADIR < PONTO < BDIR *
;*****

;SE NÃO CASO ANTERIOR (PONTO<=ADIR)
MOVF  PONTO,W
SUBWF BDIR,W
SKPC  ;SE PONTO<=AESQ
GOTO  ENDIF_T4
MOVF  PONTO,W
SUBWF BDIR,W
MOVWF RESULTLO ;RESULTLO = BDIR - PONTO
MOVF  M2,W
CALL  MUL8X8 ; (BDIR - PONTO)* M2
MOVF  RESULTLO,W
MOVWF DOM ;JOGA O RESULTADO EM DOM
GOTO  FIM_TRAPEZOIDAL
ENDIF_T4

;*****
;* ELSE - PONTO > BDIR RETORNA 0 *
;*****
;SE NENHUMA DAS ANTERIORES (PONTO>BDIR)
MOVLW 0X00
MOVWF DOM ;RETORNA 0 EM DOM

FIM_TRAPEZOIDAL

RETURN

```

```

;*****
;FUNÇÃO TRUNCAMENTO
;CALCULA, A PARTIR DE UM VALOR DE PERTINÊNCIA (DOM - DEGREE OF MEMBERSHIP), O VALOR DE SAÍDA
;(FUNÇÃO DE PERTINÊNCIA DE SAÍDA) PARA REALIZAR A DEFUZZIFICAÇÃO MÉDIA DO MÁXIMO.
;PARÂMETROS DE ENTRADA:
;BES,AESQ,ADIR,BDIR - PARÂMETROS DA FUNÇÃO DE PERTINÊNCIA DE SAÍDA
;DOM - VALOR DE PERTINÊNCIA
;PARÂMETROS DE SAÍDA
;S1,S2 - VALORES DA FUNÇÃO DE PERTINÊNCIA PARA O VALOR DOM
;*****

```

F_TRUNCAMENTO

```

;MACRO PARA PASSAR OS PARÂMETROS RESULTADOS DA MULTIPLICAÇÃO PARA
;DIVISÃO

```

```

TRANSFORMA MACRO
MOVF RESULTHI,W
MOVWF ACCbHI
MOVF RESULTLO,W
MOVWF ACCbLO
MOVLW 0xFF ;255
MOVWF ACCaLO
MOVLW 0X00
MOVWF ACCaHI
ENDM ;FIM DA MACRO

```

```

;*****
;* SE DOM = 0
;*****
MOVLW 0x00
SUBWF DOM,W
SKPZ ;SE DOM = 0
GOTO ENDFDEFUZZ1
MOVF BDIR,W
MOVWF S1
MOVF BESQ,W
MOVWF S2
GOTO ENDF_DEFUZZ

```

```

;*****
;* ELSE SE 0<DOM<1 *
;*****
ENDFDEFUZZ1 ;ELSE (0<DOM<1(255))

```

```

;*****
;* RETORNO DE S1 *
;*****
MOVF DOM,W
MOVWF RESULTLO
ADDLW 1
SKPNC
GOTO ENDFDEFUZZ2
MOVF BESQ,W
SUBWF AESQ,W ;W = (AESQ - BESQ) -> AESQ>BESQ
CALL MUL8X8 ; RESULTHI:RESULTLO = DOM * (AESQ - BESQ)
TRANSFORMA
CALL D_divF
MOVF ACCbLO,W
ADDWF BESQ,W ;W = DOM*(AESQ - BESQ)/255 + BESQ
MOVWF S1

```

```

;*****
;* RETORNO DE S2 *
;*****

```

```

MOVF DOM,W
MOVWF RESULTLO
MOVF ADIR,W

```

```

SUBWF  BDIR,W  ;W = BDIR - ADIR
CALL   MUL8X8 ;RESULTHI:RESULTLO = DOM*(BDIR - ADIR)
TRANSFORMA
CALL   D_divF
MOVF  ACCbLO,W      ;W = ACCbLO = DOM*(BDIR - ADIR)/255
SUBWF  BDIR,W
MOVWF  S2

```

```

;*****
;* SE DOM = 1
;*****

```

```

ENDFDEFUZZ2
MOVLW 0xFF
SUBWF  DOM,W
SKPZ           ;SE DOM = 1(255)
GOTO  ENDF_DEFUZZ
MOVF  AESQ,W
MOVWF S1
MOVF  ADIR,W
MOVWF S2

```

ENDF_DEFUZZ

RETURN

```

;*****
; FUNÇÃO MÉDIA DO MÁXIMO
;REALIZA A MÉDIA DOS VALORES MÁXIMOS - DEFUZZIFICAÇÃO
;ENTRADA: VALORES MÁXIMOS S1 E S2
;SAIDA: SFMEDIA
;*****

```

F_MEDIA_DO_MAXIMO

```

;*****
;* RETORNO DE (S1+S2)/2 *
;*****

```

```

MOVF  S1,W
MOVWF R1L
MOVF  S2,W
MOVWF AD2L
MOVLW 0X00
MOVWF R1H
MOVWF AD2H
CALL  SOMA16B ;A SOMA PODE ULTRAPASSAR 255, ENTÃO USA-SE SOMA 16BITS
;RETORNA S1 + S2 EM AD2H:AD2L

MOVF  AD2H,W
MOVWF ACCbHI
MOVF  AD2L,W
MOVWF ACCbLO ;CARREGA REGISTRADORES NECESSÁRIOS PARA DIVISÃO
MOVLW 0X02
MOVWF ACCaLO
MOVLW 0X00
MOVWF ACCaHI
CALL  D_divF ;RETORNA (S1 + S2)/2
MOVF  ACCbLO,W
MOVWF SFMEDIA;GRAVA A SAÍDA EM SDEFUZZ

```

RETURN

```

;*****
; FUNÇÃO F_AND - OPERADOR MÍNIMO (AGREGAÇÃO)
;CORRESPONDE AO CONECTIVO E(AND) DE ACORDO COM A INTERSECÇÃO FUZZY
;DEVEM SER PASSADOS COMO PARÂMETROS DOIS VALORES (X1 E X2) QUE SERÃO COMPARADOS. RETORNA O
;MÍNIMO ENTRE X1 E X2 EM SAIDA_FAND
;*****

```

F_AND

```
MOVF X1,W
SUBWF X2,W
SKPNC ;SE X2 < X1
GOTO ENDIF_A1
MOVF X2,W
MOVWF SAIDA_F_AND
GOTO FIM_F_AND
ENDIF_A1 ;ELSE
MOVF X1,W
MOVWF SAIDA_F_AND
FIM_F_AND

RETURN
```

```
*****
;
; FUNÇÃO F_OR - OPERADOR MÁXIMO (COMPOSIÇÃO)
;CORRESPONDE AO CONECTIVO OU(OR) DE ACORDO COM A UNIÃO FUZZY
;DEVEM SER PASSADOS COMO PARÂMETROS DOIS VALORES (Z1 E Z2) QUE SERÃO COMPARADOS.
;RETORNA OS VALORES S1 E S2 REFERENTES AO MAIOR VALOR DE PERTINÊNCIA (Z1 OU Z2)
;O VALOR QUE SE DESEJA COMPARAR DEVE SER JOGADO EM Z1
*****
```

F_OR

```
MOVF Z1,W
SUBWF Z2,W
SKPNC ;IF Z2 < Z1
GOTO FIM_F_OR
;ENDIF_A1
MOVF Z1,W
MOVWF DOM
MOVWF Z2 ;FAZ-SE ESSA OPERAÇÃO COMO GARANTIA CASO HAJA UMA PRÓXIMA
;CHAMADA DESSA FUNÇÃO
CALL F_TRUNCAMENTO
FIM_F_OR

RETURN
```

```
*****
;
; FUNÇÕES DE PERTINÊNCIA DE ENTRADA
*****
```

```
*****
;
; FUNÇÃO DE PERTINÊNCIA - TEMPERATURA SALA FRIO
*****
```

TEMPERATURA_FRIO

```
MOVLW .0
MOVWF BESQ
MOVLW .5
MOVWF AESQ
MOVLW .30
MOVWF ADIR
MOVLW .42
MOVWF BDIR
MOVLW .51
MOVWF M1
MOVLW .28
MOVWF M2
CALL F_TRAPEZOIDAL

RETURN
```

```
*****
```

```
; FUNÇÃO DE PERTINÊNCIA - TEMPERATURA SALA AGRADÁVEL  
;*****
```

TEMPERATURA_AGRADAVEL

```
MOVLW .30  
MOVWF BESQ  
MOVLW .42  
MOVWF AESQ  
MOVLW .52  
MOVWF ADIR  
MOVLW .58  
MOVWF BDIR  
MOVLW .21  
MOVWF M1  
MOVLW .42  
MOVWF M2  
CALL F_TRAPEZOIDAL
```

RETURN

```
;*****  
; FUNÇÃO DE PERTINÊNCIA - TEMPERATURA SALA MORNO  
;*****
```

TEMPERATURA_MORNO

```
MOVLW .45  
MOVWF BESQ  
MOVLW .58  
MOVWF AESQ  
MOVLW .64  
MOVWF ADIR  
MOVLW .82  
MOVWF BDIR  
MOVLW .19  
MOVWF M1  
MOVLW .14  
MOVWF M2  
CALL F_TRAPEZOIDAL
```

RETURN

```
;*****  
; FUNÇÃO DE PERTINÊNCIA - TEMPERATURA SALA QUENTE  
;*****
```

TEMPERATURA_QUENTE

```
MOVLW .63  
MOVWF BESQ  
MOVLW .74  
MOVWF AESQ  
MOVLW .84  
MOVWF ADIR  
MOVLW .100  
MOVWF BDIR  
MOVLW .23  
MOVWF M1  
MOVLW .15  
MOVWF M2  
CALL F_TRAPEZOIDAL
```

RETURN

```
;*****  
; FUNÇÃO DE PERTINÊNCIA - TEMPERATURA SALA MUITO QUENTE  
;*****
```

TEMPERATURA_MUITO_QUENTE

```
    MOVLW .90
    MOVWF BESQ
    MOVLW .100
    MOVWF AESQ
    MOVLW .250
    MOVWF ADIR
    MOVLW .255
    MOVWF BDIR
    MOVLW .25
    MOVWF M1
    MOVLW .51
    MOVWF M2
    CALL F_TRAPEZOIDAL

    RETURN

;*****
;      FUNÇÕES DE PERTINÊNCIA DE SAÍDA
;*****

;*****
;      FUNÇÃO DE PERTINÊNCIA - CICLO PWM DESLIGA
;*****

PWM_DESLIGA

    MOVLW .0
    MOVWF BESQ
    MOVWF AESQ
    MOVWF ADIR
    MOVWF BDIR

    RETURN

;*****
;      FUNÇÃO DE PERTINÊNCIA - CICLO PWM BAIXO
;*****

PWM_BAIXO

    MOVLW .0
    MOVWF BESQ
    MOVLW .2
    MOVWF AESQ
    MOVLW .51
    MOVWF ADIR
    MOVLW .61
    MOVWF BDIR

    RETURN

;*****
;      FUNÇÃO DE PERTINÊNCIA - CICLO PWM BAIXO
;*****

PWM_BAIXO_MEDIO

    MOVLW .48
    MOVWF BESQ
    MOVLW .71
    MOVWF AESQ
    MOVLW .94
    MOVWF ADIR
    MOVLW .107
    MOVWF BDIR

    RETURN
```

```
*****
;
;   FUNÇÃO DE PERTINÊNCIA - CICLO PWM ALTO
;
*****
PWM_MEDIO

    MOVLW .84
    MOVWF BESQ
    MOVLW .112
    MOVWF AESQ
    MOVLW .137
    MOVWF ADIR
    MOVLW .160
    MOVWF BDIR

    RETURN

*****
;
;   FUNÇÃO DE PERTINÊNCIA - CICLO PWM ALTO
;
*****
PWM_MEDIO_ALTO

    MOVLW .137
    MOVWF BESQ
    MOVLW .155
    MOVWF AESQ
    MOVLW .188
    MOVWF ADIR
    MOVLW .206
    MOVWF BDIR

    RETURN

*****
;
;   FUNÇÃO DE PERTINÊNCIA - CICLO PWM ALTO
;
*****
PWM_ALTO

    MOVLW .183
    MOVWF BESQ
    MOVLW .214
    MOVWF AESQ
    MOVLW .250
    MOVWF ADIR
    MOVLW .255
    MOVWF BDIR

    RETURN

*****
;
;   FUNÇÃO DE PERTINÊNCIA - CICLO PWM ALTO
;
*****
PWM_MAX

    MOVLW .183
    MOVWF BESQ
    MOVLW .214
    MOVWF AESQ
    MOVLW .250
    MOVWF ADIR
    MOVLW .255
    MOVWF BDIR

    RETURN

; *****
; *   CONFIGURAÇÕES INICIAIS DE HARDWARE E SOFTWARE   *
; *****
; NESTA ROTINA SÃO INICIALIZADAS AS PORTAS DE I/O DO MICROCONTROLADOR E AS
; CONFIGURAÇÕES DOS REGISTRADORES ESPECIAIS (SFR). A ROTINA INICIALIZA A
; MÁQUINA E AGUARDA O ESTOURO DO WDT.
```

CONFIG

```

CLRF  PORTA  ;GARANTE TODAS AS SAÍDAS EM ZERO
CLRF  PORTB
CLRF  PORTC
CLRF  PORTD
CLRF  PORTE

BANK1

MOVLW B'11111111' ;CONFIGURA I/O DO PORTA
MOVWF TRISA       ;TODOS OS PINOS (RA0:RA5) SÃO R/W

MOVLW B'11111111' ;CONFIGURA I/O DO PORTB
MOVWF TRISB       ;TODOS OS PINOS (RA0:RA5) SÃO R/W

MOVLW B'11111001' ;CONFIGURA I/O DO PORTC
MOVWF TRISC       ;PINOS RC1 E RC2 COMO SAÍDAS (PWM)

MOVLW B'00000000' ; CONFIGURA I/O DO PORTD
MOVWF TRISD       ; PINOS LEITURA (LCD, POR EXEMPLO)

MOVLW B'00000100'
MOVWF TRISE

MOVLW B'11011111' ;CONFIGURA OPTION
MOVWF OPTION_REG  ;PULL-UPS DESABILITADOS
                  ;INTERRUPÇÃO NA PORTA DE SUBIDA
                  ;TMR0 INCREMENTADO PELO CICLO DA MÁQUINA
                  ;PRESCALER APLICADO AO WDT
                  ;WDT 1:128 TMRO 1:1

MOVLW B'00000000'
MOVWF INTCON

MOVLW .255
MOVWF PR2         ;CONFIGURA O PERÍODO DO PWM

MOVLW B'00000000' ;TODOS AS PORTAS USADAS COMO CANAIS ANALÓGICOS DE ENTRADA
MOVWF ADCON1      ;JUSTIFICADA A ESQUERDA

BANK0

MOVLW B'00001111'
MOVWF CCP2CON     ;CONFIGURA CCP2CON PARA PWM (PINO RC1)

MOVLW B'00000111' ;CONFIGURA TMR2
MOVWF T2CON       ;TIMER 2 LIGADO
                  ;PRESCALE 1:16  POSTSCALE 1:1

BANK0

CLRF  CCP2L      ;INICIA COM DUTY CLICLE 0%

;*****
;* ESTOURO DO WDT
;*****

; AS INSTRUÇÕES A SEGUIR FAZEM COM QUE O PROGRAMA TRAVE QUANDO HOUVER UM
; RESET OU POWER-UP, MAS PASSE DIRETO SE O RESET FOR POR WDT. DESTA FORMA,
; SEMPRE QUE O PIC É LIGADO, O PROGRAMA TRAVA, AGUARDA UM ESTOURO DE WDT
; E COMEÇA NOVAMENTE. ISTO EVITA PROBLEMAS NO START-UP DO PIC.
; QUANDO NA SIMULAÇÃO, COLOCAR PONTO E VÍRGULA PARA AGILIZAR

BTFSZ STATUS,NOT_TO ; RESET POR ESTOURO DE WATCHDOG TIMER ?
GOTO  $              ; NÃO - AGUARDA ESTOURO DO WDT
                  ; SIM
    
```


APÊNDICE D – TESTE DO DRIVER COM PWM

```
VARIABLE I      ;CONTADOR

;*****
;DELAYS
;*****+
DELAY_MS
    MOVWF      TEMPO1
    MOVLW     .250
    MOVWF      TEMPO0
    CLRWDT
    DECFSZ     TEMPO0,F
    GOTO       $-2

    DECFSZ     TEMPO1,F
    GOTO       $-6
RETURN
DELAY_5S

    I = 0

    WHILE I < .20
        MOVLW .256
        CALL DELAY_MS
        I+=1
    ENDW
RETURN

;*****
;CONFIGURAÇÕES INICIAIS DE HARDWARE E SOFTWARE
;*****
CONFIG
    CLRF      PORTA
    CLRF      PORTB
    CLRF      PORTC
    CLRF      PORTD
    CLRF      PORTE ;RESETA TODAS AS PORTAS

    BANK1     ;SELECIONA BANCO 1 DA MEMÓRIA RAM

    MOVLW    B'11111111'
    MOVWF    TRISA
    MOVLW    B'11111111'
    MOVWF    TRISB
    MOVLW    B'11111101'
    MOVWF    TRISC
    MOVLW    B'00000000'
    MOVWF    TRISD
    MOVLW    B'00000100'
    MOVWF    TRISE
    MOVLW    B'11011111'
    MOVWF    OPTION_REG ;CONFIGURA OPTIONS
                                ;PULLUPS DEDESABILITADOS
                                ;INTER. NA BORDA DE SUBIDA DE RB0
                                ;TIMER0 INCREMENTADO PELO CICLO DE MAQ
                                ;WDT 1:1
                                ;TIMER 1:1

    MOVLW    .255
    MOVWF    PR2 ;CONFIGURACAO DO PERIODO DO PWM
    BANK0    ;SELECIONA BANCO 0 DA RAM
    MOVLW    B'00000111'
    MOVWF    T2CON ;CONFIGURA TMR2 ->LIGADO
                                ;PRESCALE 1:16
                                ;POSTSCALE 1:1

    MOVLW    B'00001111'
    MOVWF    CCP2CON ; CONFIGURA CCP2 PARA PWM
```

```

MOV LW B'00001111'
MOV WF CCP1CON          ; CONFIGURA CCP1 PARA PWM
;*****
;WATCH DOG
;*****
    BTFS  STATUS,NOT_TO
    GOTO  $
;*****
;INICIALIZAÇÃO DA RAM
;*****
    MOV LW 0X20
    MOV WF FSR

LIMPA_RAM
    CLR F INDF          ;LIMPA A POSIÇÃO
    INC F FSR,F        ;INCREMENTA PONTEIRO P/ PROXIMA POSICAO
    MOV F FSR,W
    XOR LW 0X80        ;COMPARA O PONTEIRO COM A ULTIMA POS
    BTFS  STATUS,Z     ;JA LIMPOU TODAS AS POSIÇÕES?
    GOTO  LIMPA_RAM    ;NÃO - LIMPA A PROXIMA POSIÇÃO
                    ;SIM

PWM MACRO
    MOV WF CCP2L
    BCF  CCP2CON,5
    BCF  CCP2CON,4
    CALL DELAY_5S
ENDM

;*****
;PWM COM 0% DE DUTY CICLE (CCP2 - PINO 16) E 100% (CCP1 - PINO 17)
;*****
    LOOP

    CLR F CCP2L        ;ZERA O REGISTRADOR CCP2L
    BCF  CCP2CON,5     ;ZERA OS BITS 5 E 4 DO CCP2CON
    BCF  CCP2CON,4     ;(LSB DO DUTY CICLE)
                    ;PERIODO DO PWM
                    ;Tp = CCP2L:CCP2CON<5,4>*Tosc*TMR2 PRESCALE
                    ;Tp = 0*250NS*16 = 0

    CALL DELAY_5S

;*****
;PWM COM 25% DE PWM (CCP2 - PINO 16) E 75% (CCP1 - PINO 17)
;*****
    MOV LW 0X40
    PWM

;*****
;PWM COM 50% DE PWM (CCP2 - PINO 16 E CCP2 - PINO 17)
;*****
    MOV LW 0X80
    PWM

;*****
;PWM COM 75% DE PWM (CCP2 - PINO 16) E 25% (CCP1 - PINO 17)
;*****
    MOV LW 0XC0
    PWM

;*****
;PWM COM 100% DE PWM (CCP2 - PINO 17) E 0% (CCP1 - PINO 17)
;*****
    MOV LW 0XFF
    PWM
    GOTO LOOP

END

```