

Fernando de Melo Luna Filho

Controle Inteligente para Automação Predial

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
2002**

Agradecimentos

Agradeço ao professor e orientador Adolfo Bauchspiess, pelo acompanhamento, revisão de estudo, orientação e dedicação prestados durante a execução deste trabalho.

Agradeço ao professor Marco Egito pela experiência prática que se mostrou indispensável para a execução do projeto.

Agradecimentos especiais aos meus familiares, em especial ao meu pai, Fernando de Melo Luna, e à minha mãe, Valéria Maria Bezerra Silva, que tornaram tudo que realizei e tudo que ainda pretendo realizar possível.

Agradeço à minha namorada Débora Amaral de Medeiros pela compreensão nos momentos de ausência e ajuda em todas as ocasiões nas quais dela precisei.

Agradeço a todos os componentes dos laboratórios de Engenharia Elétrica que agiram de maneira rápida no intuito de facilitar a execução de atividades do projeto. Agradecimentos também a todos os professores que contribuíram direta ou indiretamente para a realização das atividades citadas.

Índice Analítico

1.	INTRODUÇÃO.....	8
1.1	SISTEMAS INTELIGENTES.....	8
1.2	CONTROLE CLÁSSICO (PID) E CONTROLE INTELIGENTE.....	9
1.3	MODELAGEM DE PROCESSOS EM SISTEMAS DE CONTROLE.....	10
1.4	SISTEMAS EMBARCADOS – “EMBEDDED SYSTEMS”.....	10
1.5	JUSTIFICATIVAS.....	11
2.	REVISÃO BIBLIOGRÁFICA	14
2.1	FAMILIARIZAÇÃO COM A LÓGICA “FUZZY”.....	14
2.2	MICROCONTROLADORES E MICROPROCESSADORES.....	16
2.2.1	<i>Microcontroladores da Família PIC (Programmable Integrated Controller)</i>	16
2.2.2	<i>Modelo de microcontrolador utilizado: PIC16F877</i>	17
2.2.3	<i>Organização da memória</i>	18
2.2.4	<i>Principais Registradores de Funções Especiais (SFR)</i>	19
2.3	CONVERSOR D/A.....	20
2.4	CONTROLE PWM.....	21
2.5	CONVERSÃO CC-CC.....	22
2.6	SISTEMAS TÉRMICOS	23
3.	METODOLOGIA.....	25
3.1	PESQUISA DE MERCADO - CONTROLADORES INTELIGENTES.....	25
3.2	PROCEDIMENTOS PARA USO DO PIC 16F877	28
3.3	EXPERIMENTOS REALIZADOS.....	30
3.4	SISTEMA OBJETO DE ESTUDO – MAQUETE DE ESCRITÓRIO COM APARELHOS DE AR - CONDICIONADO.....	34
4.	RESULTADOS OBTIDOS	46
4.1	MODELAGEM DO SISTEMA FÍSICO	46
4.2	SIMULAÇÃO DE SISTEMAS EM MALHA ABERTA	47
4.3	PROJETO DOS CONTROLADORES.....	49
4.4	SIMULAÇÃO DE SISTEMAS EM MALHA FECHADA	55
4.5	RESULTADOS EXPERIMENTAIS	59
5.	CONCLUSÕES.....	67
6.	BIBLIOGRAFIA	70

Lista de Figuras

Figura 1 - Esquema Geral de um sistema <i>Fuzzy</i>	15
Figura 2 - Relação de pinos do PIC16F877, encapsulamento PDIP	18
Figura 3 - Exemplo de conversor D/A usando rede R-2R	20
Figura 4 - PID Autotune <i>Fuzzy</i> Logic Controller - Fuji Electric PXV3	27
Figura 5 - 1/32 DIN <i>Fuzzy</i> Logic Controllers HCS	27
Figura 6 - Controlador de Temperatura e Processo ES100 OmRON	27
Figura 7 - Circuito para gravação de PIC's	28
Figura 8 - Interface do programador de PIC's ProgPIC	29
Figura 9 - Esquema de operação dos microcontroladores PIC	30
Figura 10 - Esquema de circuito	31
Figura 11 - Esquema de Circuito com LCD	33
Figura 12 - Maquete usada como sistema térmico	35
Figura 13 - Esquema de circuito de um secador de cabelo típico	37
Figura 14 - Driver de Potência projetado	39
Figura 15 - Resultado de simulação do driver de potência - Tensão sobre a carga e seu valor médio	40
Figura 16- Esquemas dos processos térmicos	42
Figura 17 - Processos térmicos com controladores digitais	42
Figura 18 - Interfaces criadas para observação das temperaturas desejadas	44
Figura 19 - Diagrama de blocos com a lógica de apresentação de dados	45
Figura 20- Lugar geométrico das raízes do processo 1,2	47
Figura 21 - Lugar geométrico das raízes do processo 2,5	48
Figura 22 - Resposta ao degrau unitário do sistema 1,2 em malha aberta	48
Figura 23 - Resposta ao degrau unitário do sistema 2,5 em malha aberta	49
Figura 24 - LGR do processo 1,2 com controlador PI calculado	52
Figura 25 - LGR do processo 2,5 com controlador PI calculado	52
Figura 26 - Funções de pertinência da variável de "Erro"	54
Figura 27 - Funções de pertinência da variável de "Ciclo de Trabalho"	54
Figura 28 - Base de Regras implementada	55
Figura 29 - Sistema simulado com controlador PI com parâmetros calculados anteriormente	56
Figura 30- Resposta do processo 1,2 com controlador PI ao degrau unitário	56
Figura 31 - Resposta do processo 2,5 com controlador PI ao degrau unitário	57
Figura 32 - Sistema simulado com controlador difuso	58
Figura 33 - Resposta do processo 1,2 com controlador proporcional <i>fuzzy</i> ao degrau unitário	58
Figura 34 - Resposta do processo 2,5 com controlador proporcional <i>fuzzy</i> ao degrau unitário	59
Figura 35- Resposta do processo 1,2 com controlador proporcional integral à onda quadrada 40-45	60
Figura 36- Resposta do processo 1,2 com controlador proporcional integral à onda quadrada 45-50	60
Figura 37- Resposta do processo 1,2 com controlador proporcional integral à onda quadrada 50-55	61
Figura 38- Resposta do processo 2,5 com controlador proporcional integral à onda quadrada 29-31	61
Figura 39- Resposta do processo 2,5 com controlador proporcional integral à onda quadrada 31-33	62
Figura 40- Resposta do processo 2,5 com controlador proporcional integral à onda quadrada 33-35	62
Figura 41- Resposta do processo 1,2 com controlador <i>fuzzy</i> à onda quadrada 40-45	63
Figura 42- Resposta do processo 1,2 com controlador <i>fuzzy</i> à onda quadrada 45-50	63
Figura 43- Resposta do processo 1,2 com controlador <i>fuzzy</i> à onda quadrada 50-55	64
Figura 44- Resposta do processo 2,5 com controlador <i>fuzzy</i> à onda quadrada 29-31	64
Figura 45- Resposta do processo 2,5 com controlador <i>fuzzy</i> à onda quadrada 31-33	65
Figura 46- Resposta do processo 2,5 com controlador <i>fuzzy</i> à onda quadrada 33-35	65

Lista de Tabelas

Tabela 1 - Dados de consumo de energia em empresas do DF – Fonte: Metroquattro Engenharia	11
Tabela 2 – Controladores da marca Fuji	25
Tabela 3 - Controladores da marca HCS	26
Tabela 4 - Controladores da marca OmRON.....	26
Tabela 5 - Descrição dos pinos do LCD HD44780	33
Tabela 6 - Sensor de temperatura LM35.....	36
Tabela 7 - Dispositivos usados para implementar driver de potência	38
Tabela 8- Parâmetros dos processos estudados	47
Tabela 9 - Descrição do sistema de inferência <i>fuzzy</i> simulado	55

Resumo

Os sistemas inteligentes procuram fornecer respostas que solucionam problemas, de forma apropriada às situações específicas destes problemas, mesmo sendo novas ou inesperadas. Automação predial indica o conjunto de tecnologias associadas para uso racional de energia elétrica e automatização de algumas atividades dentro de casas e edifícios.

Sistemas embarcados constituem uma solução para diversos problemas onde a intervenção humana direta e contínua é desnecessária ou impossível. Estes sistemas são implementados geralmente usando-se unidades de processamento dedicado de tamanho e capacidade de computação reduzidos, como microcontroladores e microprocessadores.

O presente trabalho visa estudar, através de metodologia incluindo simulações e execução de testes em maquetes, processos onde a variável que se deseja controlar é a temperatura. As simulações e o projeto de controladores foram feitos através do software MatLabTM.

A maquete selecionada constitui a configuração de um escritório típico com dois secadores de cabelo representando dois aparelhos de ar-condicionado e sensores de temperatura para captura do valor da variável desejada. A captura de dados para análise posterior foi feita usando-se uma placa de aquisição de dados da empresa “National Instruments” associada ao software “LabVIEWTM” do mesmo fabricante.

A unidade de processamento adotada para controle da temperatura nas salas da maquete é um microcontrolador da família PIC, produzido pela empresa MICROCHIPTM. O modelo selecionado é o 16F877. Este modelo possui “8k” de memória de programa “Flash”, com instruções de 14 bits, memória de dados de 256 bytes e memória EEPROM. Sua velocidade máxima de operação é de “20MHz”.

Os controladores foram implementados no microcontrolador através da linguagem de programação C, através de programas que foram compilados no ambiente de desenvolvimento “HPDPIC”. A gravação dos programas na memória do microcontrolador foi feita através do software “ProgPIC” disponível na internet.

A técnica de controle inteligente “Lógica *Fuzzy*” apresentou desempenho superior ao controlador proporcional integral, justificando seu uso nas diversas situações analisadas. Os resultados fornecem base quantitativa para a análise do desempenho destas duas estratégias de controle em aplicações de controle de temperatura.

Abstract

Intelligent systems look for to supply answers that solve problems, of appropriate form to specific situations of these problems, even being new or unexpected. Building automation indicates the set of technologies associated for rational use of electric energy and automatization of some activities in houses and buildings.

Embedded systems constitute a solution for diverse problems where the direct and continuous human being intervention is unnecessary or impossible. These systems are implemented generally using dedicated processing units with little size and reduced capacity of computation, such as microcontrollers and microprocessors

This work aims to study, through methodology including simulation and execution of tests in mockups, processes where the variable to be controlled is temperature. Simulations and project of controllers was made through MatLabTM software.

The selected mockup constitutes the configuration of a typical office with two driers of hair representing two devices of air-conditioning and temperature sensors for capture of the value of the desired variable. The capture of data for posterior analysis was made using a board of data acquisition made by the company “National Instruments” associated with software LabVIEWTM of the same manufacturer.

The processing unit adopted to control the temperature in the rooms of the mockup is a PIC microcontroller, produced by MICROCHIPTM. The selected model is 16F877. This model has “8k” of Flash program memory, with 14 bits instructions, 256 bytes of data memory and also EEPROM memory. Its maximum operating speed is 20MHz.

Controllers were implemented in the microcontroller through the programming language “C”, through programs that had been compiled in the development environment HPDPIC. The writing of the programs in the memory of the microcontroller was made with the software ProgPIC.

The technique of intelligent control called “*Fuzzy Logic*” presented superior performance to the integral proportional controller, justifying its use in the diverse situations analyzed. The results supply a quantitative base for analysis of the performance of these two strategies in applications of temperature control.

1. Introdução

1.1 Sistemas Inteligentes

Lógica *Fuzzy*, Redes Neurais, Sistemas Especialistas e Algoritmos Genéticos fazem parte de um novo paradigma conhecido por *sistemas inteligentes*. Estes sistemas procuram fornecer respostas que solucionam problemas, de forma apropriada às situações específicas destes problemas, mesmo sendo novas ou inesperadas [1].

A operação destes sistemas é inspirada, em geral, por sistemas biológicos. A capacidade criativa dos seres humanos, de raciocinar de maneira imprecisa, incerta ou difusa contrasta com a forma de operar de computadores e máquinas, movidos por lógica binária e precisa. No momento em que estas máquinas perdessem esta restrição, tornar-se-iam eventualmente inteligentes, podendo raciocinar de forma imprecisa. Esta forma de raciocínio é conhecida em inglês por *Fuzzy*, tendo como tradução mais próxima a palavra “nebuloso”.

A Lógica *Fuzzy*, então, procura incorporar a forma humana de pensar em sistemas. Quando esta técnica é aplicada em malhas de controle, é chamada de Controle *Fuzzy*. Os controladores produzidos geralmente têm uso em aplicações constituídas por sistemas dinâmicos complexos. Uma aplicação de sistemas *Fuzzy* bem conhecida é o *Controle Fuzzy Supervisório*, ou Operacional, onde plantas industriais complexas são automatizadas em funções delegadas tradicionalmente a operadores humanos [1]. O controlador *Fuzzy* busca, então, capturar a experiência do operador humano fornecendo uma técnica para projeto de algoritmos de supervisão.

Uma variação do controle *Fuzzy* simples é obtida através da análise de variáveis do processo controlado e de seu ambiente. Com esta observação, um controlador *fuzzy* auto-ajustável, ou SOFLIC – “*Self-Organizing Fuzzy Logic Controller*”, é construído tendo como princípio de funcionamento a variação das regras de controle de acordo com mudanças no processo controlado ou em seu ambiente [6]. Esta técnica recente possui resultados publicados de ótima qualidade, como em [8], justificando seu uso.

Uma técnica de controle inteligente também bastante conhecida consiste nas Redes Neurais Artificiais. Estas, por sua vez, simulam funções biológicas do cérebro humano para execução de tarefas de controle. Esta técnica procura aprender como controlar um sistema através de exemplos numéricos, enquanto que lógica *Fuzzy* aprende através de

exemplos semânticos. A associação entre estas duas técnicas recebe o nome de sistema “*Neurofuzzy*” que combina as vantagens dos dois tipos de sistemas inteligentes.

1.2 Controle Clássico (PID) e Controle Inteligente

Atualmente, diversas técnicas de controle, embasadas na teoria de controle clássico estão disponíveis [4]. Estas técnicas usam as características do sistema linear a ser controlado como parâmetros para projeto dos controladores lineares. O projeto dos controladores faz uso de diversas ferramentas, como LGR – lugar geométrico das raízes, e diagramas de Bode. Estas ferramentas fornecem justificativas para o emprego de uma dentre diversas técnicas de controle linear, como: “Compensação de Atraso e/ou Avanço”, Controle PID[10].

Especificações de desempenho e robustez para o processo em malha fechada são alcançadas partindo-se do desempenho do processo em malha aberta, ou seja, sem realimentação.

Na maioria dos casos, porém, os processos possuem característica não-linear bastante acentuada e, para projeto de controladores lineares, o processo é linearizado em torno de um ponto de operação. Então, somente para este ponto de operação, os parâmetros de desempenho são obtidos com a inserção do controlador linear. Isto decorre do fato que o processo linearizado varia seus parâmetros para pontos de operação distintos. Entretanto, vários processos industriais possuem não-linearidades suaves, o que não prejudica tanto o desempenho dos controladores lineares.

Para processos de não-linearidade muito acentuada e com exigências de desempenho críticas, algoritmos de controle inteligente tornam-se uma alternativa adequada. A capacidade destes algoritmos de variar suas estratégias de controle de acordo com o processo controlado consiste em grande vantagem. Processos simples, entretanto, com comportamento aproximadamente linear e exigências de desempenho pouco restritas, não encontram justificativa em usar técnicas de controle inteligente, pois o custo computacional envolvido na inserção de regras de controle mais complexas pode inviabilizar seu emprego.

Este debate sobre viabilidade e adequação da utilização de técnicas de controle inteligente é intenso, tanto no meio acadêmico como em ambiente industrial. Discussões sobre o ganho de desempenho oferecido pelo emprego de controle *Fuzzy* podem ser observados como em [9]. Aspectos sobre as gerações de algoritmos *Fuzzy* e suas

restrições de desempenho são questionados e conclusões sobre a adequação do emprego desta técnica são feitas.

1.3 Modelagem de Processos em Sistemas de Controle

Modelagem simplifica esquematicamente situações reais para facilitar e baixar o custo da execução de ensaios e experimentos sobre os sistemas reais. Existem, convencionalmente, três enfoques usados para modelagem de plantas e processos: experimental, matemático (analítico) e heurístico.

O método experimental equivale a obter pontos discretos de uma curva característica de entrada e saída para um determinado processo. Este método encontra restrições no que diz respeito à possibilidade de execução de experimentos nos sistemas reais.

O método matemático, por sua vez, procura obter um modelo idealizado do processo a ser controlado, na forma de equações diferenciais que descrevem a dinâmica do mesmo. Para simplificação dos modelos construídos, assume-se que o processo é linear em um determinado ponto de operação. Esta propriedade de linearidade fornece técnicas conhecidas e muito eficazes para obtenção de soluções analíticas ideais. Outra propriedade assumida, a de que o sistema seja não-variante no tempo, também constitui uma restrição para este tipo de modelagem, ao assumir, por exemplo, que componentes não se desgastam. Em decorrência destas restrições, existem dificuldades no desenvolvimento de uma descrição matemática realista do processo.

O método heurístico consiste na realização de tarefas de acordo com experiência prévia, pelo uso de regras práticas e conhecidas. Estas regras, que associam conclusões com condições, assemelham-se às tabelas criadas pelo método experimental.

A Lógica *Fuzzy* surgiu da necessidade de um método que tratasse de maneira sistemática, quantidades imprecisas e mal-definidas, o que inclui a maioria dos processos físicos conhecidos.

1.4 Sistemas Embarcados – “Embedded Systems”

Sistemas com inteligência para sua operação sem necessidade da intervenção direta e contínua estão cada vez mais presentes no dia-a-dia do ser humano. Eletrodomésticos, aparelhos de comunicação móvel e automóveis com diversas funções microprocessadas

fazem parte do cenário tecnológico atual. Longe da alta tecnologia aplicada aos robôs móveis com operação remota e aos satélites, os sistemas citados anteriormente possuem a mesma funcionalidade associada: desempenho de diversas funções sem a necessidade da supervisão e intervenção direta do ser humano.

Sistemas eletrônicos embarcados combinam tarefas de software com hardware projetado e dedicado, ambos analógicos e digitais. Alguns destes sistemas podem ser implementados com um microcontrolador disponível comercialmente, um processador de sinais digitais ou um microprocessador digital associado a um software.

Alguns sistemas operacionais, reconhecendo o potencial dos sistemas embarcados, são projetados para executar em microprocessadores ou microcontroladores. Atualmente, existe o “Embedded Linux” usado em “palm-tops” cada vez mais difundidos.

1.5 Justificativas

O uso racional da energia elétrica e recursos naturais é uma tendência crescente em todas as tecnologias desenvolvidas atualmente. A preocupação com estes fatores começa a direcionar alguns projetos no intuito de produzir equipamentos que preservam o meio ambiente.

Equipamentos de ar-condicionado possuem alto consumo de energia elétrica, quando comparados a itens como iluminação e computadores. Levantamentos comprovam a grande contribuição destes aparelhos no consumo de energia elétrica dentro das empresas. Estes dados estão na tabela 1.

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

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

Pela tabela 1, pode-se dizer que quanto menor a empresa, maior a contribuição percentual dos aparelhos de ar-condicionado. O consumo destes, portanto, influencia diretamente os custos de produção e prestação de serviços destas empresas.

Em conjunto com o fator de grande consumo de energia destes equipamentos, está o crescente valor cobrado pelo uso de energia elétrica. Enquanto que em 1995, o preço do “megawatt.hora” cobrado para consumo residencial era de R\$ 72, no ano de 2001 o preço cobrado é de R\$ 172 pela mesma quantidade de energia consumida. Atualmente, este preço está em R\$ 260 por “megawatt.hora” consumido. Isto representa um crescimento maior do que 100%, afetando diretamente os custos de vida da população e das empresas em geral.

Um equipamento que pudesse, portanto, sem retirar ou substituir os aparelhos de ar-condicionado já instalados por aparelhos mais modernos, reduzir o consumo de energia envolvido seria muito útil para diversos setores da sociedade.

Soluções baseadas em “CLP’s” – controladores lógicos programáveis, estão disponíveis no mercado, porém com preços desencorajadores. Além disto, a programação destes equipamentos especializados se faz necessária. Estes dois fatores influenciam diretamente na baixa presença de mercado destes equipamentos.

A solução proposta no presente trabalho é um sistema de baixo custo, de fácil programação e simples instalação, para redução do consumo de energia dos aparelhos de ar-condicionado. Este baixo custo fornece um pequeno intervalo de tempo para retornar o valor investido para aquisição do sistema. A programação fácil é feita usando-se as vantagens conhecidas dos sistemas inteligentes baseados em regras lingüísticas que um operador sem treinamento especializado pode executar utilizando apenas a experiência adquirida no local onde o aparelho de ar-condicionado é usado. A instalação é, então, simplificada pela adoção de soluções “plug-and-play”, ou seja, que permitam a implementação deste sistema em ambientes com aparelhos de ar-condicionado já instalados.

Para implementação deste sistema, seria necessário seguir algumas etapas pré-definidas: avaliação de microcontroladores para implementar a unidade de processamento, projeto de módulo de potência para acionamento dos aparelhos de ar-condicionado, avaliação de sensores de temperatura disponíveis e execução de testes do sistema produzido em uma maquete representando um escritório típico.

2. Revisão Bibliográfica

2.1 Familiarização com a lógica “*Fuzzy*”

Lógica *Fuzzy* é uma técnica de Inteligência artificial, disciplina que procura maneiras de máquinas simularem o raciocínio humano na solução de problemas diversos [1]. Esta abordagem procura mimetizar a forma humana de atuar. Nestes sistemas o controlador pode se comportar de maneira dedutiva, processo onde conclusões são obtidas a partir de informações pré-existentes, ou indutiva, também chamado de controle *fuzzy* adaptativo.

Um sistema baseado em “Lógica *Fuzzy*” pode ter sua ação esquematizada pela figura 1. Nesta, observam-se 4 elementos:

- *Fuzzificador*,
- Regras, ou base de conhecimento,
- Inferência, ou lógica de tomada de decisões,
- Defuzzificador.

O “fuzzificador” é responsável pelo mapeamento das entradas numéricas em conjuntos *fuzzy* (variáveis lingüísticas). Para tanto, cria-se uma interface de fuzzificação para recepção e tratamento destes dados numéricos, provenientes de sensores ou dispositivos de entrada computadorizados.

A inferência é realizada mapeando-se valores lingüísticos de entrada em valores lingüísticos de saída com o uso das regras. Esta usa implicações *fuzzy* para simulação de decisões humanas, gerando ações de controle, chamados de conseqüentes, partindo-se de um conjunto de condições de entrada, chamada de antecedentes.

Esta base de conhecimento representa o modelo do sistema a ser controlado, consistindo numa base de dados e uma base de *regras fuzzy lingüísticas*. A base de dados fornece definições numéricas e a base de regras caracteriza os objetivos do controlador e sua estratégia usada, geralmente fornecida por pessoas especialistas no sistema.

O “defuzzificador” mapeia valores lingüísticos em valores numéricos de saída. Esta função é realizada por uma interface de defuzzificação, obtendo-se um valor discreto que possa ser usado numa ação de controle no mundo real. Esta função é necessária

apenas quando a saída do controlador tiver de ser interpretada como uma ação de controle discreta. Entretanto, existem sistemas que não precisam desta etapa, quando a saída *fuzzy* é interpretada de modo qualitativo. Alguns dos métodos usados para defuzzificação são os seguintes:

- Centro-da-Área (C-o-A);
- Centro-do-Máximo (C-o-M);
- Média-do-Máximo (M-o-M).

O tipo de defuzzificação a ser escolhido depende basicamente da aplicação do sistema de inferência *fuzzy* resultante. Para suporte na tomada de decisões, a técnica C-o-M é mais indicada para decisões quantitativas e M-o-M para decisões qualitativas. Em aplicações de reconhecimento de padrões, usa-se a M-o-M com maior frequência. Aplicações em malha fechada requerem que a propriedade da continuidade seja obedecida, dando-se preferência à técnica C-o-M. Para aplicações com restrições de recursos, como memória e velocidade de processamento, a técnica C-o-M mostra-se inviável por envolver integração numérica. A propriedade de continuidade pode então ser obedecida usando-se a técnica M-o-M.

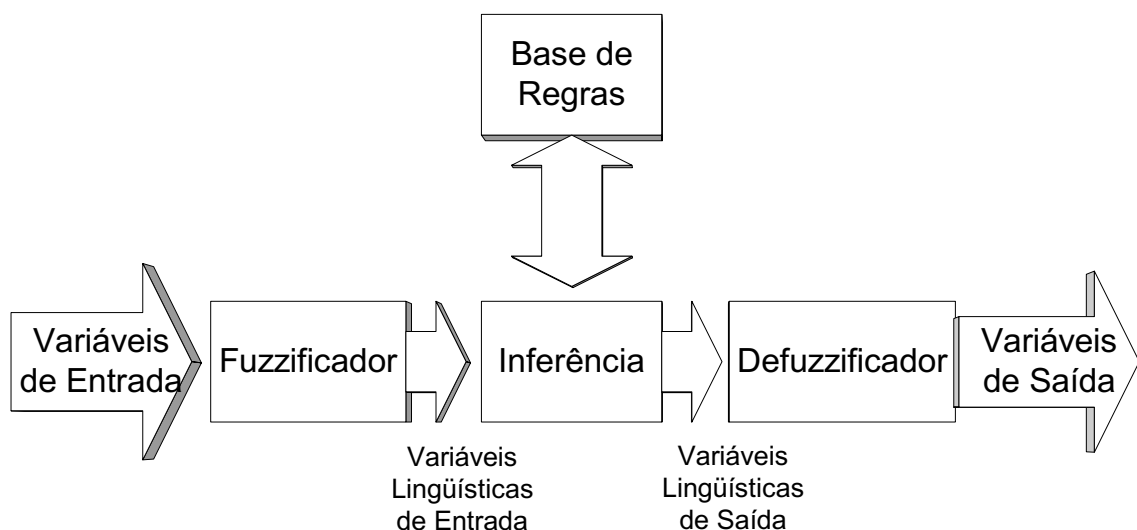


Figura 1 - Esquema Geral de um sistema *Fuzzy*

A base dos sistemas “*Fuzzy*” consiste da teoria dos conjuntos “*Fuzzy*”. Estes conjuntos são uma extensão dos conjuntos convencionais, que permitem somente que elementos sejam verdadeiros ou falsos (lógica booleana, bivalente). Conjuntos “*Fuzzy*”

permitem que seus elementos possuam um certo “grau de pertinência” associado, sendo esta propriedade conhecida como “multivalência”. Isto permite a aproximação com o mundo real que não é bivalente, é na realidade multivalente com um vasto número de opções ao invés de somente duas. A lógica *fuzzy*, então, fica habilitada a trabalhar com tais incertezas de fenômenos naturais de forma rigorosa e sistemática.

A determinação do grau de pertinência, para conjuntos “*Fuzzy*” contínuos, se dá pela análise de “funções de pertinência”. Estas funções possibilitam o cálculo do grau de pertinência de acordo com o valor assumido pela variável. Elas representam os aspectos fundamentais de todas as ações teóricas e práticas de sistemas *fuzzy*. Algumas das funções mais usadas são as seguintes: normal, triangular e trapezoidal.

A lógica “*fuzzy*” pode ser vista com um tratamento das incertezas através de funções de pertinência [1]. Isto possibilita que expressões verbais “imprecisas” sejam manuseadas por um sistema *fuzzy*, podendo então traduzir a forma humana de atuar em computadores.

2.2 Microcontroladores e Microprocessadores

Um microcontrolador pode ser definido como um dispositivo eletrônico pequeno, possuidor de diversas características de computadores pessoais (Ex: memória, processador), utilizado para controle de processos [11]. Como possui memória, este componente pode ser programado para realizar um certo número de tarefas simplesmente gravando o programa no microcontrolador. Portanto, este componente executará o programa gravado a cada vez que for alimentado.

A principal vantagem deste dispositivo é conter, em apenas um circuito integrado, diversos módulos úteis para o controle de processos, como: conversor AD, saídas PWM, temporizadores, contadores, portas de E/S, comunicação serial e paralela, etc. Então, apesar de ter uma ULA bem menos poderosa que microprocessadores, os microcontroladores possuem diversos recursos em uma única pastilha de silício [11].

2.2.1 Microcontroladores da Família PIC (Programmable Integrated Controller)

Estes dispositivos apresentam uma arquitetura alternativa à arquitetura Von-Neumann, chamada de Harvard [11]. Nesta, existem dois barramentos internos, um para dados e o outro para instruções, diferentemente da Von-Neumann que usa apenas um

barramento usado tanto para instruções quanto para dados. O barramento de dados é de 8 bits, enquanto que o de instruções pode ser de 12, 14 ou 16 bits.

Todos os microcontroladores PIC seguem a filosofia RISC, com conjunto de instruções reduzido (em torno de 35 instruções distintas, variando entre os modelos de PIC). A principal vantagem desta filosofia é a facilidade de aprendizado. Todavia, este conjunto de instruções reduzido força o programador construir diversas funções para implementar uma tarefa executada às vezes por apenas uma instrução CISC.

O sinal de “clock” nestes componentes é internamente dividido por quatro. Esta divisão se dá em 4 fases a cada ciclo de máquina. Na primeira fase, incrementa-se o contador de programa e busca-se a instrução seguinte da memória de programa; na última fase a instrução é armazenada no registrador de instruções. No próximo ciclo de máquina, entre a primeira e a última fase a instrução é decodificada e executada.

2.2.2 Modelo de microcontrolador utilizado: PIC16F877

Características principais:

- CPU RISC de alto desempenho;
- 35 instruções de 8 bits, que usam apenas um ciclo para execução (menos as que executam desvios);
- velocidade de operação de até 20MHz;
- memória dividida em 3 partes:
 - memória de programa FLASH: 8 K de 14 bits(instrução);
 - memória de dados RAM volátil: 368 bytes;
 - memória de dados não-volátil (EEPROM): 256 bytes.
- Possibilidade de uso de até 14 fontes de interrupção;
- Oscilador selecionado dentre 4 opções: ressoador, cristal, RC ou circuitos de oscilação (híbridos);
- Proteção de código programável.

Características periféricas básicas:

- 3 contadores com prescaler, um de 16 bits (TMR1) e dois de 8 bits (TMR0 e TMR2);
- 2 módulos que servem para captura, comparação e geração de sinal PWM (resolução de 10 bits);

- conversor analógico-digital embutido, possibilitando o uso de até 8 canais de entrada com resolução de 10 bits;
- uso de até 33 pinos para operações de entrada e saída.

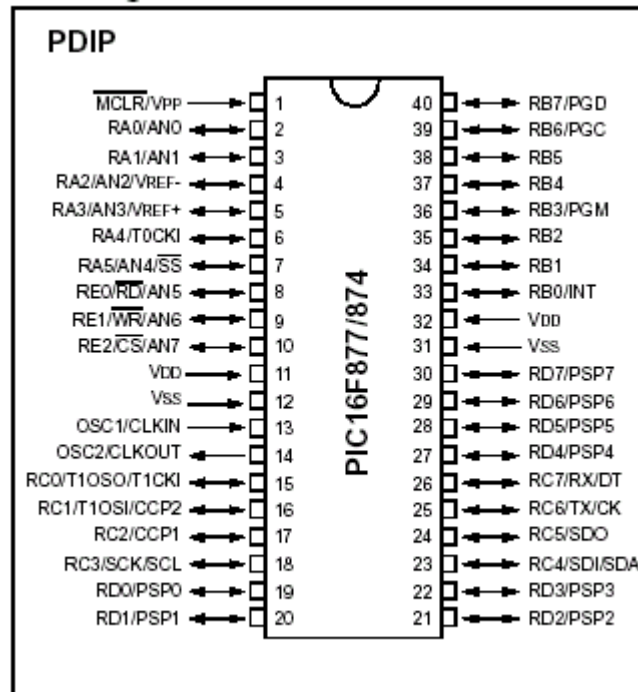


Figura 2 - Relação de pinos do PIC16F877, encapsulamento PDIP

2.2.3 Organização da memória

Como foi dito anteriormente, existem três blocos de memória nos PIC's: programa, dados e EEPROM. Os dispositivos PIC16F87X possuem um contador de programa de 13 bits capaz de endereçar um espaço de memória de 8Kx14 bits. Dois endereços desta memória de programa são reservados para duas tarefas importantes: o endereço "0x0000", para "reset" e o endereço "0x0004" para interrupções.

A memória de dados é particionada em 4 "bancos" de 128 bytes que contém dois tipos de registradores classificados de acordo com sua utilização: registradores de uso geral e registradores para funções especiais. O acesso a estes "bancos" é feita através da escrita nos bits 5 e 6 do registrador "STATUS".

A memória de dados EEPROM pode ser lida e escrita durante a operação do PIC. Estas operações são feitas usando-se apenas um byte desta memória. Uma escrita ocasiona uma operação para apagar e então escrever o novo valor, não causando

nenhum impacto sobre o dispositivo, diferentemente de uma escrita na memória de programa. A memória de dados EEPROM mostra-se bastante útil, portanto, para armazenar resultados mesmo quando o dispositivo está sem alimentação (desligado).

2.2.4 Principais Registradores de Funções Especiais (SFR)

Estes registradores são usados pela CPU e pelos módulos periféricos para controlar o dispositivo para que o mesmo opere da forma desejada. Implementados como memória RAM estática, podem ser classificados em dois tipos: “core” (CPU) e periféricos. A seguir são citados os principais registradores SFR:

- STATUS: este SFR contém o estado aritmético da ULA (unidade de lógica e aritmética), se o dispositivo foi “resetado” e os bits para seleção dos bancos da memória de dados.
- OPTION: serve para configuração de várias opções para operação do microcontrolador. Contém bits para controle do “prescaler” do contador TMR0, controle do postscaler do WDT (watch dog timer).
- INTCON: usado para identificação e configuração de diversas interrupções.
- PIE1, PIE2, PIR1, PIR2: contêm bits individuais para habilitação de interrupções periféricas (eventos externos).
- PCON: registrador para possibilitar diferenciação entre diversos tipos de “RESET” que podem ser acionados.
- PCL e PCLATH: o PCL armazena os 8 bits menos significativos do contador de programa (PC). O PCLATH armazena os 5 bits mais significativos do contador de programa.
- ADCON0 e ADCON1: registradores para configuração do módulo AD, além da seleção de 1 entre de 8 canais a ser recebido. O resultado da conversão de 10 bits é armazenado no par de registradores ADRESH (AD result high) e ADRESL (AD result low).
- CCP1CON e CCP2CON: usados para configurar os módulos PWM. Em conjunto com o registrador T2CON, definem o período do sinal PWM e o período do ciclo de trabalho, que deve ser armazenado no registrador CCPR1L (8 bits mais significativos) e nos bits 4 e 5 do registrador CCP1CON (2 bits menos significativos), totalizando os 10 bits de resolução do módulo PWM.

Os registradores a seguir também são SFR, mas são utilizados para uso das portas de E/S do PIC:

- TRISA, TRISB, TRISC, TRISD e TRISE: usados para configurar os pinos das portas: PORTA, PORTB, PORTC, PORTD e PORTE; respectivamente como entrada (1) ou saída (0).
- Os registradores PORT, por sua vez, contêm os dados a serem enviados pelo PIC quando configurados como saída ou os dados a serem recebidos quando configurados como entrada.

Especificações a respeito de detalhes sobre operação dos módulos internos do PIC16F877 e demais registradores SFR encontram-se no data-sheet do dispositivo [13].

2.3 Conversor D/A

Conversão Digital Analógica consiste basicamente em transformar um valor em determinado código binário (BCD, binário puro) em um valor de tensão ou de corrente proporcional ao valor digital. As entradas digitais provenientes do microcontrolador são, portanto, convertidas em um valor de tensão analógica, na faixa de 0 a 5V.

O circuito usado para conversão D/A é conhecido como “rede R-2R”. Este arranjo conversor-resistor usa resistores de apenas dois valores: R e 2R. A figura 3 mostra um esquemático deste arranjo.

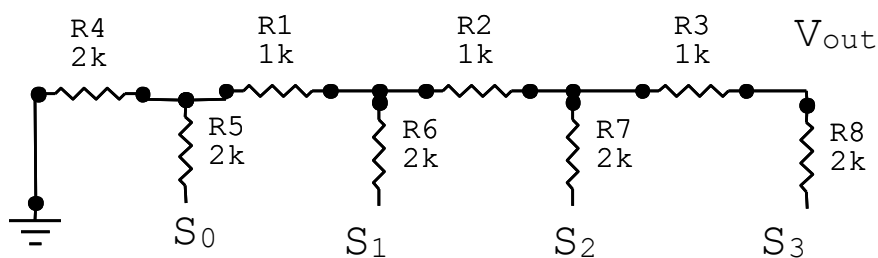


Figura 3 - Exemplo de conversor D/A usando rede R-2R

Mesmo precisando de o dobro de resistores do arranjo conhecido por “resistor ponderado”, a grande vantagem da rede R-2R consiste em apenas usar dois valores de resistores, enquanto que o arranjo de resistor ponderado deve trabalhar com tantos

valores de resistor quanto for o número de entradas digitais a serem convertidas. Este problema pode ser verificado tomando-se como exemplo um conversor de D/A de 12 bits, o arranjo resistor ponderado teria que trabalhar com um resistor de $1k\Omega$ para o bit mais significativo e de $2M\Omega$ pra o bit menos significativo, enquanto que na rede “R-2R” os resistores usados seriam apenas de $1k\Omega$ e $2k\Omega$. Os processos de fabricação atuais não permitem a fácil obtenção de resistores situados em ampla faixa de valores e mantenham a precisão que o processo necessita.

Por estas razões expostas, adotou-se o uso da rede “R-2R” como circuito para conversão D/A, por usar resistências em uma faixa de valores menos ampla.

- Especificações do Conversor D/A
 - Resolução: Como o módulo de conversão A/D do microcontrolador PIC16F877 possui 10 bits de resolução, a mesma resolução é adotada para a conversão D/A, possibilitando $2^{10} = 1024$ valores de tensão possíveis.
 - Linearidade: De modo geral, o erro de linearidade deve ser menor do que “ $\pm 1/2$ LSB” (Least Significant Bit) Este valor visa garantir a monotonicidade do conversor, ou seja, um aumento na entrada digital nunca produz uma diminuição da saída analógica.
 - Precisão: Esta especificação, em conversores D/A comerciais, deve possuir valor de “0.2% do valor de fundo de escala $\pm 1/2$ LSB”.
 - Tempo de acomodação: é o intervalo de tempo decorrido necessário para que a saída analógica se aproxima o suficiente do valor final. Este efeito reflete o transiente de elementos de circuito passivos parasitas (capacitâncias e indutâncias) e a característica de chaveamento de elementos ativos (transistores, chaves, etc...).

2.4 Controle PWM

Seja uma onda retangular de frequência conhecida “ f ” 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:

$$V_{dc} = \frac{1}{T} \left(\int_0^T v(t) dt \right) = \frac{1}{T} \left(\int_0^{T_{on}} v(t) dt + \int_{T_{on}}^T v(t) dt \right) = \frac{V}{T} \left(\int_0^{T_{on}} dt + 0 \right) = V \frac{T_{on}}{T}$$

Esta equação mostra que o valor médio da onda é diretamente proporcional à quantidade de tempo onde a onda permanece em nível alto em relação ao período da onda. O quociente “ T_{ON}/T ” é denominado “ciclo de trabalho” ou “*duty cycle*” da onda retangular e pode ser variado mudando-se o valor de “ T ”, processo conhecido por modulação por frequência de pulso “PFM”, ou variando-se valor de “ T_{ON} ”, técnica conhecida por modulação por largura de pulso “PWM”. A modulação PWM fornece uma relação linear entre a onda quadrada e seu valor médio, tendo aplicação mais adequada em sistemas lineares.

A principal vantagem envolvida no uso de sinais modulados por largura de pulso para acionamento de cargas de alta potência está relacionada à eficiência energética desta técnica. O transistor é usado nos modos de operação de corte e de saturação. Em corte o transistor praticamente não dissipa potência, enquanto que, em saturação, a queda de tensão envolvida na junção “coletor-emissor” é bem pequena, aproximadamente “0,2V”, dissipando pouca potência.

2.5 Conversão CC-CC

Conversores CC-CC são usados quando se tem uma fonte de tensão constante e deseja-se obter, a partir desta, uma fonte de tensão variável. Esta nova fonte de tensão deve ter seu valor regulável de acordo com chaves implementadas por dispositivos semicondutores.

As chaves analógicas podem ser implementadas através dos seguintes semicondutores: transistores BJT (Bipolar Junction Transistor), transistores MOSFET (Metal Oxide Semiconductor Field Effect Transistor) e tiristores tipo GTO (Gate Turnoff Thyristor). A operação destes dispositivos é feita através do controle da corrente de base, para BJT's, ou da tensão de porta, para os MOSFET's. Um certo valor de corrente de base liga o BJT, ou seja, o dispositivo entra em modo de condução. Para os MOSFET's, um certo valor de tensão na sua porta liga o componente, com mínimo consumo de corrente. Para os tiristores tipo GTO, a comutação entre os estados “ligado” e “desligado” deve ser forçada, diferentemente da comutação natural realizada em circuitos de conversão CA-CA.

O chaveamento dos dispositivos se dá através da excitação do terminal ativador do dispositivo por uma onda PWM. Através da regulação do ciclo de trabalho desta onda, varia-se a tensão que será enviada à carga partindo-se da fonte de tensão constante.

Os conversores CC-CC são comumente chamados de “Choppers”. São de dois tipos básicos:

- “Buck” ou abaixador;
- “Boost” ou elevador.

O “chopper Buck” fornece uma fonte de tensão cujo valor é menor ou igual à fonte de tensão constante, enquanto que o “chopper Boost” fornece uma fonte de tensão cujo valor é maior ou igual à fonte de tensão constante. Existe um outro tipo de “chopper” conhecido por “Buck-Boost” que pode fornecer tensão de saída menor ou maior do que a fonte de tensão constante.

Estes conversores encontram diversas aplicações em ambiente industrial. Motores de corrente contínua, chaveamento de alimentadores de potência, UPS (uninterruptible power supplies) e equipamentos operados por bateria.

2.6 Sistemas Térmicos

Diversos problemas envolvem controle de temperatura em fluidos. Controle de temperatura pode ser aplicado a sistemas de ar-condicionado e processos de fabricação com aquecimento ou resfriamento de determinada substância. Estes sistemas podem ser descritos através de parâmetros como “resistências térmicas” e “capacitâncias térmicas”. Estes parâmetros não são concentrados, como em circuitos elétricos, mas devem ser considerados distribuídos para análises mais precisas. Para processos onde se deseja apenas uma estimativa do seu comportamento dinâmico, o modelo usando parâmetros concentrados pode ser utilizado.

Estes sistemas térmicos realizam a transferência de calor entre substâncias por meio de três diferentes formas: condução, convecção e radiação. A maioria dos sistemas de controle para processos térmicos envolve condução ou convecção, já que a troca de calor por radiação é considerável apenas quando o corpo emissor está a temperatura muito maior do que o corpo receptor.

Para sistemas térmicos, a relação entre fluxo de calor e variação de temperatura pode ser descrita por uma relação diretamente proporcional. A equação (1) demonstra este comportamento, para transferência de calor por convecção:

$$q = K\Delta T = (H.A)\Delta T \quad (1)$$

Onde:

- ΔT : diferença de temperatura;
- H : coeficiente de convecção;
- A : área normal à convecção.

As grandezas “resistência térmica” e “capacitância térmica” podem ser descritas através das seguintes equações [4]:

$$q = mc\Delta T \quad (2)$$

$$R = \frac{d(\Delta T)}{d(q)} = \frac{1}{K} \quad (3)$$

$$C = \frac{q}{\Delta T} = mc \quad (4)$$

A “capacitância” ou “capacidade” térmica está diretamente relacionada ao calor específico “c” da substância, enquanto que a “resistência térmica” indica qual a oposição que o meio oferece para que o fluxo de calor se propague.

Para controle de temperatura em ambientes, o acionamento pode ser para aquecimento ou resfriamento. Para sistemas que envolvem aumento de temperatura, aquecedores e ventiladores associados são usados para fornecer o fluxo de calor necessário para variar a temperatura da maneira desejada. Para relacionar a variação de temperatura e o fluxo de calor em ambientes, a analogia com circuitos elétricos pode ser novamente usada. Considerando a resistência térmica em série com a capacitância térmica e desejando-se saber qual a temperatura no ambiente, a seguinte relação pode ser demonstrada [4]:

$$\frac{T(s)}{T_q(s)} = \frac{1}{RCs + 1} \quad (5)$$

Onde “T(s)” e “Tq(s)” são transformadas de laplace das variáveis: temperatura do ambiente e temperatura no aquecedor, respectivamente. Esta equação, que corresponde a um sistema de primeira ordem, possui comportamento bem definido e auxilia a análise e determinação experimental dos parâmetros deste modelo de representação do sistema térmico.

3. Metodologia

3.1 Pesquisa de mercado - Controladores inteligentes

Tendo como principal fonte de informações a internet, através das páginas [14], [15] e [16] fez-se um estudo comparativo de modelos de controladores “fuzzy” encontrados no mercado. As tabelas 2, 3 e 4 mostram diversas características de controladores, divulgadas pelos respectivos fabricantes. As figuras 4, 5 e 6 mostram alguns destes controladores.

Tabela 2 – Controladores da marca Fuji

Modelos	PX Series	Fuji Electric PXV3	Fuji Electric PYX Series
N° de entradas	1	1	1
N° de saídas	2	2	1
Tipo de entrada/saída	Thermocouple/RTD or Current/Voltage inputs.	Thermocouple/RTD or Current/Voltage inputs.	Thermocouple/RTD or Current/Voltage inputs.
Padrões disponíveis Norma DIN	1/4 DIN, 1/8 DIN, 72 mm, 1/16 DIN and 1/32	1/32 DIN	
Processo controlado	Temperatura	Temperatura	Temperatura
Tipo de controle usado	PID Autotune *	PID Autotune	PID Autotune
Interface com usuário	LED com 4 dígitos	LED com 4 dígitos	2 LEDs com 4 dígitos
Método de operação	3 botões	3 botões	3 botões
Input sampling cycle (s)	0.5	0.5	0.5
Control cycle (s)	0.5	0.5	
Accuracy	0.5%	0.5%	0.5%
Condições de temperatura e umidade	-10 a 50°C 90% (máx)	-10 a 50°C 90% (máx)	-10 a 50°C 90% (máx)
Peso (g)	100	100	Entre 175 e 400
Dimensões físicas	Painel: depende do	Painel: 48mmx24.5mm	Painel: depende do

(altura, espessura e comprimento)	modelo (56x100,76x76,100x100) Comprimento: 95.8mm	Comprimento: 98mm	modelo (96x48,48x48,96x96) Comprimento: 100mm
-----------------------------------	---	-------------------	---

Tabela 3 - Controladores da marca HCS

Modelos	1/16 DIN <i>Fuzzy Logic Controller</i>	1/32 DIN <i>Fuzzy Logic Controllers</i>	1/32 DIN Indicator & Controller <i>FuzyPro 2300</i>
Nº de entradas	1	1	1
Nº de saídas	2	2	1
Tipo de entrada/saída	Thermocouple/RTD or Current/Voltage inputs.	Thermocouple/RTD or Current/Voltage inputs.	PT100 ou T/C
Padrões disponíveis Norma DIN	1/16	1/32 DIN	1/32
Processo controlado	Temperatura	Temperatura	Temperatura
Tipo de controle usado	PID Autotune *	PID Autotune	PID Autotune
Interface com usuário	2 LEDs com 4 dígitos	2 LEDs com 4 dígitos	LED com 4 dígitos
Método de operação	4 botões	4 botões	3 botões
Input sampling cycle (s)	0.333	0.25	0.25
Precisão	0.5%	0.5%	0.1%
Condições de temperatura e umidade	-10 a 50°C 90% (máx)	-10 a 50°C 90% (máx)	0 to 55°C 20% to 95%
Dimensões físicas (altura, espessura e comprimento)	Painel: 48mmx48mm	-	-

Tabela 4 - Controladores da marca OmRON

Modelos	Controlador E5CK	Controlador ES100
Nº de entradas	1	1
Nº de saídas	2	2
Tipo de entrada/saída	Thermocouple/RTD or Current/Voltage inputs.	Thermocouple/RTD or Current/Voltage inputs.
Processo controlado	Temperatura e	Temperatura

	Processo	
Tipo de controle usado	PID Autotune *	PID Autotune
Interface com usuário	2 LEDs com 4 dígitos	2 LEDs com 4 dígitos
Método de operação	3 botões	4 botões
Precisão	0.3%	0.1%
Condições de temperatura e umidade	-10 a 50°C 90% (máx)	-10 a 50°C 90% (máx)
Peso (g)	170	800
Dimensões físicas (altura, espessura e comprimento)	Painel: 48mmx48mm Comprimento: 100mm	Painel: 96mmx96mm Comprimento: 160mm



Figura 4 - PID Autotune *Fuzzy Logic Controller* - Fuji Electric PXV3



Figura 5 - 1/32 DIN *Fuzzy Logic Controllers* HCS



Figura 6 - Controlador de Temperatura e Processo ES100 OmRON

3.2 Procedimentos para uso do PIC 16F877

O microcontrolador PIC pode ser programado através da escrita em sua memória de programa. Esta gravação do dispositivo é feita com o uso do programador “PROPIC” disponível gratuitamente na internet [17]. O circuito implementado em “protoboard” para gravação dos microcontroladores pode ser esquematizado pela figura 7. Este circuito usa a porta serial do computador para comunicação com o microcontrolador e pode gravar microcontroladores dos modelos “PIC16F87X” e “PIC16F8X”.

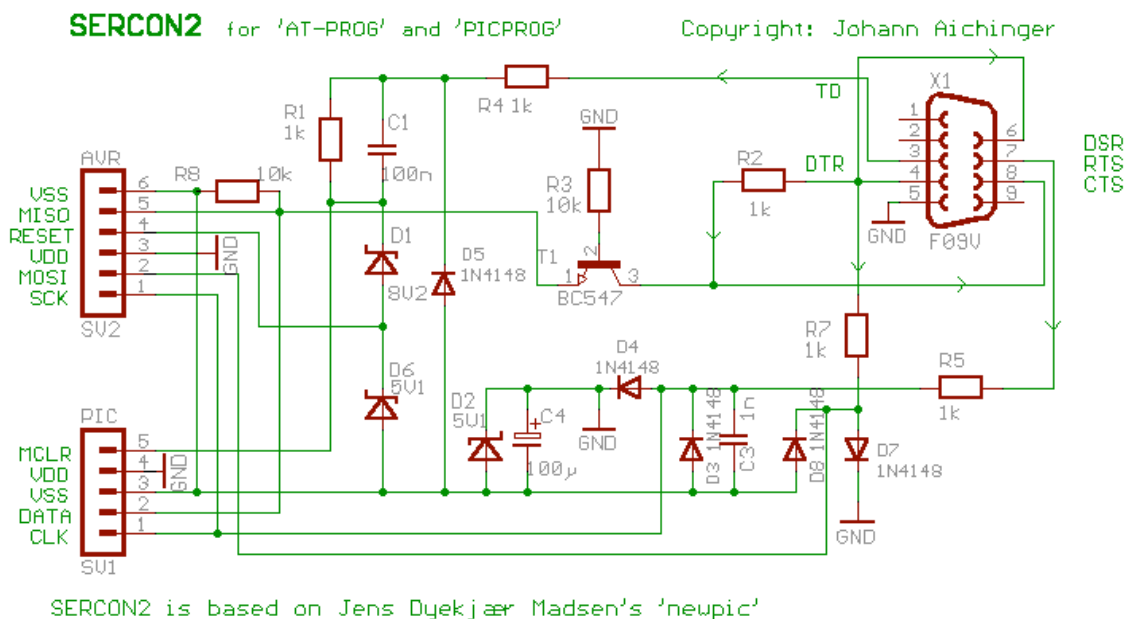


Figura 7 - Circuito para gravação de PIC's

O dispositivo pode ser programado usando-se linguagem de programação *assembly*, através do software “MPASM” disponibilizado pela empresa fabricante do componente. O PIC, sendo um processador que segue a filosofia RISC, possui apenas 35 instruções básicas. Qualquer outra operação a ser efetuada pelo PIC que não pertença a este conjunto de instruções deve ser programada, inclusive operações com números do tipo ponto flutuante. Esta propriedade dificulta sua programação em *assembly*, pois operações em variáveis do tipo ponto flutuante serão utilizadas nos algoritmos de controle projetados.

Uma alternativa à programação em baixo nível é a programação em linguagem C. Isto é possível através do *software* “HPDPIC™”. Usando-se este *software*, o código fonte desenvolvido pode ser convertido no código hexadecimal necessário à gravação

do componente. Com a inclusão de uma biblioteca denominada “pic.h”, todos os registradores do dispositivo estão disponíveis através de variáveis mnemônicas, facilitando a programação do microcontrolador. Este ambiente de desenvolvimento de programas une as facilidades de programação em alto nível e a possibilidade de acesso a funções de baixo nível, características conhecidas da linguagem C. Com este ambiente de programação, também se torna possível construir bibliotecas para uso dos módulos embutidos do PIC, o que facilita a sua programação e modularizando o programa.

Após a correta compilação do software escrito em linguagem C, o arquivo que será usado para gravar o PIC é o arquivo com extensão “hex” produzido após a compilação. Este arquivo contém a sequência de código binário em linguagem de máquina do microcontrolador que será executada.

O software para gravação do PIC possui interface amigável, tornando a tarefa de programação dos microcontroladores mais fácil. Este software também possibilita leitura do programa presente em um microcontrolador, bem como a seleção de opções de configuração para gravação e configuração da porta de comunicação que será usada no computador. A tela principal deste software está mostrada na figura 8.

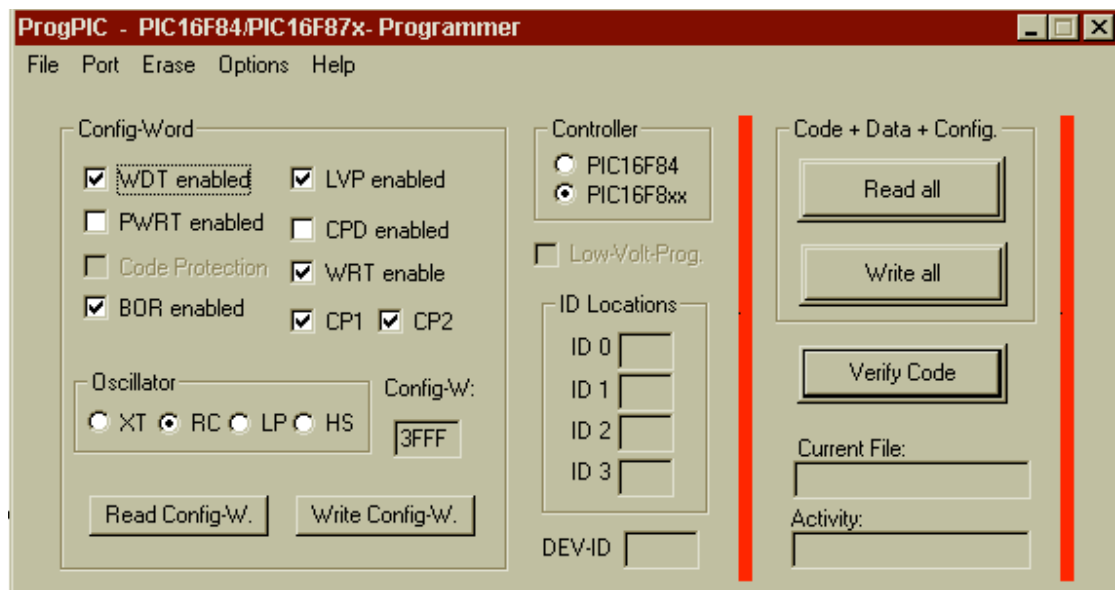


Figura 8 - Interface do programador de PIC's ProgPIC

A figura 9 procura descrever os passos para programação do microcontrolador PIC usando as ferramentas descritas:

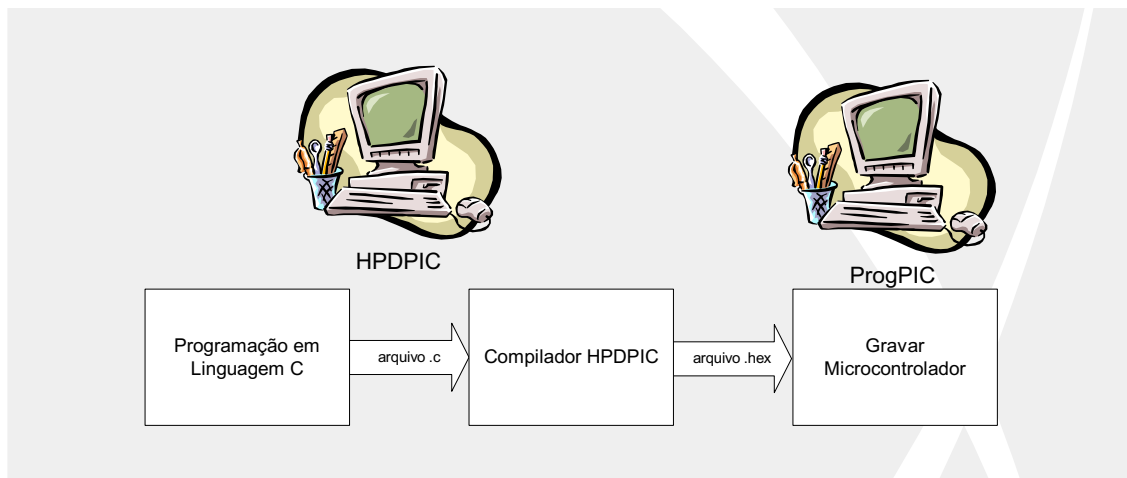


Figura 9 - Esquema de operação dos microcontroladores PIC

3.3 Experimentos realizados

Após a escolha do microcontrolador modelo PIC16F877, alguns experimentos foram realizados visando uma familiarização com os equipamentos necessários para operação do dispositivo e de seus módulos embutidos (conversão A/D, PWM).

3.3.1 Teste do módulo A/D do PIC16F877 e conversor D/A implementado:

O circuito implementado em “protoboard” para teste do módulo A/D do PIC16F877 e implementação de um conversor D/A de 10 bits utilizando uma rede R-2R pode ser visualizado pelo esquemático da figura 10:

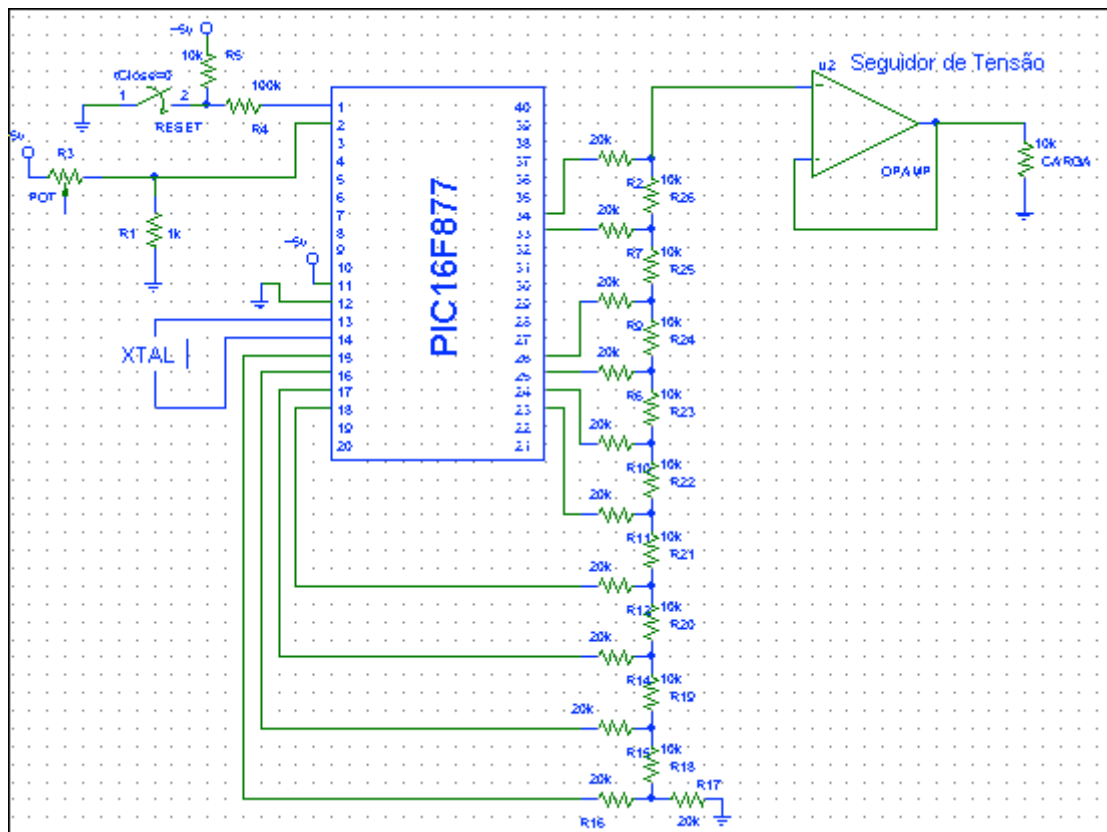


Figura 10 - Esquema de circuito

A entrada escolhida para teste do módulo AD é a “RA0”, o pino 2. Neste, conecta-se um circuito divisor de tensão simples com um resistor de 1k e um potenciômetro. Este circuito serve para variar a tensão analógica de entrada. O potenciômetro utilizado possibilita o controle desta tensão. As referências de tensão de entrada para conversão AD são:

- $V_{REF+} = 5V$;
- $V_{REF-} = 0V$.

O circuito para implementação de um conversor DA escolhido é uma rede R-2R, com resistores de 10k e 20k. Um amplificador operacional usado como seguidor de tensão e uma carga de teste resistiva de 10k é usada para observar a operação do conversor. A pouca precisão deste tipo de circuito é compensada por sua simplicidade e facilidade de implementação.

Um circuito divisor de tensão também é usado para “zerar” o programa do PIC, acionado por um botão simples, chamado de botão de “RESET”. Este circuito tem como

principal objetivo submeter o PIC16F877 a uma entrada analógica, testando seu módulo A/D de 10 bits, e escrever este valor no conversor D/A de 10 bits.

Programa escrito em linguagem C para teste

As principais tarefas do programa para implementação do circuito de teste referido são as seguintes:

- Configurar o módulo A/D para receber a entrada analógica;
- Escrever este valor nos seguintes pinos: (LSB) RC0, RC1, RC2, RC3, RC4, RC5, RC6, RC7, RB0 e RB1 (MSB).

Inicialmente, configura-se os registradores de uso geral do PIC: OPTION, INTCON e PIE. Depois, as portas de E/S PORTB e PORTC tem todos seus pinos configurados como saídas com o uso dos registradores TRISB e TRISC. As rotinas “InitializeADC()” e “ReadADC()” estão nos arquivos “adc.c” e “adc.h” e servem para inicializar um canal de entrada analógica e para ler o valor convertido, respectivamente. O resultado desta conversão AD é escrito diretamente nos registradores PORTC (8 bits menos significativos) e PORTB (2 bits mais significativos), pinos estes usados pelo conversor DA implementado.

3.3.2 Teste do LCD HD44780 utilizando o PIC16F877:

O circuito implementado em protoboard para teste do LCD HD44780, usando o PIC16F877 está demonstrado na figura 11:

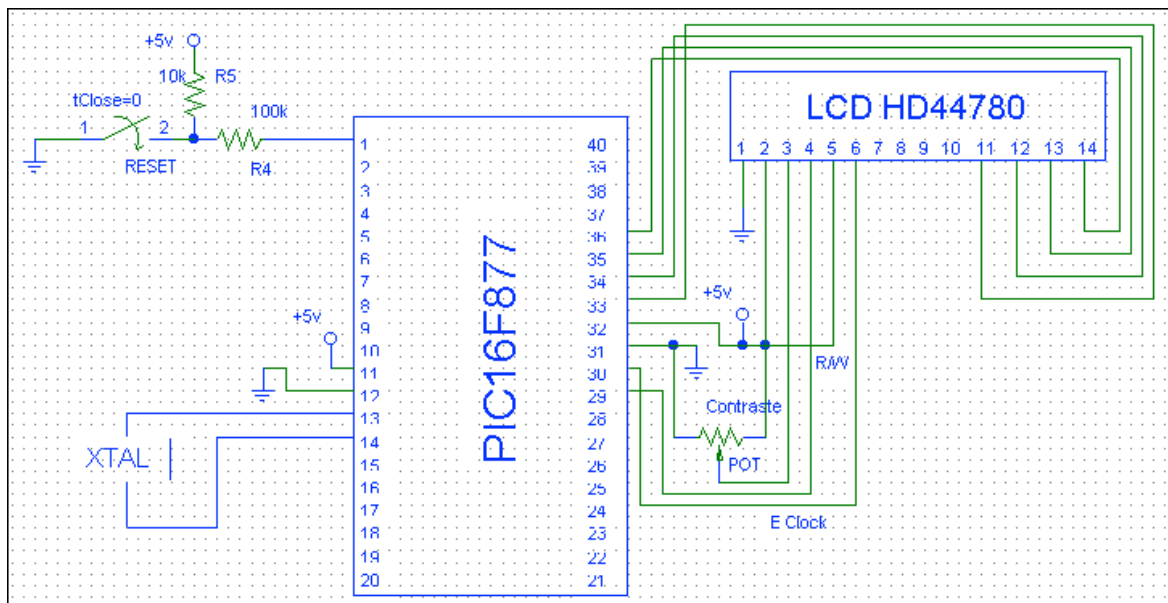


Figura 11 - Esquema de Circuito com LCD

Os pinos 7, 8, 9 e 10 não são utilizados no modo de operação de 4 bits. Apenas os pinos 11 à 14, os 4 bits mais significativos, são usados, sendo conectados aos 4 bits menos significativos do PORTB do PIC. O significado do restante dos pinos está detalhado na tabela abaixo:

Tabela 5 - Descrição dos pinos do LCD HD44780

Pino (LCD)	Descrição
1	Terra
2	+5V
3	Tensão de Contraste
4	R/S seleciona registrador (instrução ou dados).
5	R/W Ler ou escrever no LCD.
6	E-Clock (pulso de 450ns para iniciar transmissão de dados).

Para a obtenção da tensão de contraste, um potenciômetro é usado. As extremidades deste potenciômetro são conectadas a “0” e “5V” e o contato ligado diretamente ao pino 3 do LCD. Os pinos 4 e 6 são acionados via software e o pino 5 é conectado diretamente à tensão de 5V, que indica escrita no LCD.

Programa escrito em linguagem C para teste de LCD

As principais tarefas do programa para implementação do circuito de teste referido são as seguintes:

- Inicialização do LCD;
- Escrever 2 frases no LCD.

Inicialmente, configura-se os registradores de uso geral do PIC: OPTION, INTCON e PIE. Depois, as portas de E/S PORTB e PORTD tem alguns de seus pinos configurados como saídas com o uso dos registradores TRISB e TRISD. As rotinas “lcd_init()” e “lcd_puts” estão nos arquivos “lcd.c” e “lcd.h” e servem para inicializar via software o LCD e para escrever uma string no LCD, respectivamente. A rotina “lcd_clear()” apaga o conteúdo do LCD, “lcd_goto()” move o cursor para uma determinada posição e “lcd_write” escreve um dado em hexadecimal no LCD.

Para observação da escrita de “strings” no LCD, usa-se a rotina “DelayMs()” pertencente aos arquivos “delay.c” e “delay.h”. Esta rotina, de acordo com o valor da frequência do oscilador ou cristal usado, provoca um atraso em unidades de milissegundos. Com o uso do contador “count” incrementado do valor 0 até 999, este atraso proporcionado é feito em unidades de segundos.

3.4 Sistema Objeto de Estudo – Maquete de Escritório com Aparelhos de Ar - condicionado

O sistema térmico usado para controle de temperatura é uma maquete representando um escritório. Esta maquete foi confeccionada em laboratório contendo diversas divisórias, representando as divisórias existentes em um escritório que delimitam as salas, portas entre as salas e o corredor e janelas entre as salas e o ambiente externo. A figura 12 mostra um esquemático desta maquete.

Para representar os sistemas de condicionamento de ar, são utilizados dois secadores de cabelo de potência de “1kW”, posicionados da seguinte maneira: um no corredor, outro dentro da sala com maior área interna. Para controlar a temperatura dentro destas salas, interfaces adequadas entre o processo térmico e os acionamentos do sistema devem ser projetadas. Estas interfaces devem ser recebidas por uma unidade de processamento que deverá receber um valor de temperatura correspondente à

temperatura no ambiente que se deseja controlar e enviar um sinal para os acionamentos de forma que uma temperatura de referência seja atingida de acordo com critérios de desempenho especificados.

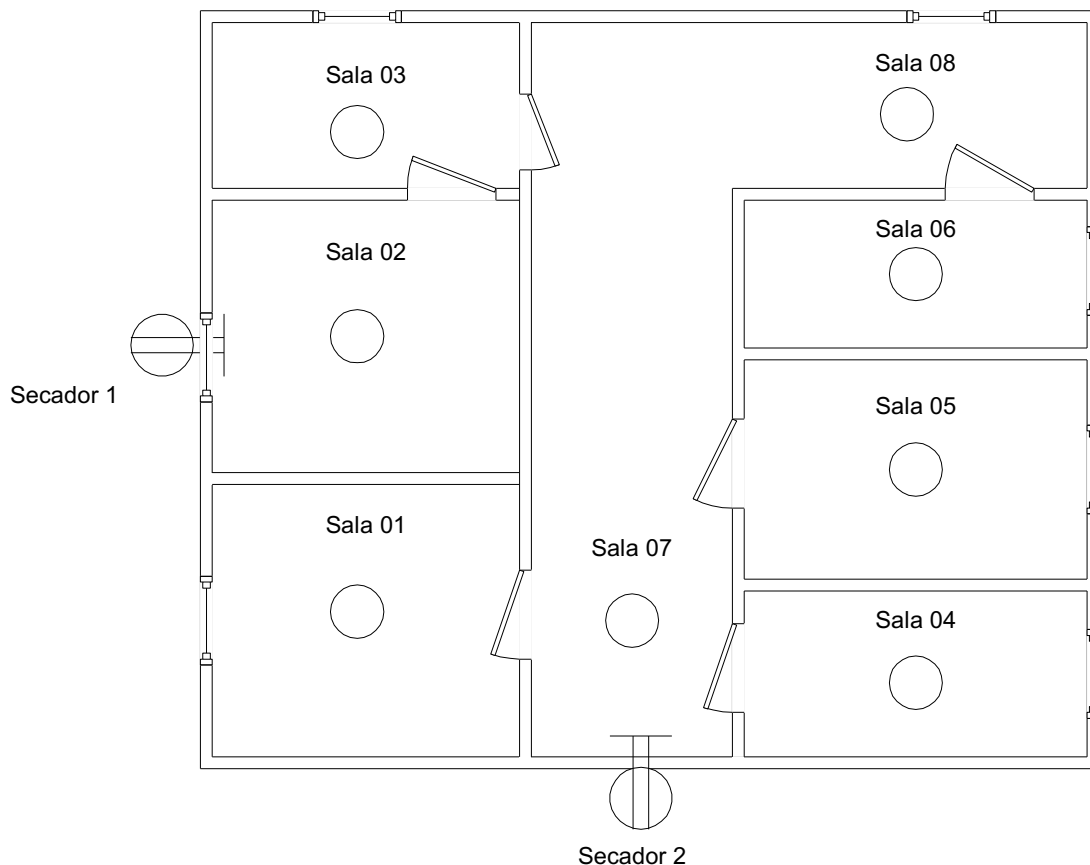


Figura 12 - Maquete usada como sistema térmico

Para este processo, deseja-se controlar a temperatura das salas “01” a “06”, a temperatura no início do corredor, a “sala 07”, e a temperatura no fim do corredor chamado de “sala 08”. Para cada um destes ambientes, um sensor de temperatura deve ser posicionado. Os sensores são posicionados nas posições especificadas por círculos na figura 12, localizados aproximadamente no centro das salas.

O presente trabalho restringe-se ao controle de temperatura dos processos concernentes ao controle de temperatura das salas “2” e “5” através dos atuadores “secador 1” e “secador 2”, respectivamente. O sistema então consiste de dois processos

de primeira ordem “SISO” – single input single output. O sinal de controle enviado para os atuadores considera apenas a temperatura medida na sala especificada.

3.4.1 Sensores de Temperatura Utilizados

Os sensores de temperatura adotados são sensores em circuitos integrados cuja tensão de saída é diretamente proporcional à temperatura onde os sensores estão, em graus Celsius. Os sensores não precisam de posterior calibração, fornecendo valor de precisão típica de $\pm 1/4^{\circ}\text{C}$ para diversos pontos de operação e $\pm 3/4^{\circ}\text{C}$ como valor máximo. Algumas das características principais de operação deste sensor estão destacadas na tabela 6.

Tabela 6 - Sensor de temperatura LM35

Característica	Valor
Relação ($^{\circ}\text{C}/\text{V}$)	$100^{\circ}\text{C} \rightarrow 1000\text{mV}$
Fator de escala	$10\text{mV}/^{\circ}\text{C}$
Faixa de operação	-55°C a 150°C
Precisão	0.5°C (em 25°C)
Não-linearidade	0.25°C (valor típico)
Consumo de corrente	Menos de $60\mu\text{A}$
Alimentação	Entre 4 e 20V

O sensor utilizado possui encapsulamento “TO-92”, com um dos terminais conectado a referência do sistema, o outro terminal conectado à “5V” e o terminal restante possui o valor de tensão correspondente à temperatura onde o sensor está inserido. Este sensor pode ser caracterizado como um ganho relacionando as grandezas temperatura e tensão, tendo a temperatura como entrada e respondendo com uma tensão de saída. Este ganho vale:

$$K_{\text{SENSOR}} = \frac{1000\text{mV}}{100^{\circ}\text{C}} = 10 \frac{\text{mV}}{^{\circ}\text{C}} \quad (6)$$

Este sensor, de fácil manuseio e operação, facilitou seu emprego no presente trabalho. O fator que limita seu uso é o valor de sua precisão: 0.5°C . Ou seja, variações

de temperatura dentro deste intervalo não são perceptíveis por variação de tensão na saída do sensor.

3.4.2 Acionamentos e projeto de “drivers” de potência

Os atuadores escolhidos para fornecimento de fluxo de calor para o processo térmico são dois secadores de cabelo de “1kW” cada. O circuito interno destes secadores pode ser esquematizado pela figura 13. Um diodo é usado para retificar a tensão da rede elétrica em modo “meia-onda”. Outro diodo é conectado reversamente aos terminais do motor que serve como ventilador do secador de cabelo, servindo como prevenção de transitórios presentes em motores durante sua partida e operação. O motor usado como ventilador do secador de cabelo é um motor DC com tensão de alimentação de “12VDC”. A resistência possui valor de “100Ω”, para ambos os secadores. Esta resistência aquece, ao passo que o ventilador puxa ar do ambiente e faz este ar passar pela resistência aquecida. Este procedimento fornece o fluxo de calor desejado.

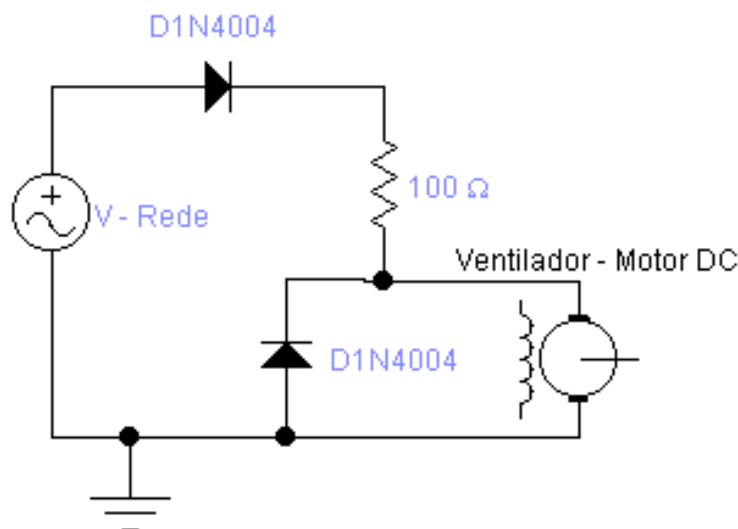


Figura 13 - Esquema de circuito de um secador de cabelo típico

Para este atuador, um “driver de potência” teve de ser projetado, na medida em que a unidade de processamento, o microcontrolador PIC, não pode fornecer esta quantidade de potência para cada um dos secadores. O projeto destes drivers também deve considerar quais as interfaces são disponibilizadas pela unidade de processamento usada. No caso do “PIC16F877”, apenas saídas digitais e duas saídas com sinal PWM

são disponibilizadas. Uma alternativa seria usar uma quantidade de saídas digitais e converter o sinal digital em analógico, como no circuito de rede R-2R apresentado na figura 3. Esta alternativa ocupa muitas portas de entrada e saída de dados e para acionar os dois secadores, usando os mesmos 10 bits disponibilizados pelo conversor A/D do microcontrolador, seriam necessários 20 pinos do PIC16F877, mais da metade dos pinos de entrada e saída disponíveis.

Usar as saídas com sinais PWM mostrou-se, então, uma alternativa mais interessante por economizar portas de entrada e saída e possibilitar o projeto de “Choppers DC” como drivers de potência. Nestes circuitos, um transistor é usado como chave para ligar e desligar uma carga à alimentação a uma determinada frequência, seguindo um valor de ciclo de trabalho de acordo com a potência que se deseja entregar à carga. A tensão contínua é fornecida ligando-se a rede elétrica de “220Vrms” a um retificador de onda completa e, em seguida, dois capacitores para redução de ondulação e fornecimento de uma tensão contínua mais elevada. Então, o “Chopper” escolhido é um tipo “Buck” ou abaixador, já que a carga será ligada a uma tensão contínua máxima. A figura 14 mostra o esquema do chopper abaixador implementado.

O transistor usado como chave, ligado em série à carga, necessita de uma corrente de base mais alta do que a disponibilizada pelas saídas do microcontrolador. Faz-se necessário o emprego de mais um transistor que forneça ganho de corrente suficiente para acionamento do transistor de potência. A tabela 7 indica quais modelos de componentes foram utilizadas, descrições sucintas das suas características básicas e os símbolos correspondentes as suas posições no circuito da figura 14.

Tabela 7 - Dispositivos usados para implementar driver de potência

Elemento – símbolo	Modelo	Características
Ponte de Diodos – D13 + D14 + D15 + D16	-	$V_r = 400V_{dc}$ e $I_d = 10A$
Transistor de Potência – Q1	C3831	$V_{ceo} = 400V_{dc}$ e $I_c = 10A$
Transistor para ganho de corrente – Q9	BC 547	$V_r = 45V_{dc}$ e $I_d = 100mA$
Diodo de proteção – D17	D1N4004	$V_r = 400V_{dc}$ e $I_d = 1A$
Resistores – R2 e R13	-	50Ω e 100Ω respectivamente
Capacitores – C1	-	$220\mu F$ e $300V$ cada

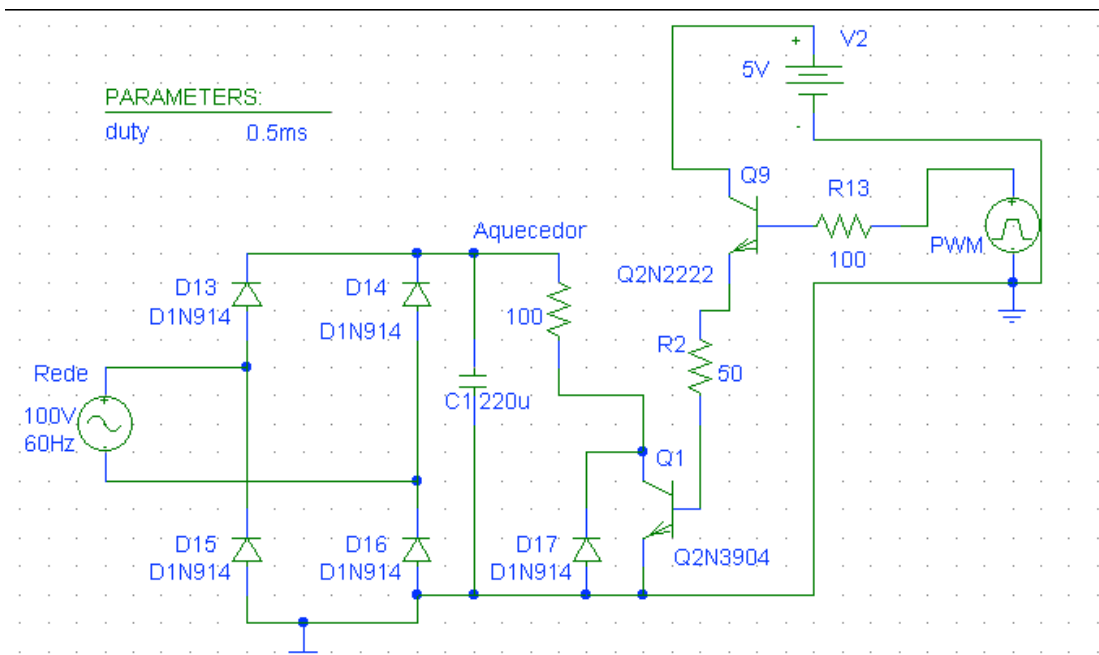


Figura 14 - Driver de Potência projetado

A simulação do circuito da figura 14, usando como sinal PWM de entrada uma onda com ciclo de trabalho de 50%, produz o gráfico da figura 15. Este gráfico também fornece a informação do valor médio da tensão submetida à carga, aproximadamente metade da tensão total disponível, validando o circuito projetado. A simulação usou como tensão de entrada uma onda senoidal de “100V” de amplitude e “60Hz”.

O gráfico da figura 15 mostra que o objetivo de entregar à carga determinada potência é atingido com o emprego do circuito da figura 14. O valor médio da tensão submetida à carga, de “45Vdc”, é equivalente à metade da tensão DC obtida pela retificação em onda completa seguida de um capacitor, cerca de “90Vdc”, ou seja, “50%” da tensão máxima, correspondendo ao valor do ciclo de trabalho da onda PWM usado como entrada. A partir disto, pode-se chegar a uma relação entre ciclo de trabalho e tensão média sobre a carga.

$$\overline{V_{CARGA}} = \frac{1}{T} \int_0^T v(t) dt = \frac{T_{ON}}{T} V_{MAX} = \frac{duty_cycle(\%)}{100\%} \cdot V_{MAX}$$

Alguns outros elementos de circuito foram adicionados para proteção dos dispositivos semicondutores, sensíveis a alterações bruscas de tensão presentes. O transistor de potência é chaveado entre “0 e 300V”. Os seguintes dispositivos foram

adicionados: diodo de proteção para a junção coletor emissor; fusível de 400V e 4A; varistores; filtros de linha; dissipadores para os transistores de potência.

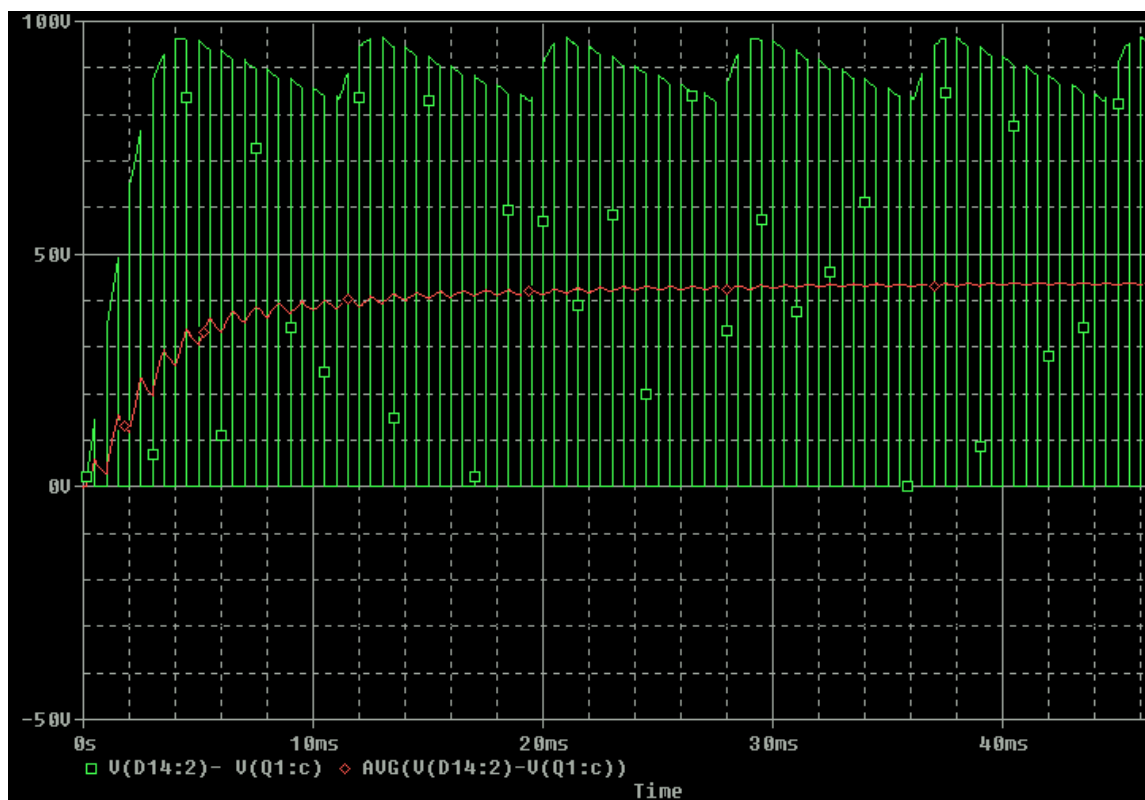


Figura 15 - Resultado de simulação do driver de potência - Tensão sobre a carga e seu valor médio

A entrada PWM é fornecida pelas saídas do microcontrolador. Este sinal possui frequência fixa de aproximadamente “1kHz” e o valor do ciclo de trabalho é ajustado pela escrita, via software, em um registrador de resolução de “10bits” que determina o tempo durante o qual o sinal PWM fica em nível alto, “ T_{ON} ”.

3.4.3 Unidade de Processamento

Como foi dito anteriormente, a unidade responsável pelo processamento dos sinais adquiridos através dos sensores térmicos e envio de sinal modulado por largura de pulso para acionamento dos atuadores é um microcontrolador da família PIC modelo 16F877.

A recepção do sinal dos sensores térmicos é feita através de dois dos oito canais de conversão analógico-digital disponíveis. A taxa de conversão máxima permitida pelo conversor interno do microcontrolador é de 75 μ s/amostra, quando o mesmo é utilizado com um sinal de *clock* de “20MHz”. O sinal de *clock* fornecido ao microcontrolador é

de “18MHz”, resultando em uma taxa de conversão máxima de aproximadamente “83.3 μ s/amostra”.

Os conversores A/D possuem resolução de “10bits” em uma faixa de leituras de “0” a “5V”. Portanto, a menor variação de tensão perceptível pelo PIC é de:

$$\Delta V_{MIN} = \frac{5}{2^{10}} = \frac{5}{1024} \cong 4.88mV$$

Este valor é menor do que o máximo erro permitido pelo sensor de temperatura calibrado, “0,5°C”, correspondendo a “5mV”. Uma alternativa seria usar um amplificador entre o sensor e o microcontrolador, com um certo ganho, para aproveitar a faixa de leitura do conversor A/D. Isto, porém, não seria justificável na medida em que a precisão do sensor seria bem menor que a resolução do conversor. O resultado observado seria diversos valores de leitura gerados pelo conversor para um mesmo valor de temperatura enviado pelo sensor. A solução adotada mostra-se, portanto, mais simples e com menor custo associado.

O sinal PWM é gerado pelo microcontrolador com frequência fixada em aproximadamente “1kHz”. Este valor de frequência é o mínimo possível quando se adota um sinal de *clock* de “18MHz”. A frequência máxima permitida para o sinal PWM é de “187.4kHz”, mas esta alta frequência reduz a resolução do valor do ciclo de trabalho para “5bits”.

O valor do ciclo de trabalho deste sinal é ajustado através da escrita em dois registradores “SFR” – CCPR1L e CCPR2L. Este valor possui resolução máxima de “10bits”. Em termos de variação percentual do ciclo de trabalho, a resolução máxima do PIC é de:

$$\Delta DutyCicle = \frac{100\%}{2^{10}} = \frac{100\%}{1024} \cong 0.097\%$$

O desenvolvimento de “software” para implementação dos controladores é feito usando o ambiente “HPDPIC”, como está detalhado na seção 3.2. Um esquema do processo a ser controlado pode ser esquematizado pela figura 16. A implementação dos controladores é feita por equações de diferenças e construção de bibliotecas em linguagem C para facilitar a modularização dos programas e possibilitar a reutilização dos algoritmos desenvolvidos.

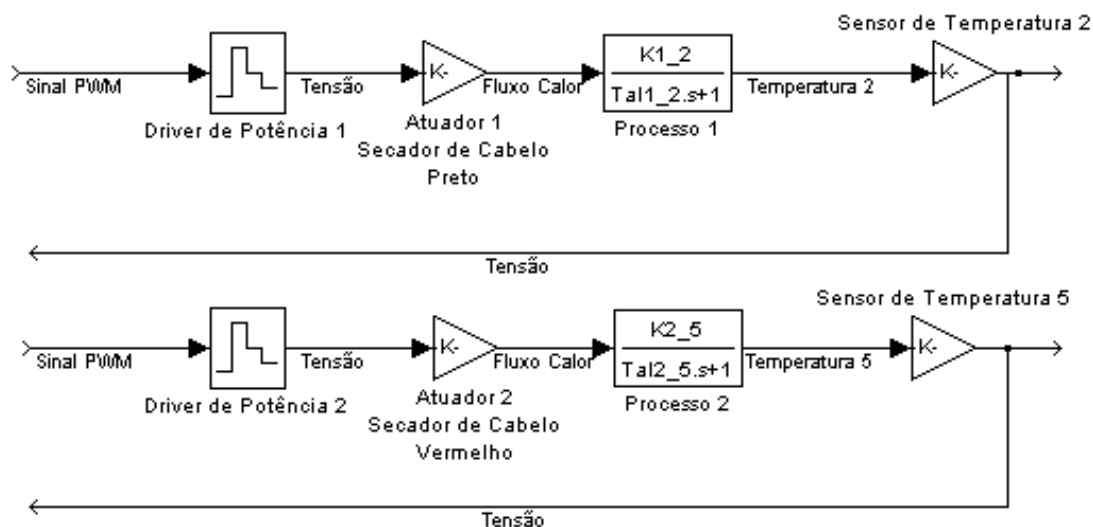


Figura 16- Esquemas dos processos térmicos

Introduzindo o microcontrolador através das interfaces dos processos, o esquema para o sistema de controle em malha fechada é representado pela figura 17. Nesta, um controlador digital é usado para receber o valor de erro, gerado internamente pela diferença entre a temperatura de referência e a temperatura medida, e calcular o valor de ciclo de trabalho para anular este erro.

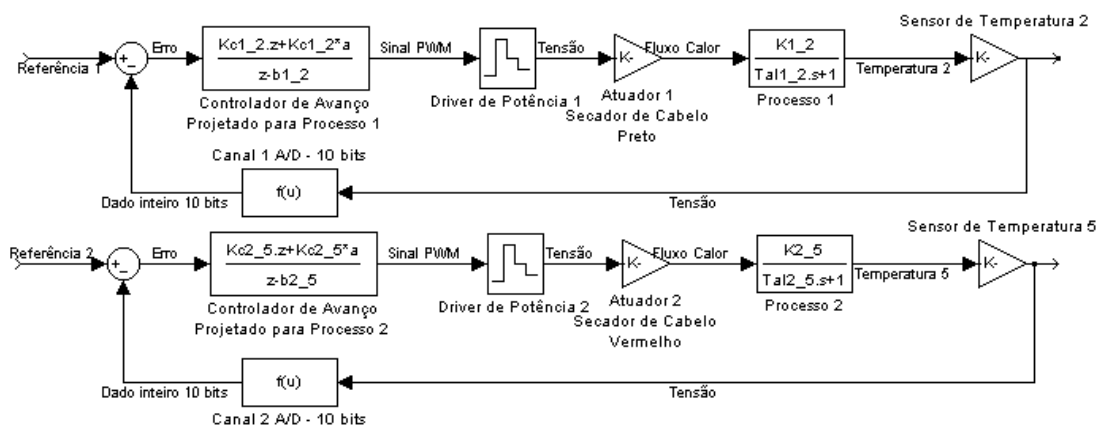


Figura 17 - Processos térmicos com controladores digitais

Toda esta parte presente na figura 17, que foi adicionada aos esquemas da figura 16, é implementada pelo software gravado no microcontrolador. Para mudança de referência e parâmetros dos controladores, novos programas têm de ser compilados e gravados no PIC.

3.4.4 Apresentação de Dados

Para recepção dos dados de temperatura dos processos controlados, diversas alternativas são possíveis. Poderiam ser utilizadas as portas paralela ou serial disponibilizadas pelo PIC para escrita de dados em memória de um computador pessoal para posteriores análises. Outra alternativa seria usar uma placa de aquisição de dados instalada em um computador que tivesse alguns canais de entrada analógica para leitura das tensões dos sensores de temperatura. Também seria possível a escrita em “displays” de cristal líquido – LCD’s – ou através de LED’s. Isto dificultaria a análise posterior dos dados do processo controlado por não fornecer nenhum meio de recuperação destes dados para produção de gráficos.

A escrita em memória EEPROM do microcontrolador também não se mostrou uma alternativa razoável, pela pequena quantidade de memória disponível neste modelo, 256 bytes. Isto tornaria a análise dos dados dos processos bem restrita devido ao pequeno tamanho da memória.

A placa de aquisição de dados PCI-6024E, do fabricante “National InstrumentsTM” foi escolhida por possuir um software pronto para usá-la e interpretar os dados por ela fornecidos. O software “LabVIEWTM”, do mesmo fabricante da placa de aquisição, é compatível com a placa e os canais de entrada da placa podem ser colocados diretamente como fontes de dados nos diagramas de blocos implementados no “LabVIEW”. Isto facilita a apresentação dos dados na tela do computador em gráficos, auxiliando a visualização do andamento da dinâmica dos processos controlados. Além disso, este software é uma ferramenta de programação visual e rápida. A programação do software LabVIEWTM para apresentação dos dados consiste na criação de um arquivo com extensão “vi”, que significa “Virtual Instrument” – instrumento virtual. Neste instrumento virtual, o programador coloca os controles e interfaces com o usuário no painel frontal. Após isto, o mesmo programador coloca no diagrama de blocos a lógica das interfaces com o usuário, indicando qual interface apresentará quais dados e durante quanto tempo.

A operação desta placa de aquisição segue os seguintes passos, para cada sensor de temperatura conectado:

- Cadastrar a entrada através do software de gerenciamento de dispositivos – MAXTM – Measurements and Automation Explorer

- Informar que tipo de entrada, no caso, um sensor de temperatura;
- Informar qual a faixa de operação do sensor e, no caso de sensores térmicos, qual a função de transferência que relaciona tensão à temperatura medida;
- Usar esta entrada como entrada para gráficos ou até arquivos para posteriores análises.

Para o presente trabalho, apenas as entradas concernentes aos processos objeto de estudo, a temperatura das salas “02” e “05”, são usadas em gráficos e gravadas em arquivos texto. Nenhum dos sinais internos ao microcontrolador é recebido pela placa de aquisição de dados, usada apenas na função de apresentação dos dados convenientes e supervisão da operação do sistema. Entretanto, esta placa também poderia executar as funções de controle do processo, por possuir canais de comunicação de dados que servem tanto para entrada de dados como para saída.

A interface produzida para apresentação das temperaturas nas salas “02” e “05”, está mostrada na figura 18. A figura 19 mostra o diagrama de blocos produzido que contém a lógica de funcionamento do instrumento virtual projetado. O único controle presente indica quando a captura de dados deve parar através do botão “PARE”.

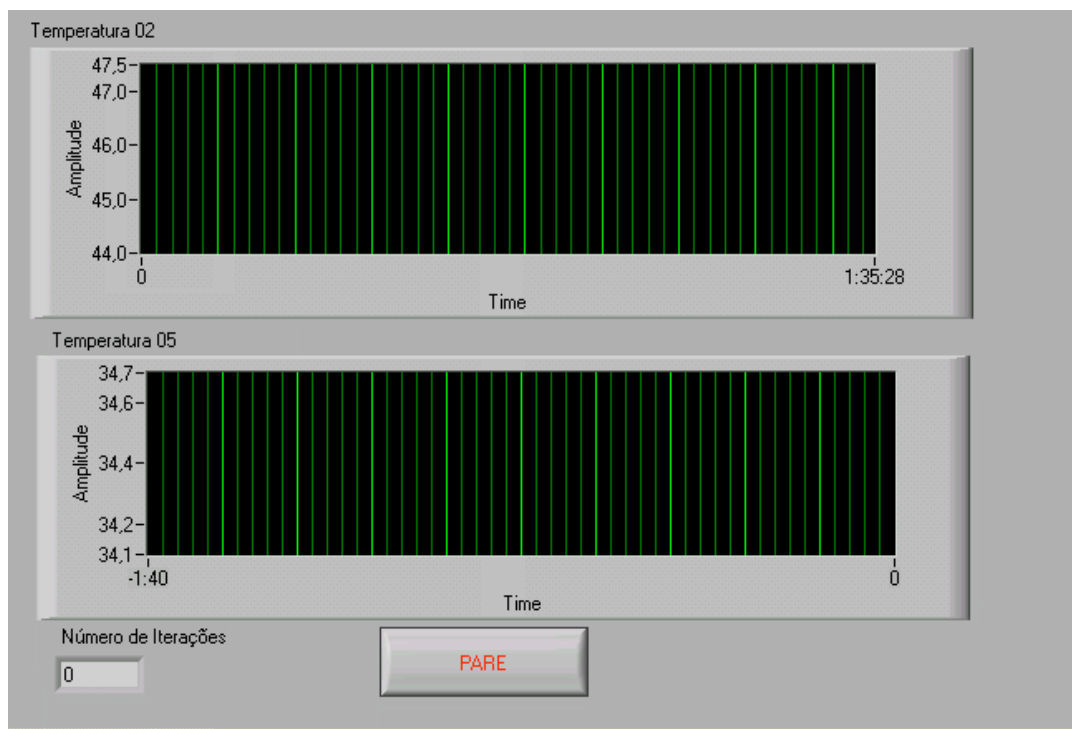


Figura 18 - Interfaces criadas para observação das temperaturas desejadas

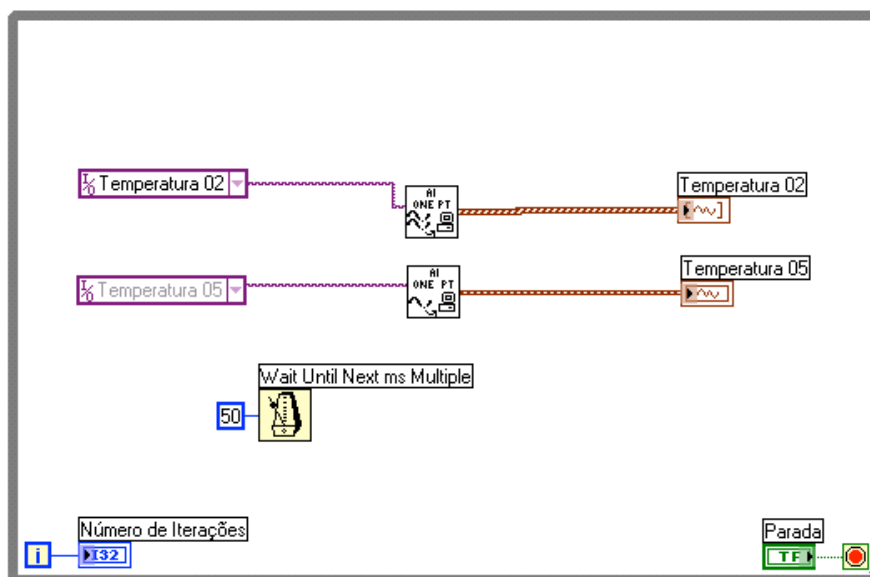


Figura 19 - Diagrama de blocos com a lógica de apresentação de dados

4. Resultados Obtidos

4.1 Modelagem do sistema físico

Para controle dos processos térmicos, a função de transferência dos mesmos, ou pelo menos uma boa estimativa desta, tem de estar disponível para tornar possível o uso de softwares de simulação, auxiliando no projeto dos controladores para os processos. Como foi colocado na seção 2.5, o processo térmico pode ser caracterizado por uma função de transferência de primeira ordem. Representando por um ganho a função de transferência dos elementos “driver de potência” e “atuador” em série, o processo térmico com atuador recebe um sinal PWM e tem como resposta um valor de temperatura. Esta função de transferência pode ser caracterizada por:

$$\frac{\text{Temperatura}}{\text{Duty_Cicle}} = \frac{Kp}{\tau s + 1} \quad (7)$$

Para cada um dos processos controlados, estes parâmetros devem ser determinados experimentalmente ou analiticamente. Através do método experimental, as seguintes denominações são feitas:

- $\tau_{i,j}$: constante de tempo do processo com atuador “i” da sala “j”;
- $Kp_{i,j}$: ganho que relaciona diretamente a temperatura máxima da sala “j” atingida quando o sinal PWM funciona em 100% de ciclo de trabalho o atuador i.

O sensor de temperatura pode ser representado por um ganho que relaciona a temperatura medida à tensão fornecida em um dos seus terminais. Este ganho possui valor de “10mV/°C”.

A determinação de “ $Kp_{i,j}$ ” vem da observação da temperatura medida quando o ciclo de trabalho da onda PWM apresentada é mantido em “100%” por um intervalo de tempo até que a temperatura estabilize em um valor. A medição experimental da constante de tempo dos processos é a quantidade de tempo necessária para o sistema, quando submetido ao sinal PWM com valor de ciclo de trabalho de “100%”, atingir “63.6%” da temperatura máxima possível. Seguindo estes procedimentos, a tabela 8 foi determinada:

Tabela 8- Parâmetros dos processos estudados

Parâmetro	Valor
$\tau_{1,2}$	40.2 segundos
$Kp_{1,2}$	77.9 °C/100% de ciclo de trabalho
$\tau_{2,5}$	175 segundos
$Kp_{2,5}$	30.2°C/100% de ciclo de trabalho

As funções de transferência determinadas para os processos são as seguintes:

$$G_{p12} = \frac{0.779}{40.32s + 1} \quad (8)$$

$$G_{p25} = \frac{0.302}{175s + 1} \quad (9)$$

4.2 Simulação de Sistemas em Malha Aberta

O modelo usado para simulações é descrito pelas funções de transferência determinadas na seção 4.1. O lugar geométrico das raízes para estes sistemas está mostrado nas figuras 20 e 21. As respostas dos sistemas em malha aberta são apresentadas nas figuras 22 e 23. A ferramenta usada para simulação dos sistemas é o software “Simulink”, parte integrante do aplicativo “MATLABTM”.

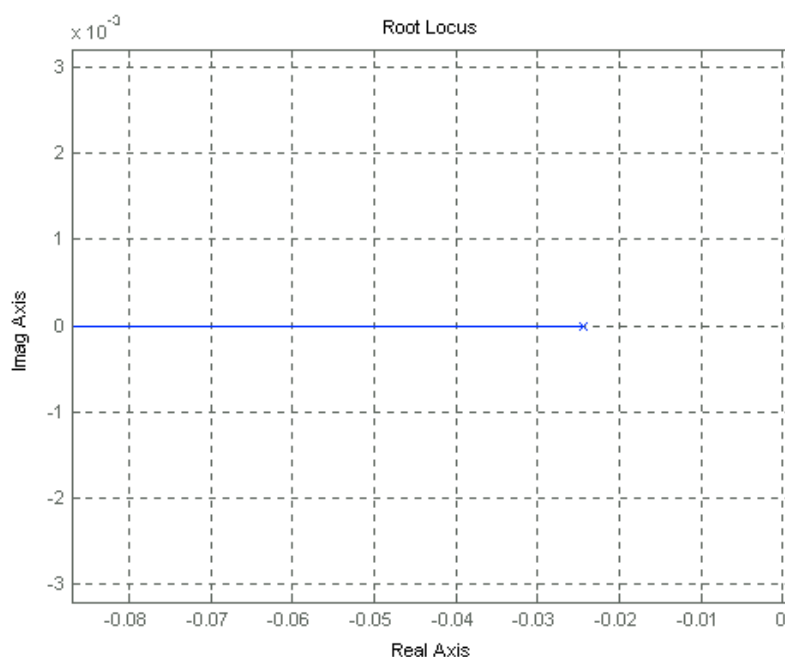


Figura 20- Lugar geométrico das raízes do processo 1,2

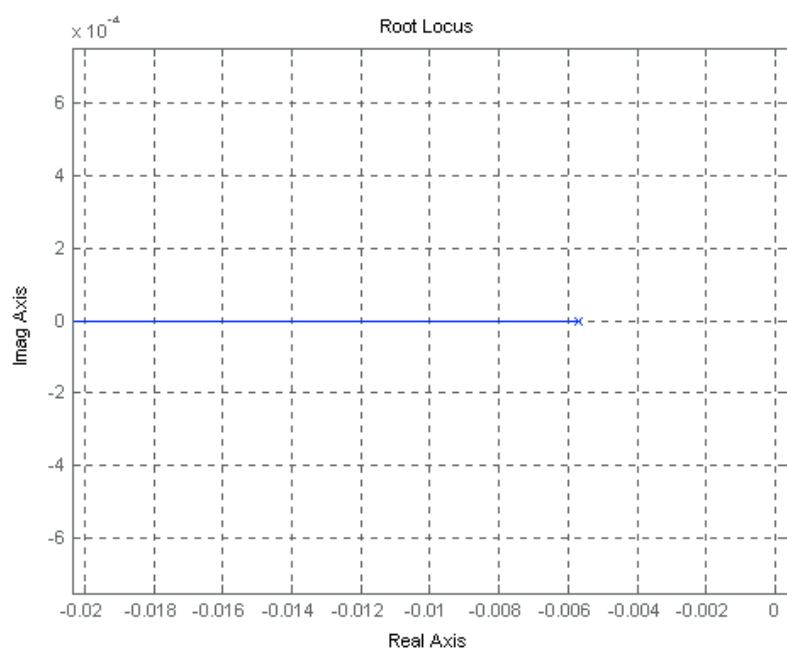


Figura 21 - Lugar geométrico das raízes do processo 2,5

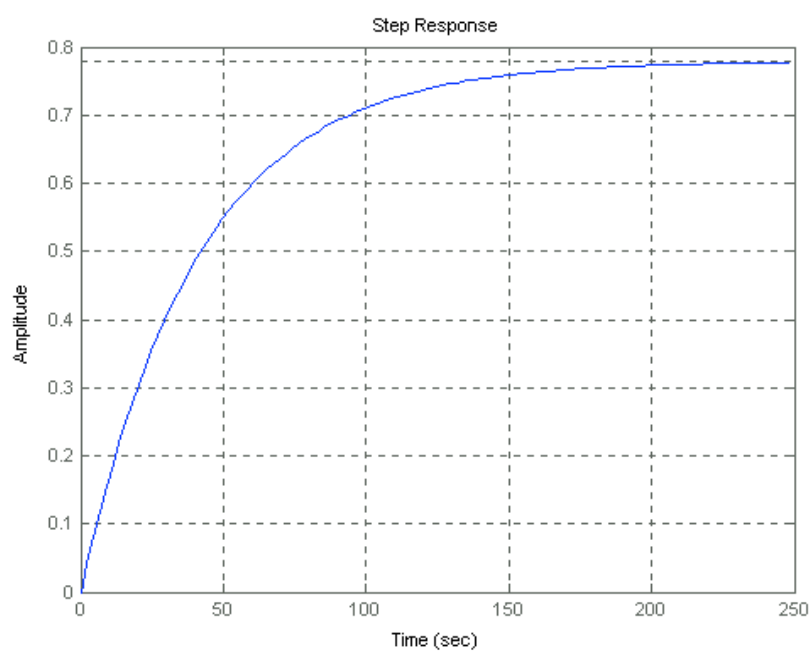


Figura 22 - Resposta ao degrau unitário do sistema 1,2 em malha aberta

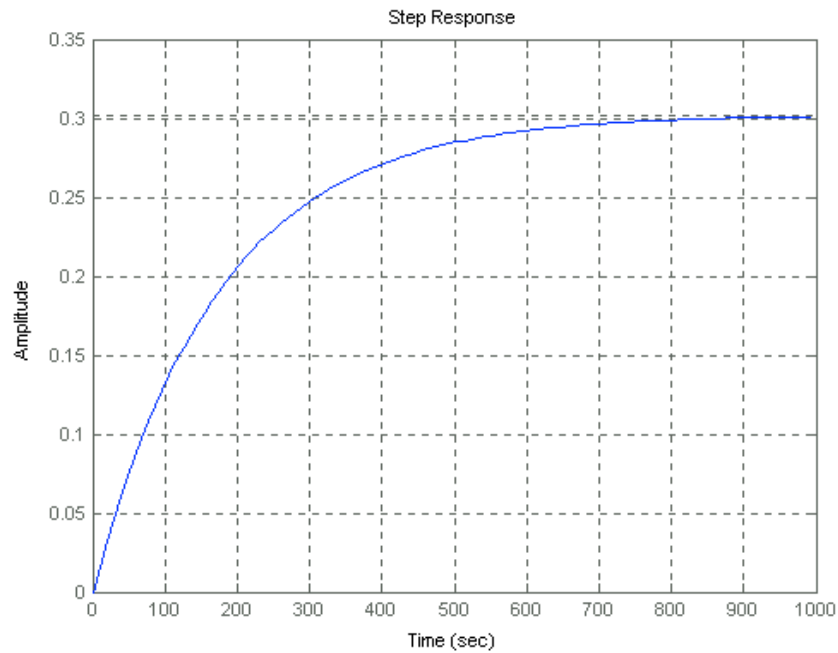


Figura 23 - Resposta ao degrau unitário do sistema 2,5 em malha aberta

4.3 Projeto dos Controladores

- **Taxa de Amostragem adotada**

A taxa de amostragem do algoritmo de controle é determinada de acordo com o número de instruções contidas no algoritmo de controle e a velocidade de execução de cada instrução de máquina. O laço de controle PI por equações de diferenças para geração de sinal de controle para os dois processos possui cerca de 500 instruções de máquina. Como mencionado anteriormente, uma instrução demora “4” ciclos de *clock* para ser executada pelo microcontrolador. A velocidade de execução de cada instrução é de:

$$T_I = \frac{4}{18.10^6} = 0.222\mu s \quad (10)$$

Entretanto, deve-se considerar que dentro do laço de controle são feitas duas conversões A/D, cada uma demorando “83,3 μs ”. Portanto, a taxa de amostragem máxima do controlador é de:

$$T_{AMOST} = 500 * T_I + 2 * T_{AD} = 111.11 + 2 * 83.3 = 277,71\mu s \quad (11)$$

Este resultado equivale a “3.6kHz”, uma taxa bastante elevada com relação às constantes de tempo do sistema. Com esta taxa, os controladores podem ser projetados no domínio “*quasi-contínuo*”.

O controle é, então, implementado em software através de equações de diferenças. A equação de diferenças equivalente a um controlador PI possui a seguinte estrutura:

$$U(k) = U(k-1) + (Kp + Ki.T_{AMOST})E(k) - Kp.E(k-1) \quad (12)$$

Para o controlador proporcional *Fuzzy* com canal integral, a seguinte equação é usada para cálculo do sinal de controle:

$$IE(k) = IE(k-1) + E(k) \quad (13)$$

$$U(k) = fuzzyP(E(k)) + IE(k) \quad (14)$$

O sinal de controle do canal nebuloso é feito através da chamada da função “fuzzyP()” que contém o algoritmo com lógica difusa, base de regras, funções de pertinência e método de *defuzzificação* adotado.

- **Projeto do controlador PI – proporcional integral**

Os controladores PI para os processos térmicos são projetados de acordo com a seguinte ordem de passos:

- Especificações de desempenho em malha fechada;
- Simulação do sistema para entrada degrau;
- Aplicação do controlador no sistema físico.

Especificações de desempenho definidas:

- Sobrepasso: 2%;
- Tempo de subida: 50 segundos para processo “1,2” e 180 segundos para processo “2,5”

O sobrepasso possui valor baixo para que o sistema não fique muito lento quando a temperatura ultrapassar o valor de referência. Para atingir estas especificações, os seguintes pólos de malha fechada devem ser dominantes:

$$P_{1,2} = -0,0682 \pm j0,0548 \quad (15)$$

$$P_{2,5} = -0,0171 \pm j0,0137 \quad (16)$$

A função de transferência para o sistema com controlador em malha aberta é da seguinte forma:

$$FTMA(s) = \frac{Kp_{i,j}K_{i,j}(Ti_{i,j}s + 1)}{Ti_{i,j}(\tau_{i,j}s + 1)} \quad (17)$$

As figuras 24 e 25 mostram o lugar geométrico das raízes para os sistemas estudados que possuem esta função de transferência em malha aberta. Em malha fechada, utilizando realimentação negativa unitária, a função de transferência para o processo controlado é a seguinte:

$$FTMF(s) = \frac{Kp_{i,j}K_{i,j}}{Ti_{i,j}\tau_{i,j}} \frac{(Ti_{i,j}s + 1)}{s^2 + \left(\frac{1 + Kp_{i,j}K_{i,j}}{\tau_{i,j}}\right)s + \frac{Kp_{i,j}K_{i,j}}{Ti_{i,j}\tau_{i,j}}} \quad (18)$$

Estes pólos desejados são atingidos com os seguintes valores de “Kp” e “Ti” dos controladores PI:

$$Kp_{1,2} = 5.8824 \quad (19)$$

$$Ti_{1,2} = 14.6321 \quad (20)$$

$$Kp_{2,5} = 16.4518 \quad (21)$$

$$Ti_{25} = 59.3596 \quad (22)$$

As funções de transferência em malha aberta para cada um dos processos controlados são as seguintes:

$$FTMA_{1,2}(s) = 5.8824 \left(1 + \frac{1}{14.63s}\right) \frac{0.779}{40.32s + 1} \quad (23)$$

$$FTMA_{25}(s) = 16.4518 \left(1 + \frac{1}{59.3596s}\right) \frac{0.302}{175s + 1} \quad (24)$$

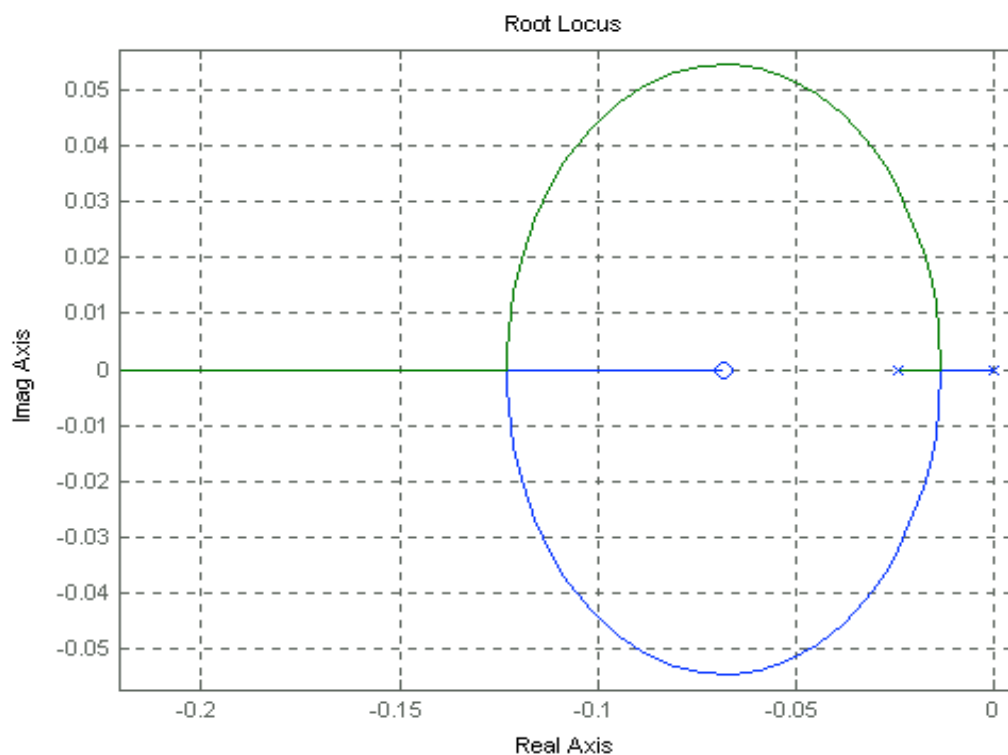


Figura 24 - LGR do processo 1,2 com controlador PI calculado

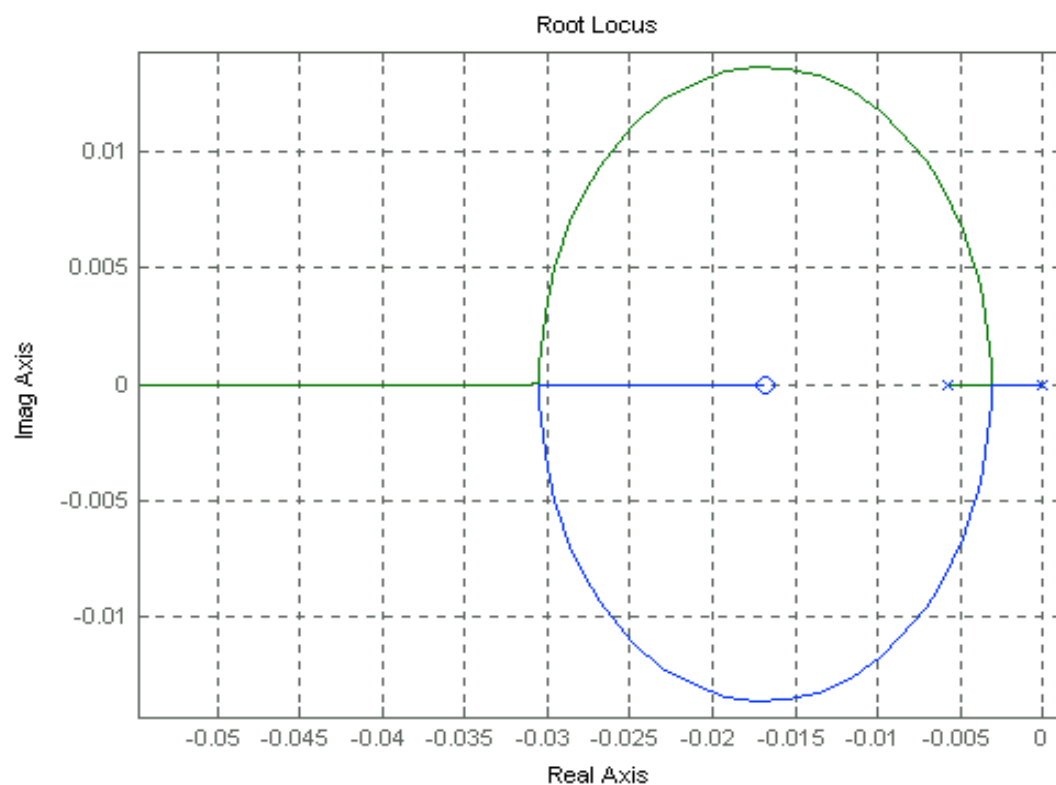


Figura 25 - LGR do processo 2,5 com controlador PI calculado

- **Projeto do controlador Proporcional *Fuzzy* com canal integral**

O canal integral é adicionado em paralelo ao controlador nebuloso para anular erro de regime permanente. Este apenas acumula o erro obtido a cada iteração e é somado ao canal *fuzzy*. Para que o valor no acumulador não atinja valores elevados e anule o efeito do controlador difuso proporcional, o canal integral é limitado entre os valores “-100 e 100”, correspondentes aos valores de saturação dos atuadores, que podem ser acionados somente por uma onda PWM com ciclo de trabalho com valores entre “0 e 100%”.

O canal que contém lógica *fuzzy* é projetado de acordo com a seguinte ordem de passos:

- Definição das funções de pertinência da entrada e saída;
- Definição da base de regras;
- Métodos de defuzzificação, implicação e agregação.

O mesmo controlador projetado é usado para os dois processos. Isto auxilia na implementação dos algoritmos, que possuem número de instruções elevado em comparação às simples equações de diferenças utilizadas para os controladores PI.

A entrada “erro” do canal *fuzzy* possui as funções de pertinência detalhadas na figura 26. Na figura 27, estão representadas as funções de pertinência da variável de saída do sistema de inferência nebuloso. A base de regras implementada está descrita na figura 28. Os métodos adotados pelo sistema de inferência *fuzzy* simulado e pelo sistema implementado são apresentados na tabela 9.

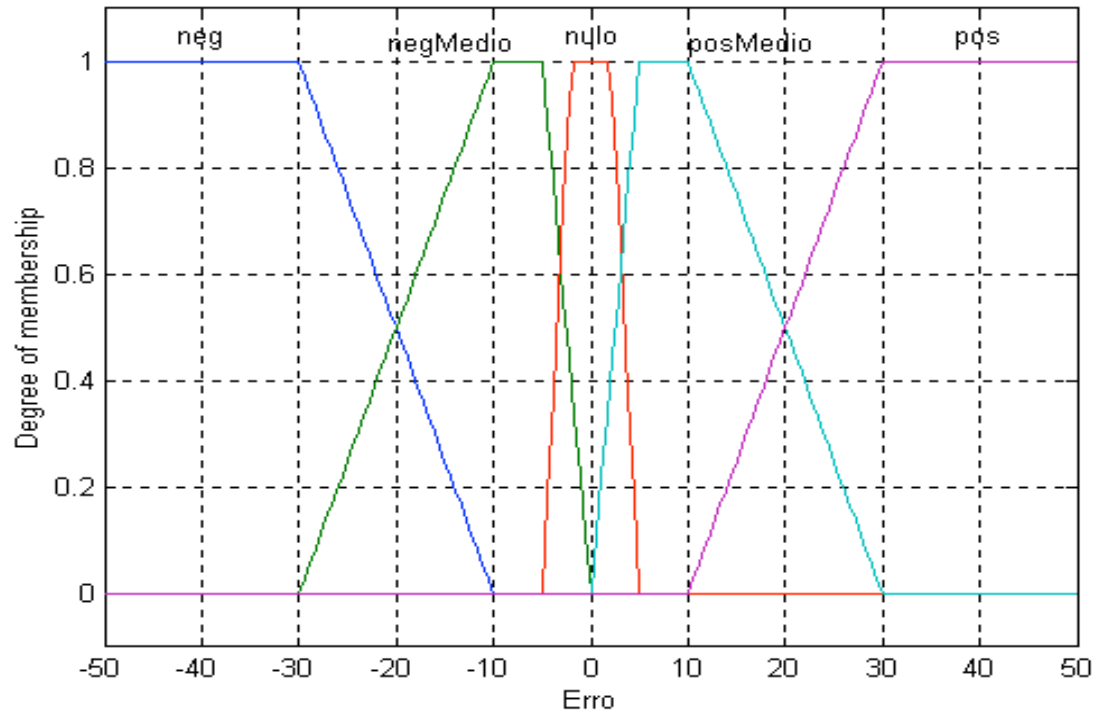


Figura 26 - Funções de pertinência da variável de "Erro"

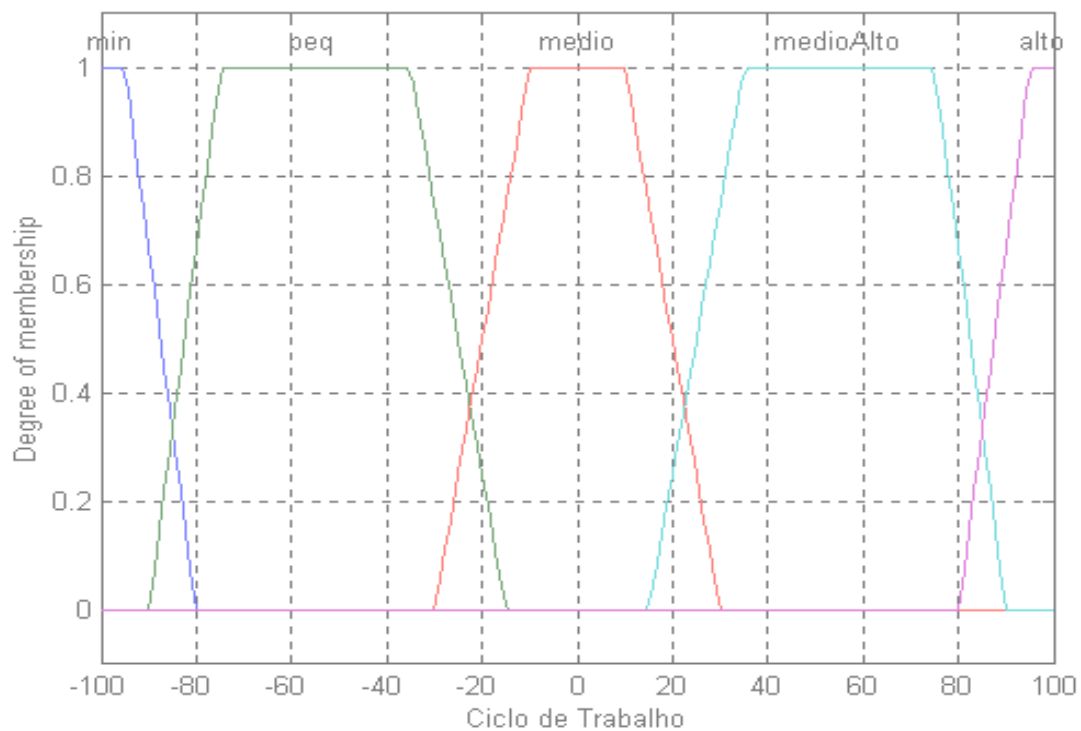


Figura 27 - Funções de pertinência da variável de "Ciclo de Trabalho"

```

f (Erro is neg) then (Ciclo de Trabalho is min) (1)
f (Erro is negMedio) then (Ciclo de Trabalho is peq) (1)
f (Erro is nulo) then (Ciclo de Trabalho is medio) (1)
f (Erro is posMedio) then (Ciclo de Trabalho is medioAlto) (1)
f (Erro is pos) then (Ciclo de Trabalho is alto) (1)

```

Figura 28 - Base de Regras implementada

Tabela 9 - Descrição do sistema de inferência *fuzzy* simulado

Características	Métodos adotados
Implicação	Mínimo
Agregação	Máximo
Defuzzificação	Média do Máximo
Método AND	Mínimo
Método OR	Máximo

O método de defuzzificação adotado consiste no cálculo da média dos valores com máximo grau de pertinência da variável de saída. Este método é escolhido por envolver pouco esforço computacional, importante fator quando se considera implementar tal algoritmo em microcontroladores, e preservar a propriedade de continuidade da saída do sistema de inferência difuso. As funções de pertinência da saída poderiam ser representadas apenas pelos valores médios dentre os valores da variável em que o grau de pertinência é máximo (100%).

4.4 Simulação de Sistemas em Malha Fechada

- **Processo com controlador PI**

O esquema de simulação adotado está esquematizado na figura 29. A entrada degrau é aplicada para os dois processos controlados e o resultado está nas figuras 30 e 31. Nestas, observa-se que o critério de tempo de subida foi atingido, com erro de regime nulo para os dois processos controlados. O valor de sobrepasso, entretanto, chegou em cerca de “10%” para os dois processos.

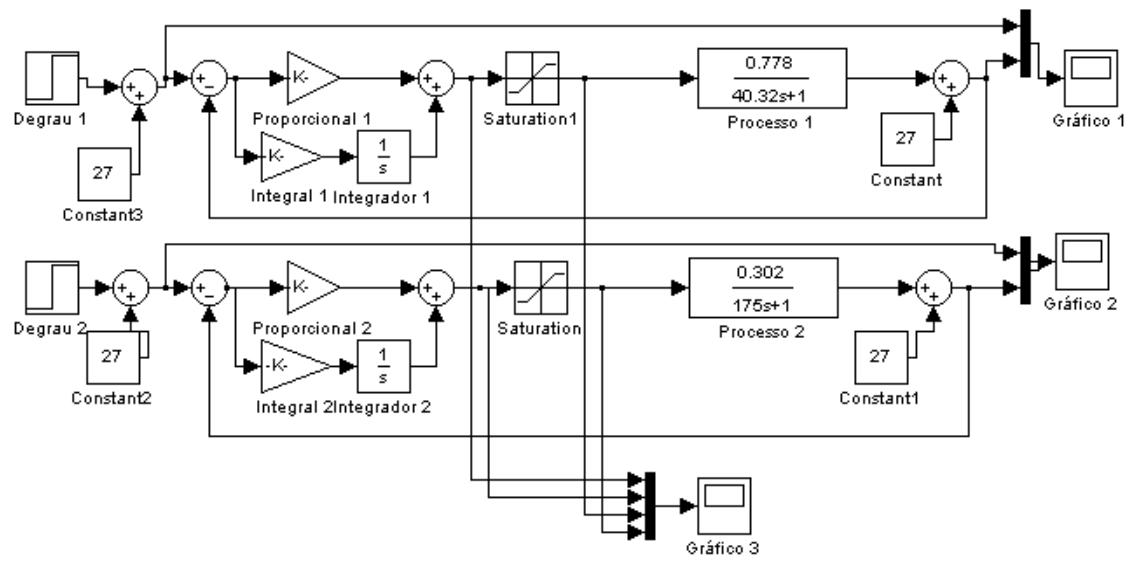


Figura 29 - Sistema simulado com controlador PI com parâmetros calculados anteriormente

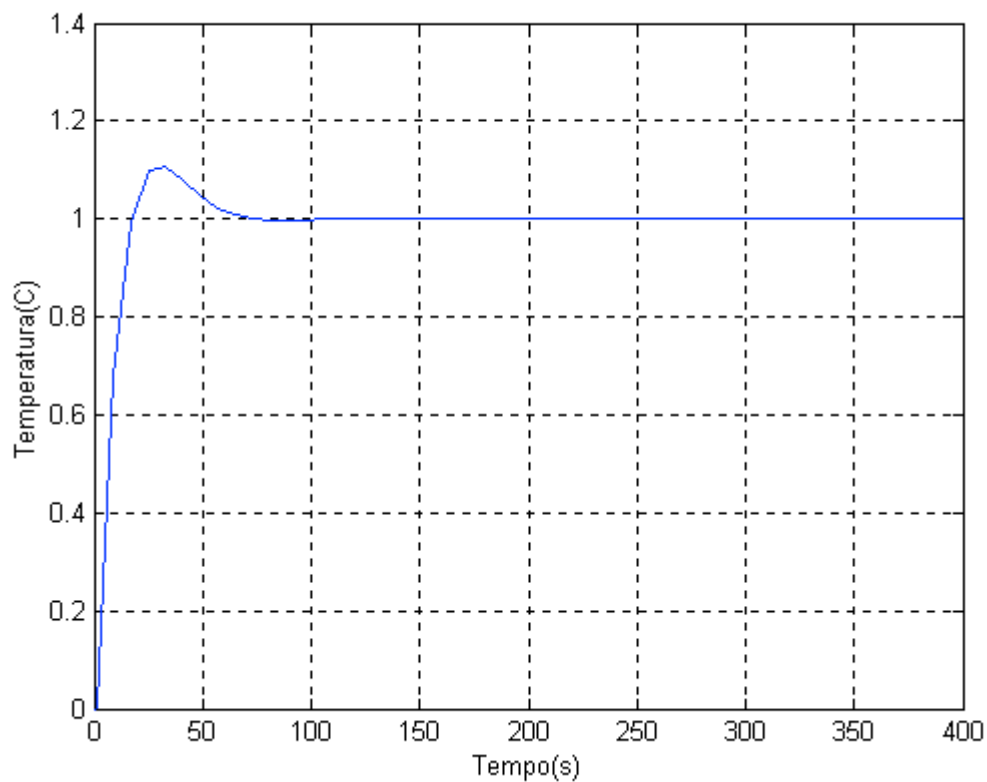


Figura 30- Resposta do processo 1,2 com controlador PI ao degrau unitário

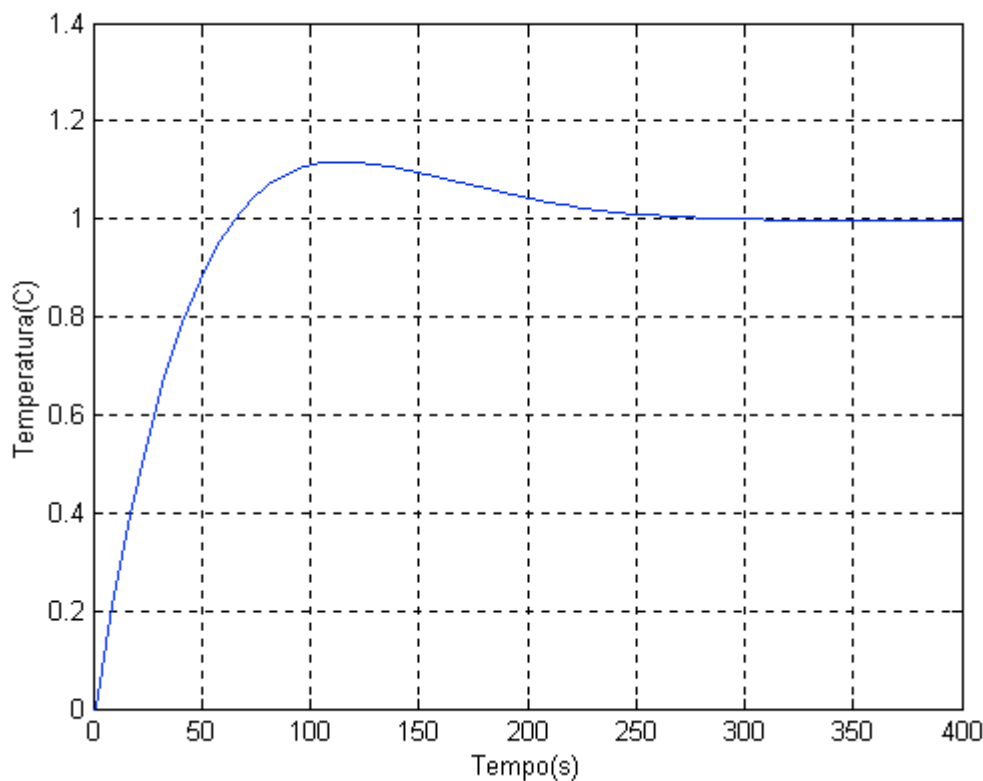


Figura 31 - Resposta do processo 2,5 com controlador PI ao degrau unitário

O desempenho dos controladores PI simulados obteve, portanto, índices satisfatórios, sendo implementados, em seguida, no algoritmo de controle do sistema térmico real objeto de estudo.

- **Processo com controlador *Fuzzy***

O esquema de simulação adotado está esquematizado na figura 32. A entrada degrau é aplicada para os dois processos controlados e o resultado simulado está nas figuras 33 e 34. As figuras 33 e 34 mostram que o controlador difuso possui tempo de subida elevado. Entretanto, o valor do sobrepasso observado é menor do que 5%. O erro de regime permanente observado, para os dois sistemas controlados, é nulo. O desempenho do controlador é adequado para os dois processos, sendo implementado no algoritmo responsável pelo controle de temperatura do sistema real e produzindo os resultados detalhados na seção 4.5.

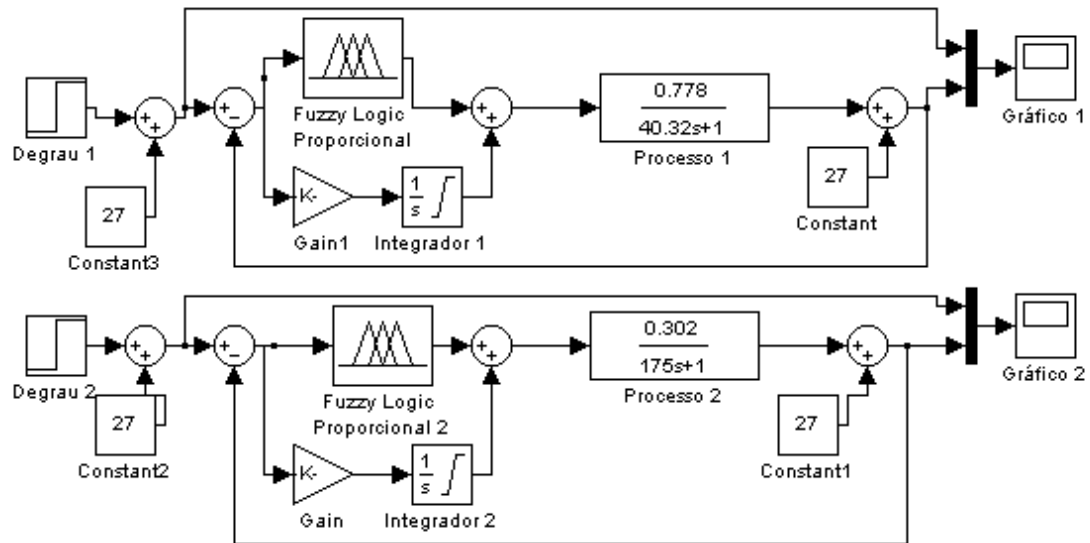


Figura 32 - Sistema simulado com controlador difuso

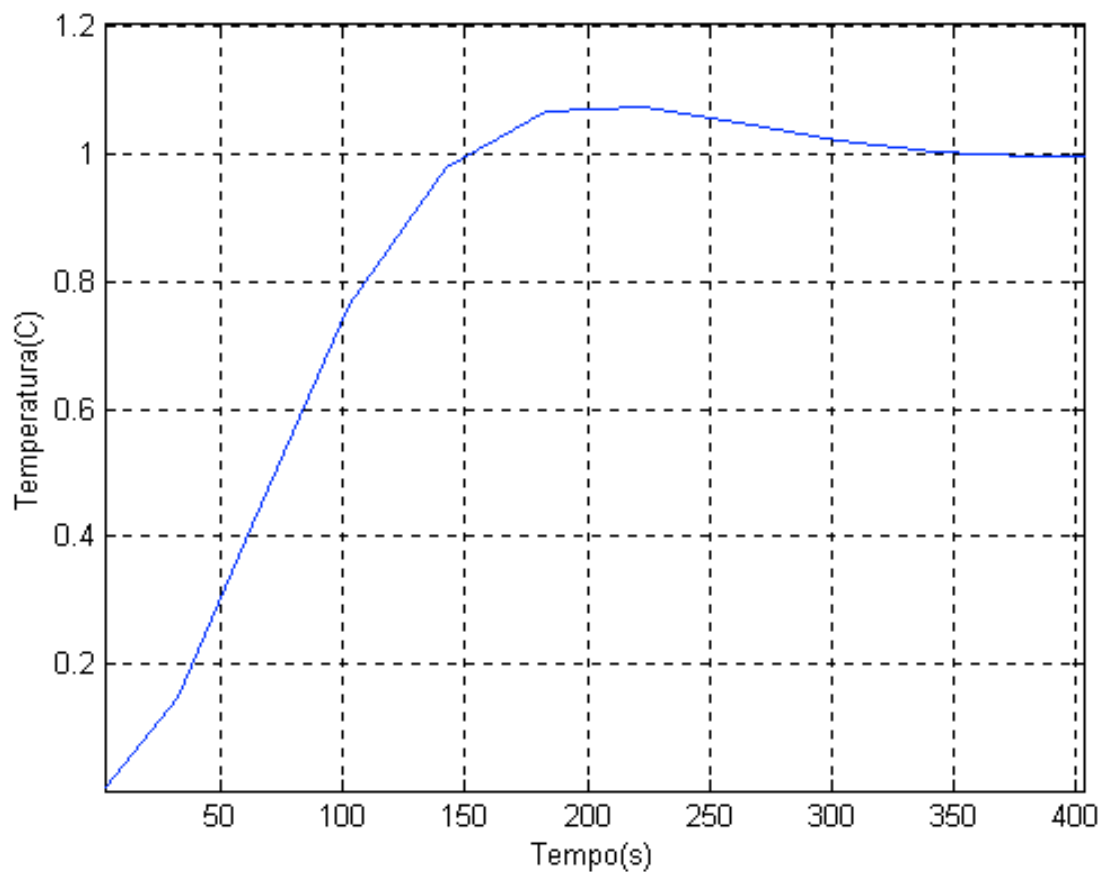


Figura 33 - Resposta do processo 1,2 com controlador proporcional *fuzzy* ao degrau unitário

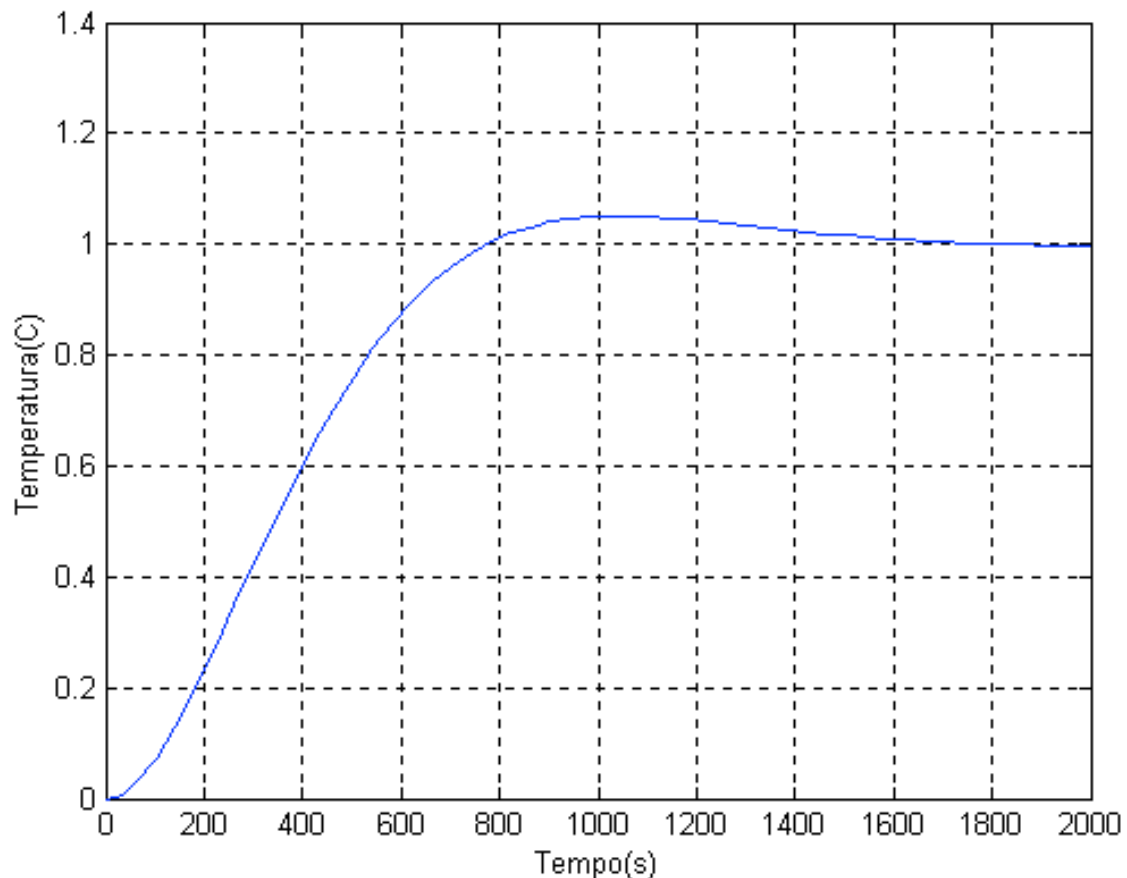


Figura 34 - Resposta do processo 2,5 com controlador proporcional *fuzzy* ao degrau unitário

4.5 Resultados Experimentais

Os sinais de teste para os processos “1,2” e “2,5” são os seguintes:

- Processo 1,2: onda de referência quadrada entre os valores de temperatura: 40-45; 45-50; 50-55. O período da onda é de 200 segundos e o nível inicial é o nível baixo.
- Processo 2,5: onda de referência quadrada entre os valores de temperatura: 29-31; 31-33; 33-35. O período da onda é de 400 segundos e o nível inicial é o nível baixo.

As figuras 35 a 40 representam os resultados obtidos pelo controlador PI para os diversos sinais de teste. As figuras 41 a 46 representam os resultados obtidos pelo controlador proporcional *fuzzy* para os mesmos sinais de teste. Com estes gráficos, torna-se possível avaliar o desempenho dos controladores em cada uma das situações e compará-los.

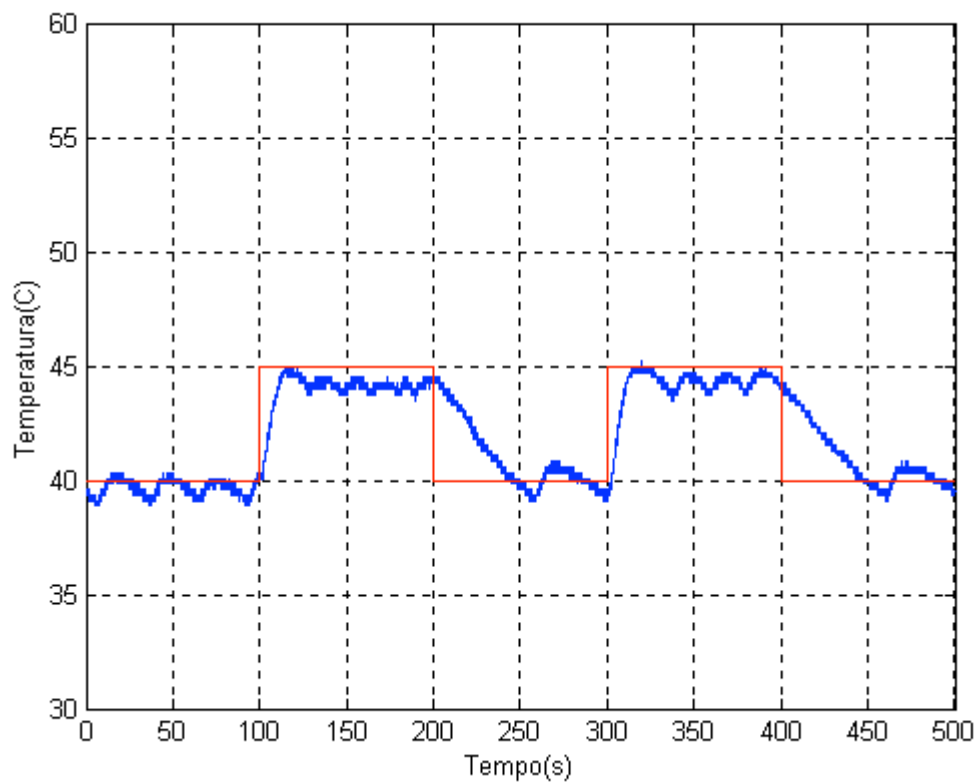


Figura 35- Resposta do processo 1,2 com controlador proporcional integral à onda quadrada 40-45

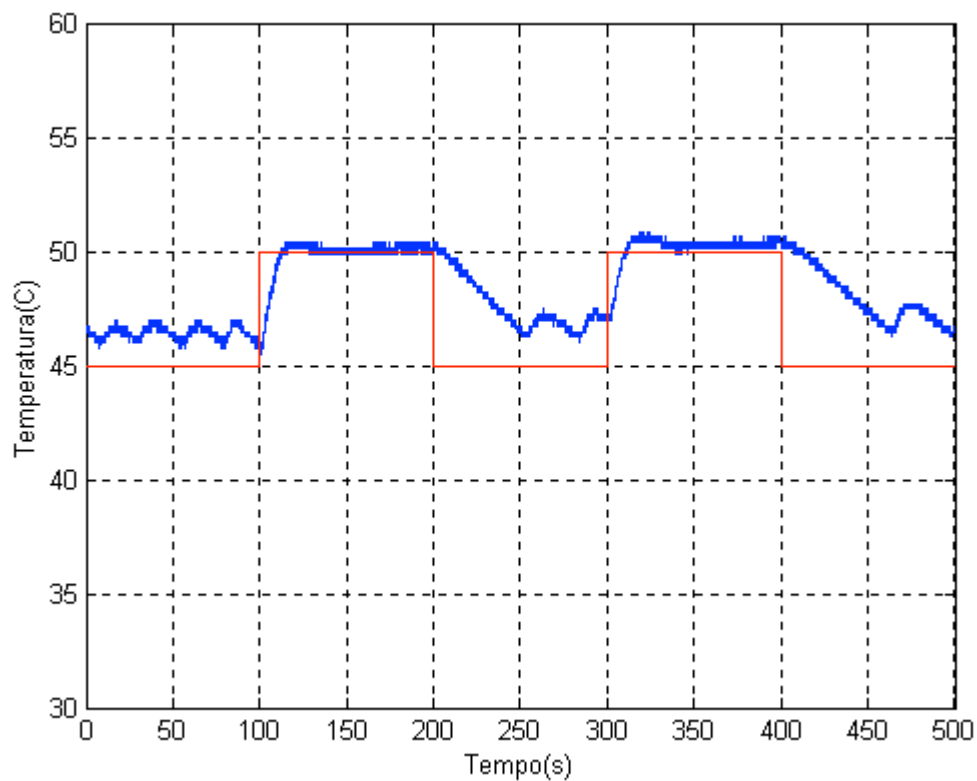


Figura 36- Resposta do processo 1,2 com controlador proporcional integral à onda quadrada 45-50

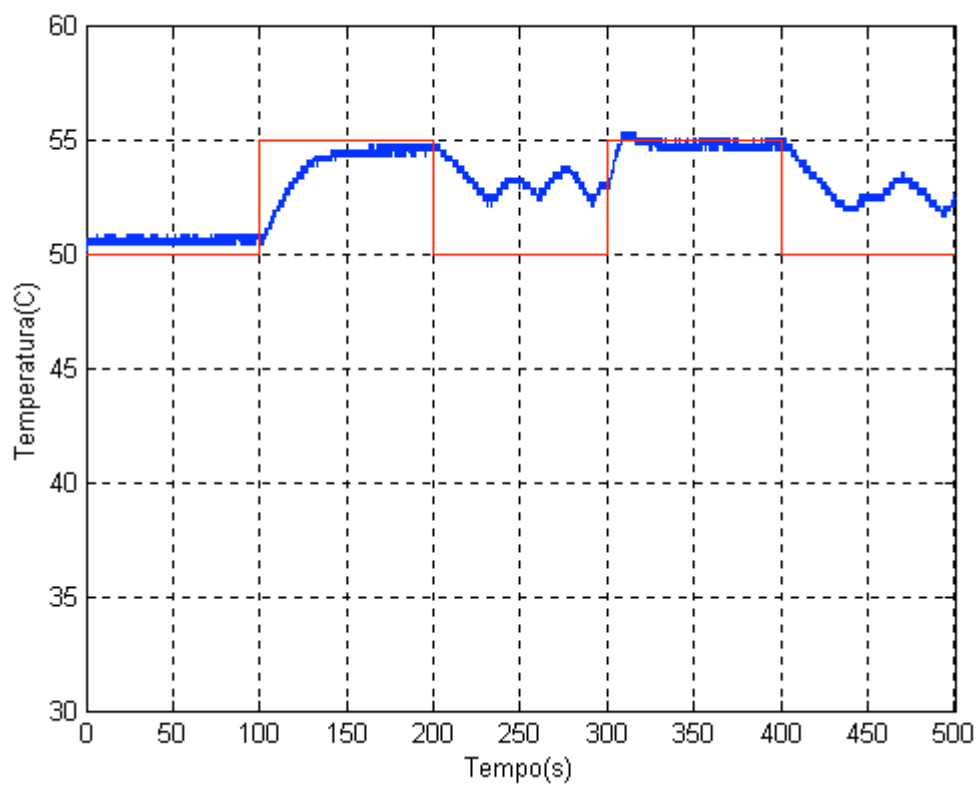


Figura 37- Resposta do processo 1,2 com controlador proporcional integral à onda quadrada 50-55

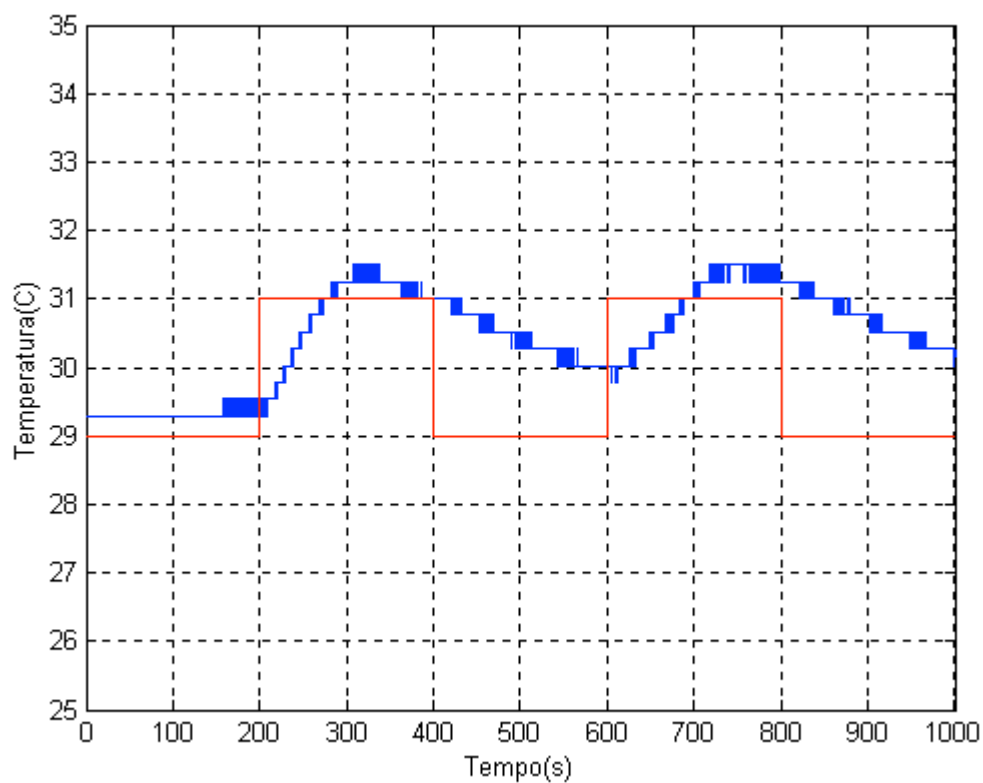


Figura 38- Resposta do processo 2,5 com controlador proporcional integral à onda quadrada 29-31

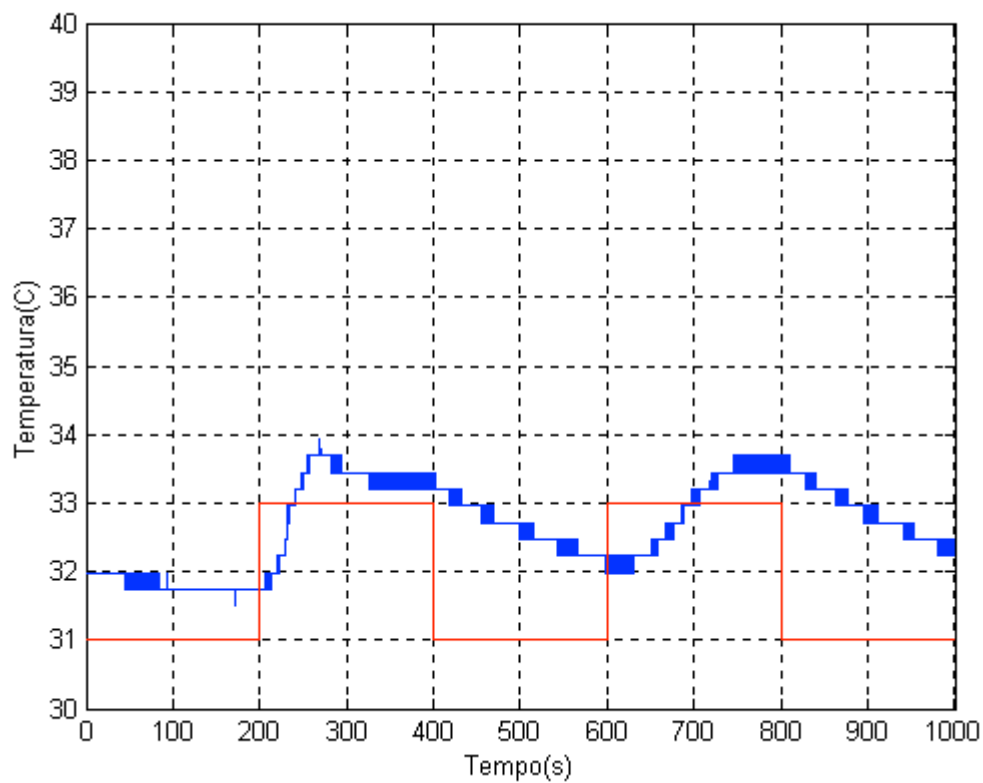


Figura 39- Resposta do processo 2,5 com controlador proporcional integral à onda quadrada 31-33

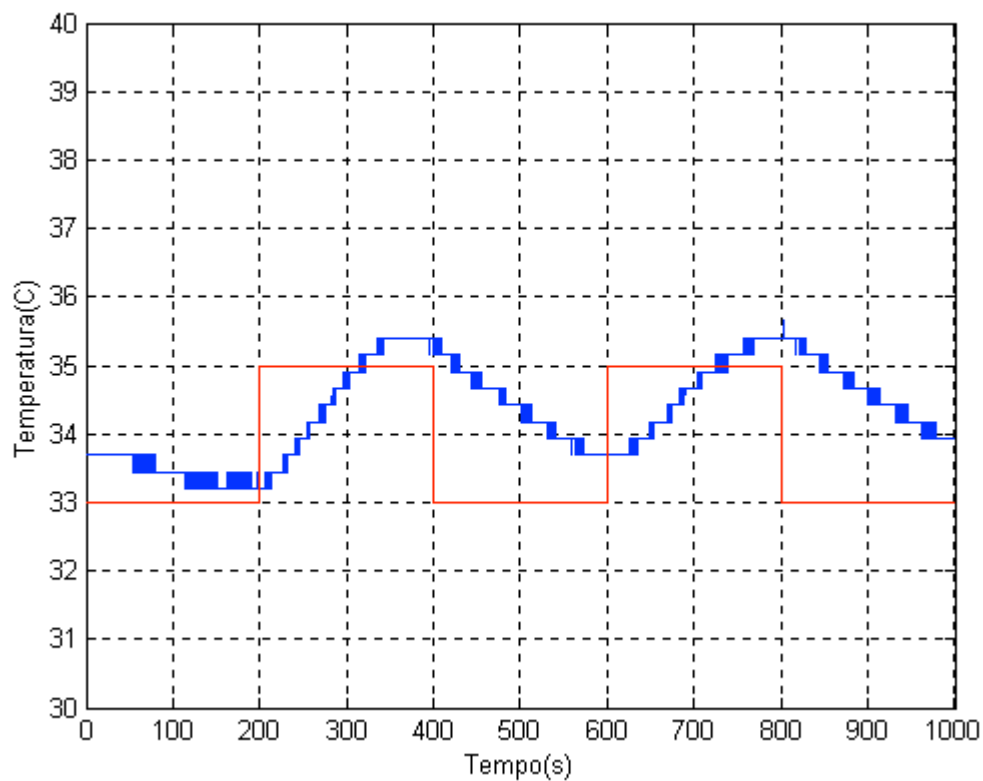


Figura 40- Resposta do processo 2,5 com controlador proporcional integral à onda quadrada 33-35

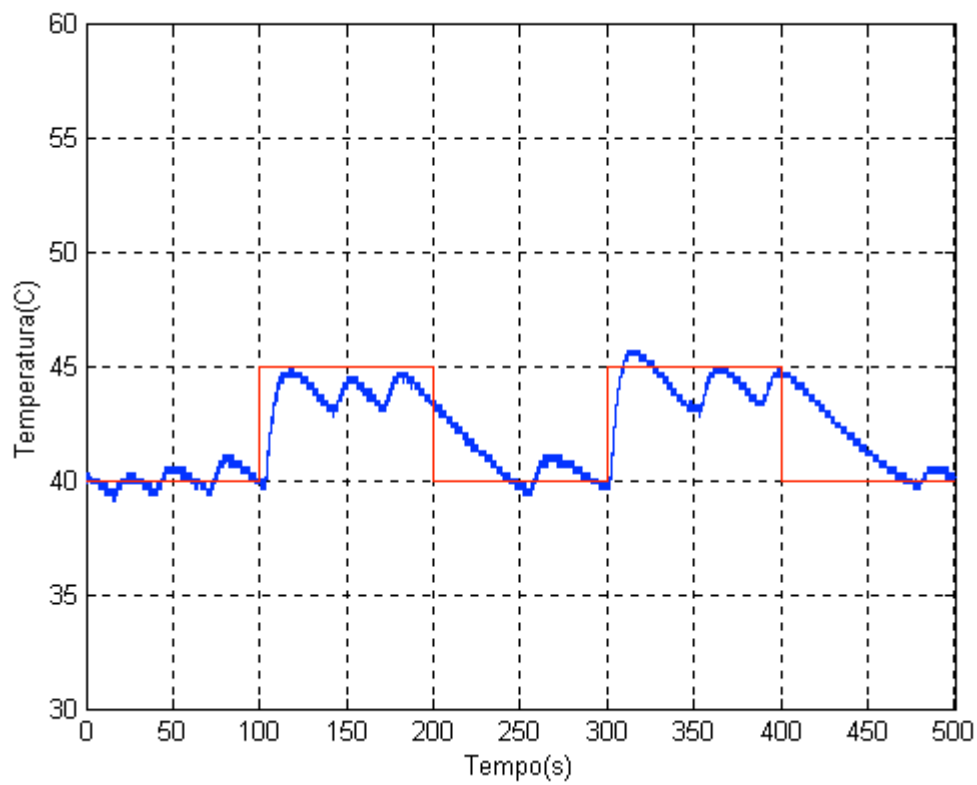


Figura 41- Resposta do processo 1,2 com controlador *fuzzy* à onda quadrada 40-45

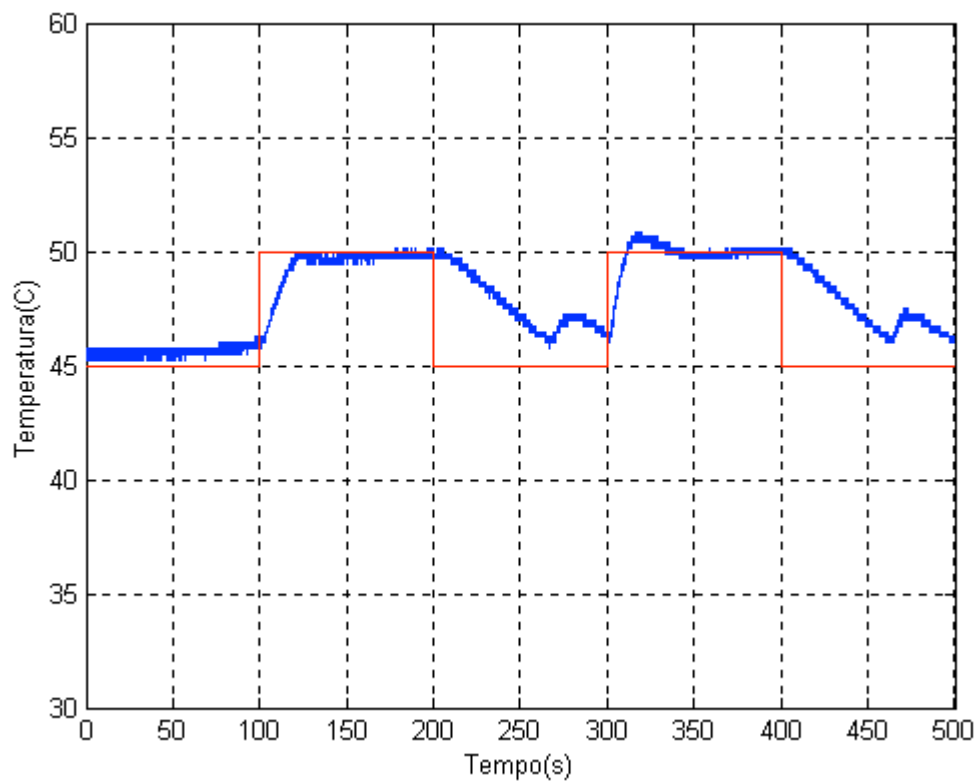


Figura 42- Resposta do processo 1,2 com controlador *fuzzy* à onda quadrada 45-50

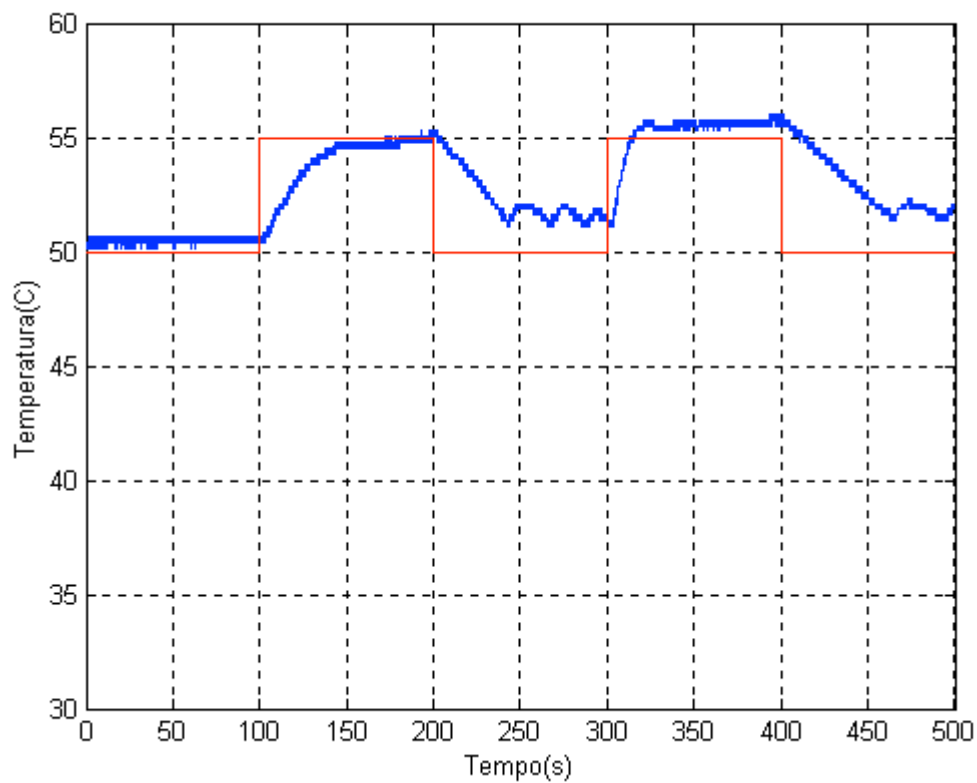


Figura 43- Resposta do processo 1,2 com controlador *fuzzy* à onda quadrada 50-55

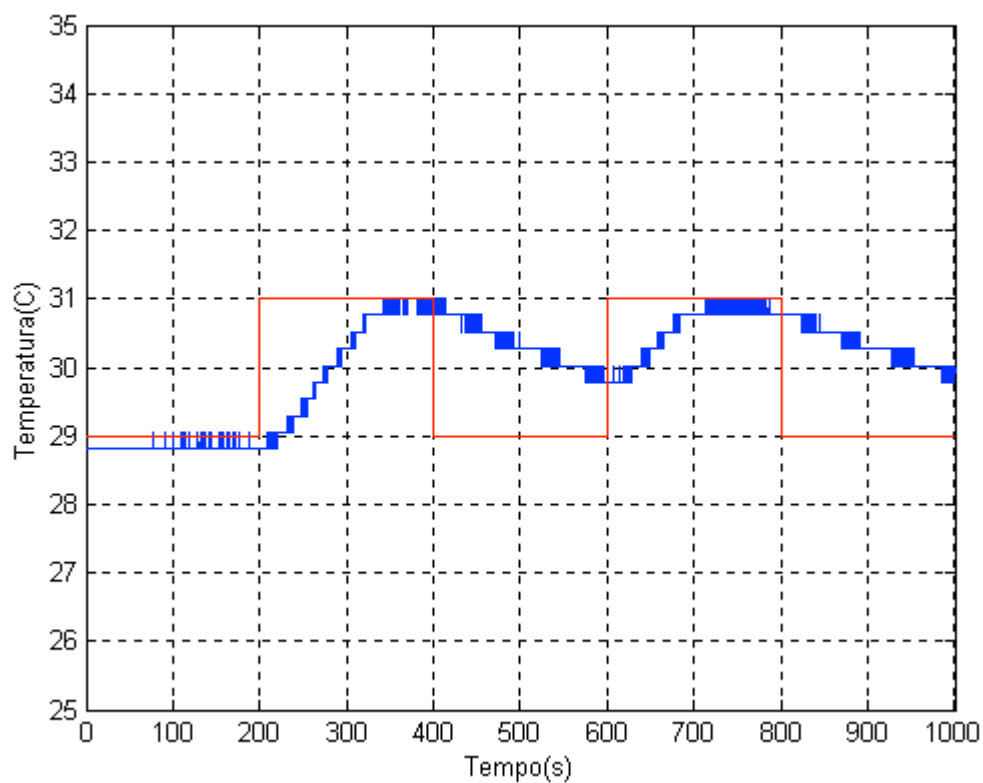


Figura 44- Resposta do processo 2,5 com controlador *fuzzy* à onda quadrada 29-31

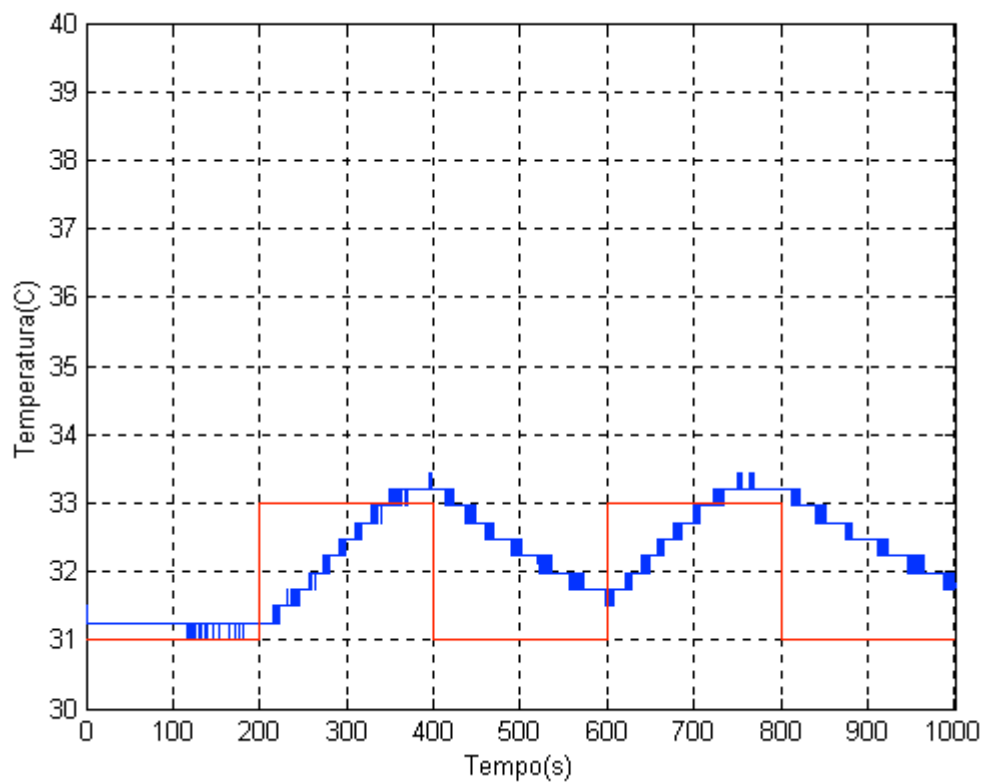


Figura 45- Resposta do processo 2,5 com controlador *fuzzy* à onda quadrada 31-33

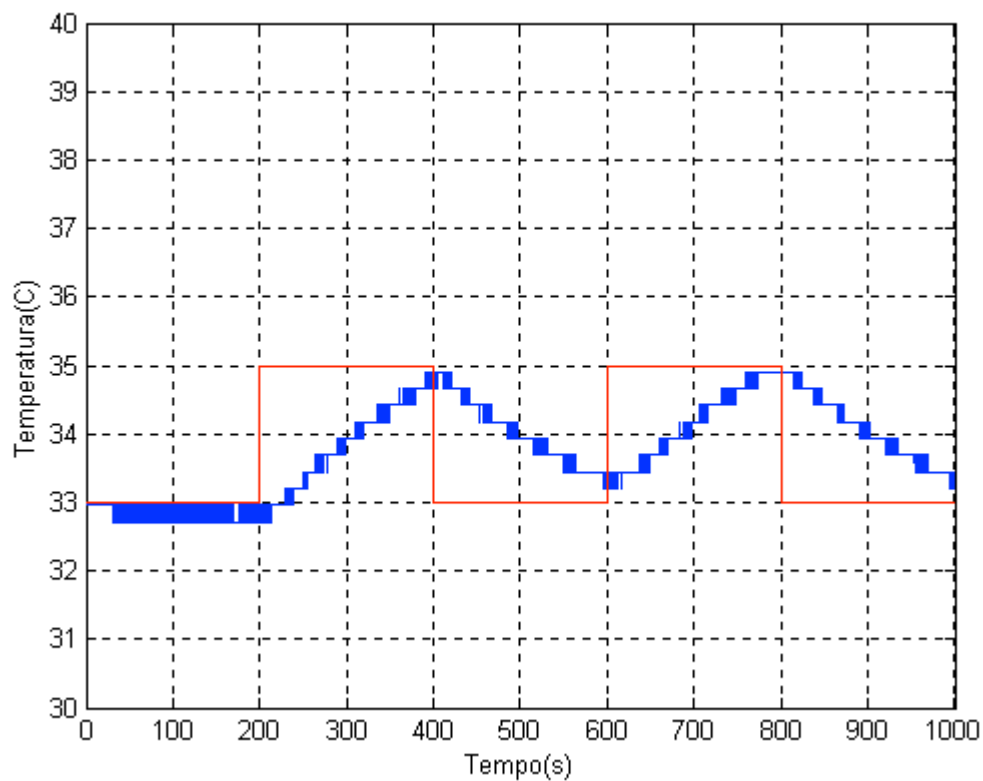


Figura 46- Resposta do processo 2,5 com controlador *fuzzy* à onda quadrada 33-35

Os gráficos 35 a 40 comprovam o projeto bem sucedido dos controladores proporcional-integral implementados via software do microcontrolador. O erro de regime nos gráficos 36 e 37 é nulo, mas nos gráficos 35, 38, 39 e 40 atingiu valor entre “0,5°C” e “1°C”. Os gráficos 35, 36 e 37 não apresentam sobrepasso, enquanto que os gráficos 38, 39 e 40 apresentaram sobrepasso entre “0,5°C” e “1°C”. Além disto, quando o sinal de referência está no nível baixo da onda quadrada, o erro não anulado em todos os gráficos exceto no gráfico 35. Este fato sugere que a onda quadrada apresentada como referência deveria ter um período maior, permitindo que a temperatura desejada fosse atingida pelos processos controlados.

Os gráficos 41 a 46 mostram que os controladores *fuzzy* implementados são eficientes com erro de regime pequeno, sendo nulo em alguns casos como nos gráficos 42, 43 e 46. O erro de regime atingido nos gráficos 41, 44 e 45 possui valor máximo de “0,5°C”. O sobrepasso é inexistente em todos os casos exceto nos gráficos 41 e 46, onde o sobrepasso atingiu o valor de “0.5°C”.

A presença de oscilação nos gráficos 35, 36, 41 e 42 deve-se ao fato que os acionamentos são motores de corrente contínua que necessitam de um certo valor mínimo de corrente de armadura para darem partida e iniciarem operação. Com isto, os controladores acionam o secador quando o erro atinge um certo valor e desligam o secador quando o erro aumenta além de um determinado valor. A resistência interna do secador, fonte de calor do acionamento, é aquecida a um pequeno valor mesmo quando o motor de corrente continua está parado.

Os pequenos saltos de valores de temperatura observados nos gráficos 38, 39, 40, 44, 45 e 46 ocorrem devido à precisão do sensor. A não-linearidade deste elemento pode atingir valor de até “0.25°C” e o erro máximo de medida é de “0.5°C”, como detalhado na seção 3.4.1. O conversor analógico digital do microcontrolador detecta variações de temperatura de até “0.5°C”, sendo necessária uma variação de temperatura de valor maior ou igual a “0.5°C” para que os controladores tenham o sinal de controle modificado.

Pela análise dos parâmetros de desempenho adotados como referência, pode-se afirmar que o controlador nebuloso apresentou desempenho superior ao desempenho do controlador proporcional integral. Em alguns casos, o desempenho é praticamente o mesmo para os dois controladores, mas nos outros o desempenho do controlador *fuzzy* justifica seu emprego com menores taxas de sobrepasso e menores erros de regime.

5. Conclusões

Os resultados das diversas simulações permitiram a avaliação prévia do comportamento do sistema quando operando com os controladores projetados. A ferramenta de simulação usada, o ambiente Matlab-Simulink®, foi muito útil para o projeto dos controladores e avaliação do desempenho dos mesmos.

O projeto de controladores *Fuzzy* foi implementado com a “*Fuzzy Logic Toolbox*”, parte integrante do *software* MATLAB, e permitiu a avaliação do desempenho do controlador com diversas configurações possíveis. Isto consiste no uso de diferentes: métodos de defuzzificação, número de regras, número de funções de pertinência, entre outros.

O sensor usado é de simples operação e custo baixo, justificando seu uso dentro do sistema implementado. A imprecisão máxima do sensor é adequada à aplicação objeto de estudo.

O acionamento escolhido, um secador de cabelo com uma resistência e um motor de corrente contínua, apresentou restrições quando o sistema foi submetido a um degrau negativo, fazendo com que o sistema funcionasse como um controle do tipo liga-desliga em algumas situações. A simplicidade deste acionamento, entretanto, foi fator decisivo para o seu emprego.

O microcontrolador adotado, o PIC16F877 apresentou limitações de memória de dados, apenas 256 bytes. Para o controle proporcional-integral, esta quantidade de memória foi bastante, enquanto que para a implementação do controlador *fuzzy* a quantidade de memória foi fator restritivo, exigindo: base de regras fixa, número de funções de pertinência para a variável de entrada fixa, método de defuzzificação simples e funções de pertinência da variável de saída do tipo *singleton*. A presença de dois canais com sinal modulado por largura de pulso e oito canais de conversão analógico-digital foi fator decisivo na sua escolha e adoção, viabilizando a implementação das estratégias de controle PI e *fuzzy*. A velocidade de operação do microcontrolador, 0,222μs/instrução, possibilitou uma alta taxa de amostragem, permitindo implementar os algoritmos de controle na forma “quasi-contínua”.

A programação em baixo nível feita em linguagem C facilitou a construção e teste rápido dos algoritmos que implementam as estratégias de controle estudadas. As

bibliotecas disponíveis também ajudaram no acesso aos recursos do microcontrolador através de simples instruções com leitura e escrita em variáveis mnemônicas. Visando permitir a reutilização do código produzido, diversas bibliotecas foram desenvolvidas envolvendo os algoritmos para conversão analógico digital, escrita no LCD e cálculo do valor de saída do sistema de inferência *fuzzy* projetado para controle dos processos da maquete analisada. A presença anterior de código desenvolvido para uso dos microcontroladores da família PIC e a larga presença de projetos implementados com o uso destes microcontroladores na “internet” ajudaram na produção dos algoritmos citados e na utilização das interfaces disponibilizadas por estes dispositivos.

O software “LabVIEWTM” de programação gráfica e a placa de aquisição de dados PCI-6024E usada foram muito úteis para a produção dos gráficos de resposta das figuras 35 à 46, permitindo a análise e comparação do desempenho dos controladores implementados. A placa de aquisição de dados também possui diversas interfaces de entrada e saída que poderiam ser usadas em outros projetos para controle de processos diretamente através de computador pessoal, usando-se apenas controles gráficos do software “LabVIEWTM” para definição de sinais de referência e parâmetros de controladores desejados.

A maquete usada como sistema para implementação dos algoritmos de controle e estudo comparativo do desempenho dos mesmos foi bastante útil. Algumas não-linearidades que não foram consideradas na modelagem dos processos, como a interação com o meio ambiente e a transferência de calor envolvida pelas paredes da maquete representantes das divisórias dentro do escritório, não apresentaram influência que comprometessem os resultados alcançados. Esta maquete poderia ser usada em projetos futuros envolvendo controle de temperatura em ambientes, podendo até mesmo ser controlada remotamente com algoritmos de controle fornecidos por usuários conectados remotamente.

A técnica de inteligência artificial “Lógica *Fuzzy*” mostra-se bastante útil para controle de plantas comprovadamente não-lineares. A observação dos gráficos de resposta, tanto os simulados quanto os reais, mostram sua adequação para o tratamento de problemas não-lineares. A sua associação com outras técnicas de controle inteligente, como Redes Neurais (sistema *Neurofuzzy*), poderia apresentar ganhos significativos em termos de desempenho do sistema em malha fechada.

Os elementos utilizados para controle dos processos dentro da maquete estudada poderiam ser usados para controle de temperatura usando como acionamento o aparelho

de ar-condicionado. Alterações substanciais seriam necessárias no projeto do driver de potência, já que este aparelho possui potência muito maior do que a potência do secador de cabelo usado com “1kW”. Para construção do sistema proposto na seção 1.5, também seria necessário o desenvolvimento de uma interface de programação de mais alto nível, em computador pessoal, dos microcontroladores. Esta programação exigiria do operador apenas a definição da base de regras do sistema nebuloso. A geração do algoritmo de controle e programação do microcontrolador seria feita automaticamente através de linguagens de programação de mais alto nível, como Java e C++.

Feitas estas modificações, o sistema proposto seria implementado e testado em laboratório, consistindo em um primeiro passo em direção à transformação deste projeto em um produto disponibilizado no mercado com custo baixo, resultando em uma baixa taxa de retorno para os usuários que investissem no mesmo. A programação e operação seriam simples, tendo a maioria das tarefas automatizada e exigindo dos operadores apenas a definição das regras lingüísticas.

6. Bibliografia

- [1] SHAW, I. S., SIMÕES, M. G., “Controle e Modelagem *Fuzzy*”, Edgard Blücher, São Paulo, 1999.
- [2] HINES, J. W., “*Fuzzy and Neural Approaches in Engineering*”, John Wiley & Sons, New York, 1997.
- [3] GANG, J. S. R., GULLEY, N., “*Fuzzy Logic Toolbox – For use with Matlab*”.
- [4] OGATA, K., “*Modern Control Engineering*”, Prentice Hall, Londres, v.3, 1997.
- [5] KASABOV, N. K., “*Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*”, MIT Press, 1996.
- [6] Harris, C. J., Moore, C. G., Brown, M., “*Intelligent Control – Aspects of Fuzzy Logic and Neural Nets*”, World Scientific Publishing Co., Singapore, v.2, 380p, 1993.
- [7] Bauchspiess, A., “*Servocontrole preditivo de robôs guiados por sensores*”, *Universität Erlangen-Nürnberg*, 1995.
- [8] Kovacic, M. Z., Balenovic, S. B., “*Sensitivity-Based Self-Learning Fuzzy Logic Control for a Servo System*”, *IEEE Control Systems*, vol. 18, número 3, p.p. 41-52, (1998).
- [9] Abramovitch, D. Y., Bushnell, L. G., “*Report on the Fuzzy versus Conventional Control Debate*”, *IEEE Control Systems*, vol. 19, número 3, p.p. 88-91.
- [10] Normey-Rico, J. E., “*Projeto de Controladores PID industriais*”, Santa Catarina, 2001.
- [11] Souza, D. J., “*Desbravando o PIC*”, Editora Érica Ltda, 2ª ed., São Paulo, 200p, 2000.
- [12] Jacquot, R. G., “*Modern Digital Control Systems*”, Marcel Dekker Inc., 2ª ed., New York, 1994.
- [13] Home page da empresa Microchip™, www.microchip.com.
- [14] Home page da empresa Cyronix™, www.cyronix.com.
- [15] Home page da empresa HCS™, www.hcs1.com/hcs_controls_directory.htm.
- [16] Home page da empresa OmRON™, www.omron.com.br.
- [17] Home page de Johann Aichinger, <http://jaichi.virtualave.net/electr-e.htm>
- [18] Data-sheet dos seguintes componentes: PIC16F877, LM35DZ, BC548, C3831, D1N4004.