Fernando Ranieri

Sistema Supervisório de Parâmetros de Máquinas Elétricas via TCP/IP e Painel Eletrônico de Mensagens

Dissertação apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, sendo parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Ivan Nunes da Silva

São Carlos

2007

FOLHA DE JULGAMENTO

Candidato: Engenheiro **FERNANDO RANIERI**

Dissertação defendida e julgada em 17/08/2007 perante a Comissão Julgadora:

Prof. Associado **IVAN NUNES DA SILVA (Orientador)**
(Escola de Engenharia de São Carlos/USP)

APROVADO

Prof. Dr. **JOSE ÂNGELO CAGNON**
(Universidade Estadual Paulista "Julio de Mesquita Filho"/UNESP-Campus de Bauru)

APROVADO

Prof.^a Dr.^a **MARIA STELA VELUDO DE PAIVA**
(Escola de Engenharia de São Carlos/USP)

APROVADO

Prof. Associado **GERALDO ROBERTO MARTINS DA COSTA**
Coordenador do Programa de Pós-Graduação em
em Engenharia Elétrica e
Presidente da Comissão de Pós-Graduação da EESC

AGRADECIMENTOS

A realização deste trabalho só se tornou possível graças à colaboração direta e indireta de muitas pessoas que cruzaram minha vida e principalmente a Deus.

Primeiramente à minha família, que sempre me apoiou e me ajudou nas horas difíceis, bem como demais familiares.

Ao meu irmão Claudio, que sempre me ajudou nos meus projetos e que sempre tinha soluções para resolver meus problemas.

Meus pais Rubens e Mariza que sempre me ajudaram quando precisei, e que até hoje me ligam para saber se estou me alimentando direito.

A minha namorada Suelen, minha companheira, que me suportou todo esse tempo, e que sempre me apoiou nas horas em que precisei.

Ao meu orientador, o Prof. Ivan Nunes da Silva, grande amigo, que sempre me incentivou, me orientou e dedicou seu precioso tempo comigo.

Professor Paulo R. Veronese, e demais professores da EESC que sempre me ajudaram com informações que tornaram esse projeto possível.

Aos colegas do LAIPS, Spatti, Suetake, Rodrigo, Cristiano e Sérgio pela disposição em ajudar.

Aos meus principais amigos da graduação, Leandro Azeka, João Paulo da Silva, Fábio Rosseto e Auricir Augusto.

Aos meus amigos da Escola Técnica Federal de São Paulo.

À FAPESP pelo suporte financeiro, sem a qual esse projeto não poderia ser realizado.

E finalmente, aos meus grandes amigos da UNESP de Bauru, Marcelo Cavachioli e Gustavo Galli, que sempre me ajudaram e me incentivaram a chegar o mais longe possível.

RESUMO

RANIERI, F. (2007). Sistema Supervisório de Parâmetros de Máquinas Elétricas via TCP/IP e Painel Eletrônico de Mensagens – Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, 2007.

Com o advento da *internet*, a comunicação entre sistemas remotos foi amplamente facilitada, tornando viável a aplicação de sistemas supervisórios através dela, principalmente via protocolo TCP/IP. Embora esses sistemas já sejam empregados na indústria, principalmente utilizando como *driver* os controladores lógicos programáveis (CLP's), a implementação de qualquer sistema mais simples de supervisionamento teria um custo elevado, devido tanto ao preço do *hardware* quanto do *software* envolvido. Além disso, os programas supervisórios existentes fornecem um número limitado de recursos de visualização, que se restringem principalmente ao *software*. Assim, o que se propõe nesse trabalho é um sistema supervisório de baixo custo, via TCP/IP, aplicado em máquinas elétricas, mas que pode ser generalizado na supervisão de outros tipos de sistemas, onde o monitoramento remoto é necessário. Além da exibição convencional dos parâmetros supervisionados através de gráficos e *labels*, o sistema permite também que esses sejam exibidos através de painéis eletrônicos, fornecendo assim, uma nova interface visual ao usuário com o intuito de facilitar sua supervisão.

Palavras chave: Sistema supervisório via TCP/IP, máquinas elétricas, painel eletrônico monocromático, painel eletrônico tricolor, protocolo TCP/IP.

ABSTRACT

RANIERI, F. (2007). Supervisory System of Electric Machine's Parameter via TCP/IP and Message Displays.

The advent of Internet simplified the communication between remote systems, allowing the application of supervisory systems through the Internet, mainly using the TCP/IP protocol. Although these systems have already been employed in industry, using the Programmer Logic Controller (PLC's) as driver, the implementation of any simple supervisory system could have an expensive cost, due to the prices of hardware and software employed. Moreover, the existing supervisory systems provide limited visual resources, restricted to the used software. So, this work presents a low cost supervisory system, via TCP/IP, applied to electric machines but it can be generalized in the supervision of different types of systems, where the remote monitoring is necessary. Besides the conventional displaying of the monitored parameters through graphics and labels, the developed system allows the exhibition of the supervised parameters using Electronic Panels, providing an alternative visual interface to the user so that their supervision becomes easier.

Keywords: Supervisory system via TCP/IP, electric machines, monochromatic message display, tricolor message display, TCP/IP protocol.

SUMÁRIO

AGRADECIMENTOS.....	III
RESUMO.....	V
ABSTRACT.....	VII
LISTA DE FIGURAS.....	XI
LISTA DE TABELAS.....	XV
LISTA DE ABREVIATURAS E SIGLAS.....	XVII
1 INTRODUÇÃO.....	1
1.1 <i>Motivação e Relevância do Trabalho.....</i>	<i>1</i>
1.2 <i>Objetivo e Justificativa da Dissertação.....</i>	<i>2</i>
1.3 <i>Organização da Dissertação.....</i>	<i>3</i>
2 ESTADO DA ARTE E ASPECTOS DE SISTEMAS SUPERVISÓRIOS.....	5
2.1 <i>Introdução.....</i>	<i>5</i>
2.2 <i>Tipos de Sistemas Supervisórios.....</i>	<i>8</i>
2.3 <i>Composição Estrutural dos Sistemas Supervisórios.....</i>	<i>9</i>
2.4 <i>Sistemas Supervisórios via TCP/IP.....</i>	<i>12</i>
3 ESTRUTURA DO SISTEMA SUPERVISÓRIO DESENVOLVIDO.....	21
3.1 <i>Introdução.....</i>	<i>21</i>
3.2 <i>Visão Geral do Sistema Supervisório Desenvolvido.....</i>	<i>22</i>
3.3 <i>Sistema de Aquisição de Dados.....</i>	<i>25</i>
3.3.1 <i>Aquisição da Temperatura do Motor e Temperatura Ambiente.....</i>	<i>37</i>
3.3.2 <i>Aquisição da Corrente e Tensão de Armadura do Motor.....</i>	<i>41</i>
3.3.3 <i>Aquisição da Tensão da Rede Elétrica.....</i>	<i>43</i>
3.3.4 <i>Aquisição da Frequência da Rede Elétrica.....</i>	<i>47</i>
3.3.5 <i>Aquisição da Velocidade de Rotação do Eixo do Motor.....</i>	<i>50</i>
3.3.6 <i>Aquisição dos Demais Parâmetros DC.....</i>	<i>57</i>
3.3.7 <i>Circuito Final de Aquisição.....</i>	<i>57</i>
3.4 <i>Sistema de Controle.....</i>	<i>62</i>
3.4.1 <i>Controle da Velocidade de Rotação do Eixo do Motor DC.....</i>	<i>62</i>
3.4.2 <i>Circuito do Sistema de Ajuste de Tensão.....</i>	<i>71</i>
3.5 <i>Sistema de Transmissão e Recepção de Dados via TCP/IP.....</i>	<i>73</i>

3.5.1 Programa do Servidor	73
3.5.2 Programa do Cliente.....	79
3.6 Sistema de Supervisão de Parâmetros	81
3.7 Sistema de Visualização de Parâmetros via Painel Eletrônico de Mensagens ..	84
3.7.1 Painel Eletrônico Monocromático	89
3.7.2 Painel Eletrônico Tricolor.....	97
3.8 Interface de Entrada via Controle Remoto Infravermelho.....	107
4 RESULTADOS DA APLICAÇÃO DO SISTEMA SUPERVISÓRIO	
 DESENVOLVIDO	115
4.1 Introdução	115
4.2 Módulos de Interface Visual do Sistema Supervisório.....	116
4.2.1 Visualização do Sistema Supervisionado através de uma Câmera Digital.	116
4.2.2 Visualização do Sistema Supervisionado através de Labels.....	118
4.2.3 Visualização dos Parâmetros Supervisionados através de Gráficos.....	120
4.2.4 Visualização dos Parâmetros Supervisionados através de Painéis	
Eletrônicos	121
4.3 Resultados do Sistema de Aquisição de Dados.....	123
4.4 Resultados do Sistema de Controle da Velocidade de Rotação do Eixo do	
Motor	125
4.5 Resultados do Sistema de Transmissão e Recepção de Dados via TCP/IP	129
4.6 Resultados do Sistema de Supervisão de Parâmetros	140
4.7 Resultados do Sistema de Visualização de Parâmetros via Painel Eletrônico	
de Mensagens	143
4.8 Resultados da Interface de Entrada via Controle Remoto Infravermelho.....	148
5 CONCLUSÕES	153
BIBLIOGRAFIA	155

LISTA DE FIGURAS

Figura 2.1. Composição estrutural de um sistema supervisório do tipo Monoposto.	10
Figura 2.2. Composição estrutural de um sistema supervisório do tipo Servidor/Cliente.	11
Figura 2.3. Composição estrutural de um sistema supervisório do tipo Servidor/Multi-Clientes.	11
Figura 2.4. Composição estrutural de um sistema supervisório do tipo Múltiplos Servidores/Clientes.	12
Figura 3.1. Diagrama em blocos do sistema supervisório desenvolvido.	22
Figura 3.2. Ilustração da conexão entre motor, sistema de aquisição de dados, sistema de ajuste de tensão do motor e computador servidor.	28
Figura 3.3. Representação de um circuito <i>multiplex</i> de 8 canais de entrada.	29
Figura 3.4. Sistema de aquisição – entradas analógicas.	31
Figura 3.5. Sistema de aquisição – entrada pulsada.	36
Figura 3.6. Curva característica do NTC-10k.	37
Figura 3.7. Circuito para aquisição das temperaturas do motor e ambiente.	39
Figura 3.8. Circuito para aquisição da corrente e tensão do motor.	41
Figura 3.9. Circuito para aquisição da tensão da rede elétrica.	44
Figura 3.10. Formas de onda no ponto A com a chave S aberta e fechada.	44
Figura 3.11. Circuito para obtenção da frequência da rede.	48
Figura 3.12. Inversor lógico.	49
Figura 3.13. Circuito final para obtenção da frequência da rede elétrica.	50
Figura 3.14. Representação da hélice conectada ao eixo do motor.	51
Figura 3.15. Circuito projetado para aquisição da velocidade de rotação do eixo do motor.	52
Figura 3.16. Acoplamento mecânico entre o motor e o gerador DC.	54
Figura 3.17. Circuito para aquisição da velocidade de rotação do eixo do gerador.	56
Figura 3.18. Circuito de aquisição final projetado.	59

Figura 3.19. Oscilador externo projetado.....	60
Figura 3.20. Esquema de ligação do <i>Driver/Receiver</i> MAX232.	61
Figura 3.21. Diagrama eletromecânico do motor DC.....	65
Figura 3.22. Método da última sensibilidade.....	69
Figura 3.23. Circuito do sistema de ajuste de tensão.	72
Figura 3.24. Diagrama das conexões de 3 clientes ao servidor.	78
Figura 3.25. Tipos de faixas de supervisão: (a) faixa de supervisão máxima; (b) faixa de supervisão mínima; (c) faixa de supervisão intermediária.	82
Figura 3.26. Matriz de <i>led</i> catodo comum C-5570X.....	90
Figura 3.27. Circuito de acionamento para 3 matrizes de <i>led</i>	91
Figura 3.28. Seleção e escrita em uma das colunas do painel monocromático.	93
Figura 3.29. <i>Driver</i> de acionamento das linhas do painel eletrônico monocromático.....	95
Figura 3.30. Circuito de acionamento das colunas do painel eletrônico monocromático.....	96
Figura 3.31. Diagrama esquemático da matriz colorida XMURG60C8.....	97
Figura 3.32. Diagrama esquemático do acionamento das colunas de uma matriz de <i>led</i> colorida.	98
Figura 3.33. Multiplexação da porta P1 do AT89S52.	100
Figura 3.34. Diagrama simplificado de acionamento de uma única matriz de <i>led</i> colorida.....	101
Figura 3.35. Representação do efeito de rolagem de cima para baixo de uma dada mensagem.....	106
Figura 3.36. Organização dos <i>bits</i> transmitidos pelo controle infravermelho da Sanyo.	109
Figura 3.37. Diagrama de tempo para os dados enviados pelo LC7461M.	110
Figura 3.38. Forma de onda da portadora.	110
Figura 3.39. Módulo receptor de infravermelho TSOP1238.....	111
Figura 3.40. Circuito receptor de infravermelho.....	111

Figura 4.1. Visualização do sistema através de uma câmera digital: (a) Imagem do sistema vista pelo programa do servidor; (b) Imagem do sistema vista pelo programa do cliente.	117
Figura 4.2. Visualização dos parâmetros através de <i>labels</i> : (a) Programa do servidor; (b) Programa do cliente.	119
Figura 4.3. Visualização dos parâmetros supervisionados através de gráficos: (a) Programa do servidor; (b) Programa do cliente.	121
Figura 4.4. Painéis eletrônicos projetados para exibição de dados e informações do sistema supervisionado.	122
Figura 4.5. Aquisição dos parâmetros supervisionados pelo programa do servidor.	124
Figura 4.6. Exibição dos dados coletados através de <i>labels</i>	125
Figura 4.7. Resposta de ensaio ao degrau.	126
Figura 4.8. Resposta ao degrau utilizando o PID projetado.	127
Figura 4.9. Resposta ao degrau modificando os coeficientes do PID projetado.	128
Figura 4.10. <i>Logs</i> gerados no programa do cliente ao receber dados e imagens do servidor.	130
Figura 4.11. Configuração da ação de controle pelo programa do cliente.	132
Figura 4.12. Configuração da ação de controle pelo programa do servidor.	133
Figura 4.13. Implementação do <i>chat</i> no programa supervisorio: (a) <i>Chat</i> no programa do servidor; (b) <i>Chat</i> no programa do cliente 1; (c) <i>Chat</i> no programa do cliente 2.	136
Figura 4.14. Tabela de clientes conectados.	138
Figura 4.15. Tabela do serviço câmera.	138
Figura 4.16. Tabela do serviço dados.	139
Figura 4.17. Edição das faixas de supervisão dos parâmetros supervisionados pelo programa do servidor.	141
Figura 4.18. Seleção do tipo de faixa de supervisão.	141
Figura 4.19. Edição dos patamares da faixa de supervisão selecionada.	142
Figura 4.20. Programa para exibição de mensagens através dos painéis eletrônicos.	143
Figura 4.21. Programa do painel eletrônico implementado no servidor e no cliente: (a) Programa do servidor; (b) Programa do cliente.	145

Figura 4.22. Tela de configuração de mensagens automáticas: (a) Mensagens automáticas do painel de <i>status</i> ; (b) Mensagens automáticas do painel de dados.....	146
Figura 4.23. Receptor de infravermelho projetado e controle remoto utilizado.....	149
Figura 4.24. Edição da tensão de armadura do motor através do painel eletrônico de <i>status</i>	149
Figura 4.25. Edição da velocidade de rotação do eixo do motor através do painel eletrônico de <i>status</i>	150

LISTA DE TABELAS

Tabela 3.1. Saída do circuito <i>multiplex</i> de 8 canais, em função de suas entradas digitais.	30
Tabela 3.2. Tabela verdade para o inversor lógico.	49
Tabela 3.3. Valores tabelados para o PID discreto, utilizando o método proposto por Ziegler-Nichols.	70

LISTA DE ABREVIATURAS E SIGLAS

A/D – Analógico/Digital
BCD – *Binary Coded Decimal*
CLP – Controlador Lógico Programável
D/A – Digital/Analógico
DCS – *Distributed Control Systems*
ERB – Estação Rádio Base
EPROM – *Erasable and Programmable Read Only Memory*
FTP – *File Transfer Protocol*
GPRS – *General Packet Radio Services*
HTTP – *HyperText Transfer Protocol*
IP – *Internet Protocol*
IPX/SPX – *Internetwork Packet Exchange/Sequenced Packet Exchange*
IRC – *Internet Relay Chat*
ISP – *In System Programming*
J-FET – *Junction Field Effect Transistor*
LSB – *Least Significant Bit*
MSB – *Most Significant Bit*
MSN – *Microsoft Service Network*
NETBEUI – *NETBIOS Extended User Interface*
NETBIOS – *Network Basic Input/Output System*
NTC – *Negative Temperature Coefficient*
PCS – *Process Control Systems*
PROM – *Programmable Read Only Memory*
RAM – *Random Access Memory*
RED – *Random Early Detection*
RMS – *Root Mean Square*
ROM – *Read Only Memory*
SCADA – *Supervisory Control and Data Acquisition Systems*
SMD – *Surface Mounting Device*
TCP – *Transmission Control Protocol*
WWW – *World Wide Web*

1 Introdução

1.1 Motivação e Relevância do Trabalho

O surgimento da *internet* teve um papel tão importante na sociedade como o surgimento do telefone ou mesmo da imprensa. Através da *internet*, o fluxo das informações foi alterado, de forma a permitir que qualquer pessoa conectada na rede possa obter informações de qualquer assunto relevante em poucos segundos.

Essa facilidade de transmitir e receber informações de qualquer parte do mundo possibilitou o emprego de sistemas supervisórios por intermédio da *internet*, permitindo assim, que sistemas distantes geograficamente possam se comunicar, sem que seja necessário um grande investimento para realizar essa comunicação.

No Brasil, as empresas têm utilizado principalmente sistemas supervisórios do tipo SCADA, os quais utilizam como *drivers* os controladores lógicos programáveis (CLP's).

Outra alternativa encontrada pelas empresas é a utilização de placas de aquisição que possuem *software* próprio, tais como as placas de aquisição da *National Instruments*.

Entretanto, ambas alternativas apresentam o mesmo problema, ou seja, tanto o *hardware* quanto o *software* empregado possuem custo muito elevados, que só se justificariam se a planta industrial a ser controlada tivesse uma grande dimensão.

Assim, por mais simples que fosse o sistema a ser supervisionado, o custo permaneceria o mesmo, o que na maioria das vezes inviabiliza o sistema de supervisionamento.

Outra limitação dessas metodologias é que o usuário do sistema supervisório fica limitado aos recursos do *software* adotado, que podem não prover todos os tipos de serviços necessários pelo cliente.

Portanto, o desenvolvimento de um sistema supervisório mais compacto, mas que possua um baixo custo agregado, é de suma importância, pois permitirá que sistemas mais simples possam também ser supervisionados, sem a necessidade de ter que arcar com os elevados custos quando se utiliza de sistemas supervisórios comerciais.

1.2 Objetivo e Justificativa da Dissertação

O objetivo desta dissertação é apresentar um sistema supervisório via TCP/IP, aplicado em máquinas elétricas, que possua um custo muito inferior ao das alternativas mencionadas anteriormente, podendo ser aplicado na supervisão de diferentes tipos de sistemas, sendo, portanto, uma alternativa viável na supervisão remota de sistemas.

O sistema a ser supervisionado nesse projeto consiste de um motor DC, e os parâmetros a serem supervisionados consistem de suas grandezas dinâmicas, tais como corrente de armadura, tensão de armadura, temperatura do motor, velocidade de rotação do eixo do motor, dentre outros.

Uma das propostas desse projeto é a utilização de uma câmera digital de forma a permitir que cada cliente possa ter acesso à imagem do sistema em tempo real, podendo assim, realizar uma inspeção visual do sistema, caso seja necessário.

Entretanto, a principal inovação em relação aos programas supervisórios comerciais é a utilização de painéis eletrônicos para a exibição dos dados e

informações do sistema supervisionado, fornecendo assim, uma nova interface visual de forma a facilitar sua supervisão.

Para esse projeto é proposta a utilização de dois painéis eletrônicos conectados ao computador remoto (Cliente), sendo um monocromático e um tricolor. O painel monocromático tem a função de exibir as mensagens de *status* e informações do sistema, enquanto que o painel tricolor tem a função de exibir os dados supervisionados.

Para este último painel, a cor de exibição de cada parâmetro é mostrada de acordo com seu *status* em relação à sua faixa de supervisão pré-programada. Assim, as cores representam uma função de *status* e não apenas um apelo visual, como é o caso da maioria das aplicações dos painéis eletrônicos coloridos.

O projeto dos circuitos de acionamento dos painéis eletrônicos nessa pesquisa representa uma nova contribuição, pois vem para desfazer o domínio tecnológico presente nas indústrias de painel eletrônico, trazendo essa tecnologia para o meio acadêmico, permitindo assim, que novos trabalhos nessa área possam ser realizados a partir desse.

Uma interface de entrada via controle remoto infravermelho também é proposta, permitindo que o usuário possa ajustar no painel de status a ação de controle, utilizando-se para isso de um controle remoto convencional.

1.3 Organização da Dissertação

No Capítulo 2 são abordados o estado da arte e aspectos gerais dos sistemas supervisórios, descrevendo suas partes constituintes, seus tipos e topologias.

No Capítulo 3 são tratadas as principais estruturas do sistema supervisorio desenvolvido, apresentando todas as metodologias empregadas desde o sistema de aquisição de dados até o sistema de supervisionamento e exibição de parâmetros.

Todos os resultados obtidos a partir do sistema supervisorio desenvolvido são apresentados no Capítulo 4.

Finalmente, no Capítulo 5 são descritas as conclusões relativas ao trabalho desenvolvido, enfatizando os temas de projetos futuros passíveis de serem investigados a partir desse.

2 Estado da Arte e Aspectos de Sistemas Supervisórios

2.1 Introdução

O advento da *internet* teve um impacto tão profundo na sociedade, como o telefone ou mesmo a imprensa. Enquanto o telefone demorou cerca de 75 anos para que o número de utilizadores chegasse a 50 milhões, a World Wide Web (WWW) realizou o mesmo feito em apenas 4 anos.

Por intermédio da *internet*, alteraram-se as relações sociais, e o fluxo da comunicação científica foi reestruturada, permitindo que a informação fosse produzida e armazenada em diferentes espaços e acessada por usuários distantes geograficamente, facilitando assim, o desenvolvimento de novas pesquisas.

No Brasil, o estado da arte na comunicação entre sistemas remotos está com a utilização de sistemas de aquisição conectados a módulos de celular com tecnologia GPRS. Por meio desses sistemas, torna-se possível transmitir os dados adquiridos, via GPRS, a uma ERB (Estação Rádio Base), a qual os disponibiliza em um IP, podendo assim, serem acessados por qualquer computador conectado à *internet*.

Embora os sistemas remotos sejam amplamente utilizados nas mais diversas áreas da engenharia e computação, na área de máquinas elétricas esse assunto é relativamente novo, merecendo assim um destaque especial.

Nas últimas duas décadas, os sistemas de controle e supervisionamento sofreram enormes mudanças com o advento dos computadores, que tem se tornado

dominante em todas as áreas da engenharia, tanto como ferramenta de análise e projeto quanto como dispositivo de automação e controle de processos.

Em [1], compara-se os sistemas de controle clássicos com os controles do tipo TCP/IP. A vantagem deste último em relação aos demais se deve ao compartilhamento de dados entre programas supervisores e dispositivos, possibilitando assim que diferentes sistemas se comuniquem, independentemente da plataforma adotada.

Dessa forma, um servidor que execute seu programa em plataforma *Linux* pode se comunicar com seus clientes que façam o mesmo em plataforma *Windows*.

No entanto, um dos problemas encontrados ao se fazer o controle via TCP/IP se deve ao congestionamento na rede.

Em [2] é relatado que a qualidade de serviços utilizando a *Internet*, tais como serviços de áudio e vídeo, é dependente da qualidade da rede.

Quanto mais precária for a qualidade da rede, menor terá que ser a qualidade do tipo de serviço prestado, de forma a se enviar um número mínimo de bytes à rede. Caso contrário, uma grande perda de pacotes poderia ocorrer, o que poderia gerar uma realimentação positiva de pacotes, aumentando ainda mais seu congestionamento.

Essa realimentação positiva se deve ao fato que todo pacote perdido na rede terá que ser reenviado novamente. Nesse caso, a rede ficaria ainda mais congestionada e se degradaria ao longo do tempo.

Se fosse necessário, por exemplo, transmitir imagens com uma maior resolução, arquivos maiores teriam que ser enviados, o que acarretaria em um maior número de *bytes* a serem transmitidos pelo servidor, e um maior tempo para se receber a imagem pelo cliente.

A única maneira de se manter o mesmo intervalo de envio e recepção seria aumentando a banda utilizada, o que acarretaria em maiores custos para se manter o sistema, e mesmo assim, o problema de congestionamento na rede continuaria sendo um fator crítico do sistema.

Embora novas arquiteturas, tais como o TCP/IP *Differentiated Services*, tenham sido introduzidas para melhorar o tráfego, o problema de congestionamento permanece crítico e de alta prioridade.

Como uma tentativa de melhorar o tráfego, são propostos em [2] esquemas alternativos, tais como o *Random Early Detection* (RED) e suas variantes, de modo a utilizar sistemas *Fuzzy* para se determinar o melhor intervalo de tempo para se enviar os dados à rede, podendo obter dessa forma uma melhor qualidade para os diversos tipos de tráfego.

Em [3], uma formulação canônica no Espaço de Estados é proposta, levando-se em consideração a natureza não determinística do controle e dos atrasos obtidos na rede. Segundo os autores, tais atrasos podem desestabilizar e degradar o desempenho de sistemas teleoperados através da *internet*.

Um sistema supervisório pode ser definido como um sistema que monitora processos executados em uma planta industrial, a partir da visualização das variáveis do processo, bem como das ações tomadas pelo sistema de automação [4].

Esses sistemas são empregados com a finalidade de tornar possível o reconhecimento de possíveis falhas em componentes da planta, antes que estas efetivamente ocorram.

Dessa forma, os sistemas supervisórios auxiliam o usuário na monitoração do processo, de forma a identificar possíveis defeitos, e permitir a tomada de ações de controle, de modo a manter um bom funcionamento do sistema.

2.2 Tipos de Sistemas Supervisórios

Os Sistemas de Supervisão e Aquisição de Dados, ou abreviadamente SCADA, são sistemas que utilizam *software* para supervisionar as variáveis e dispositivos do sistema, conectados por meio de *drivers* específicos [4].

Os *drivers* mais utilizados atualmente nos sistemas SCADA são os CLP's, os quais são compostos por sistemas microprocessados que desempenham funções de controle de diversos tipos e níveis de complexidade [5].

O CLP é um equipamento com *hardware* e *software* compatível com aplicações industriais, que utiliza uma memória programável para armazenar instruções e para implementar funções específicas, tais como lógica, seqüenciamento, temporização, contagem e aritmética, controlando por meio de módulos de entradas e saídas vários tipos de máquinas ou processos.

Em [6] é descrita a supervisão de sistemas industriais através da utilização de sensores e atuadores conectados a um único computador, o qual é responsável por supervisionar e controlar o sistema, consistindo, portanto, de uma topologia Monoposto ou *Standalone*. Se esse computador for ligado em rede, de forma a permitir o compartilhamento de dados e informações do sistema entre vários clientes, ter-se-á uma topologia Servidor/Multi-Clientes.

Estendendo essa idéia, os sistemas supervisórios podem assumir basicamente quatro tipos de topologias, ou seja: Monoposto, Servidor/Cliente, Servidor/Multi-Clientes, e Múltiplos Servidores/Clientes.

Na topologia Monoposto existe somente um computador, e este é responsável por fazer a supervisão e controle do sistema.

Na topologia Servidor/Cliente existem dois computadores, sendo um com a função de prover serviços (Servidor), enquanto o outro se utiliza desses serviços (Cliente). Nesse caso, os dados coletados pelo servidor são acessados remotamente por um único cliente conectado na rede, o qual é responsável pela sua supervisão.

Na topologia Servidor/Multi-Clientes, há apenas um servidor de dados, mas diferentemente da topologia anterior, existem múltiplos clientes que podem se conectar simultaneamente ao mesmo servidor de dados e realizar sua supervisão.

A última topologia consiste de Múltiplos Servidores/Clientes. Nesse caso, ao invés de haver somente um servidor de dados, múltiplos servidores podem estar disponíveis, permitindo que cada cliente possa se conectar a mais de um servidor ao mesmo tempo, provendo assim um maior número de serviços.

2.3 Composição Estrutural dos Sistemas Supervisórios

Os componentes físicos de um sistema de supervisão podem ser resumidos de forma simplificada em: sensores e atuadores; rede de comunicação; estações remotas (aquisição/controle); e estações de monitoração [4].

Os sensores são dispositivos que convertem grandezas físicas, tais como velocidade, força e temperatura, em sinais analógicos, passíveis de serem lidos pela estação remota.

Os atuadores são dispositivos que intervêm sobre o sistema, de forma a executar determinada ação de controle, como por exemplo, ligar ou desligar determinada máquina ou equipamento.

O processo de aquisição de dados inicia-se nas estações remotas, que fazem a coleta dos dados e posterior envio destes, por meio de sua rede de comunicação, às estações de monitoração.

Existem vários tipos de rede de comunicação que podem ser utilizadas na comunicação entre estações remotas e de monitoração, tais como cabos *Ethernet*, fibras ópticas, linhas *dial-up*, rádio-modems, etc.

As estações de monitoração são as unidades responsáveis por coletar as informações geradas pelas estações remotas e agir em conformidade com os eventos detectados, podendo ser centralizadas em um único computador ou distribuídas através de uma rede de computadores.

Nas Figuras 2.1 a 2.4 são representados os diagramas estruturais para os quatro tipos principais de topologia de sistemas supervisórios, conforme mencionado anteriormente.

Figura 2.1. Composição estrutural de um sistema supervisório do tipo Monoposto.

Figura 2.2. Composição estrutural de um sistema supervisor do tipo Servidor/Cliente.

Figura 2.3. Composição estrutural de um sistema supervisor do tipo Servidor/Multi-Clientes.

Figura 2.4. Composição estrutural de um sistema supervisorio do tipo Múltiplos Servidores/Clientes.

2.4 Sistemas Supervisorios via TCP/IP

No Brasil, os sistemas supervisorios via TCP/IP têm sido utilizados em diversas aplicações. Em [7], Cagnon e Fontes propõem a utilização do programa supervisorio Elipse SCADA para gerenciar e racionalizar os recursos utilizados em um viveiro de mudas. Nesse projeto, é empregada uma unidade remota de sensores, a qual coleta e transmite os dados via RF a um concentrador de dados, enquanto um CLP é utilizado para controlar as válvulas e bombas, a fim de proporcionar o fornecimento de água e nutrientes às mudas e sementes do viveiro.

Segundo os autores, a utilização de sistemas supervisórios na automação de sistemas de produção constitui uma alternativa vantajosa, pois, além da flexibilidade que oferecem, visa também o uso racional dos recursos e a qualidade no processo produtivo.

Trabalho semelhante a esse é apresentado por Saraiva e Cansado em [8], onde é utilizado um sistema supervisório via TCP/IP para monitorar e controlar uma casa de vegetação. Em tal trabalho, um CLP é utilizado para o controle da temperatura, umidade do ar e outras variáveis de interesse da estufa. Segundo os autores, a automação de casas de vegetação desempenha um papel fundamental, pois possibilita o controle apropriado para a produção agrícola de forma otimizada, além de reduzir os custos por unidade de produção, visando sempre o aumento da qualidade.

Em [9] é discutido que a introdução crescente de sistemas de automação industrial torna cada vez mais complexa a tarefa de monitorar, controlar e gerenciar os sistemas produtivos. Dentre os diversos sistemas de controle e gerenciamento de produção, os que mais têm se difundido segundo os autores são os PCS, os SCADA e os DCS.

Uma IHM de supervisão de um CLP foi desenvolvida por Pupo e Gonzaga em [10], sendo utilizada no controle de processos por intermédio da *internet*. As IHM são interfaces amigáveis e eficientes, sendo construídas visando os operadores como usuários finais.

Os sistemas de supervisão e monitoramento por meio da *internet* são também abordados em [11], onde é discutido que atualmente os sistemas supervisórios podem ter uma arquitetura aberta, ligados em rede, possibilitando que o fluxo de

dados do processo ultrapasse o limite das paredes da empresa e percorra o mundo usando dos meios de comunicação existentes.

Em [12], é proposto um sistema supervisor via *internet* para o monitoramento do sistema de esgoto sanitário da CAESB (Companhia de Saneamento Ambiental do Distrito Federal). Nesse sistema, um CLP é utilizado para a coleta da vazão e do nível do poço de sucção, e similarmente a esse projeto, é utilizada uma câmera digital no sistema. Entretanto, ao invés de monitorar o sistema, essa é utilizada para a vigilância patrimonial da unidade.

Uma metodologia para a implantação de laboratórios controlados através da *internet* é apresentada em [13], de modo a permitir o desenvolvimento de experimentos remotos na área de automação e robótica.

Sistemas de automação teleoperados via *internet* são estudados em [14]. Nesse trabalho, é apresentado um robô *wireless* controlado remotamente através da *internet*. Com a utilização de uma câmera digital instalada no robô, o usuário pode fazer sua movimentação através do teclado, baseado nas imagens recebidas pela rede.

Da mesma forma que em [14], é apresentado por Goldberg et. al. em [15] um robô controlado por meio da *internet*, o qual usando de um braço mecânico pode jogar jogos de tabuleiro.

Segundo Hu et. al. em [16], os sistemas teleoperados pela *internet* devem possuir um alto grau de autonomia e inteligência local, de forma a contornar problemas de atraso na rede e largura de banda limitada.

A vantagem de se monitorar e controlar um sistema remotamente é poder realizar tarefas perigosas sem expor os trabalhadores a algum tipo de risco [17]. Por

essa razão, tais sistemas são essenciais em usinas nucleares, químicas, plataformas de petrolíferas, etc.

Em [18] é proposta a utilização da linguagem de programação Java para se realizar o controle e monitoração remota de sistemas através da internet. Segundo os autores, o Java tem a vantagem da portabilidade e independência de plataforma, permitindo realizar o controle e supervisão do sistema por meio de qualquer navegador de internet.

Entretanto, o Java não gera arquivo executável, o que faz com que o usuário tenha que possuir o *plugin* de Java instalado na máquina.

O sistema supervisorio desenvolvido nessa pesquisa utiliza como linguagem de programação o Borland Delphi, o qual além de ser atualmente um dos compiladores mais rápidos em disponibilidade, gera arquivos executáveis que não dependem de outros arquivos ou DLLs. Além disso, a interface de programação do Delphi é bem mais amigável e intuitiva do que a do Java, e por essa razão, foi o compilador adotado nesse projeto.

Em [19], é apresentado um sistema de telemanufatura através da *internet* aplicado ao processo de oxicorte. Com a utilização de uma *webcam*, o processo realizado pela máquina de oxicorte é controlado e supervisionado remotamente.

Na pesquisa realizada por Luo et. al. em [20], são apresentadas as vantagens e oportunidades que a *internet* trouxe nos mais variados campos de trabalho, ressaltando o uso da internet no controle de máquinas e processos remotamente, e mostrando algumas de suas limitações, tais como: perda de dados; atraso na rede; problemas de transmissão segura de dados, etc.

Dessa forma, conclui-se que os sistemas supervisorios via TCP/IP são uma poderosa ferramenta no monitoramento e controle de sistemas, permitindo que

informações de um processo possam ser acessadas por qualquer computador conectado à rede, podendo ser empregados nas mais diversas áreas e aplicações.

Para se descrever o funcionamento de um sistema supervisório via TCP/IP, torna-se necessário primeiramente definir os elementos básicos de uma rede de computadores. Não sendo o foco desse trabalho o aprofundamento nesse assunto, serão apresentados a seguir apenas os elementos básicos para a compreensão do funcionamento de um sistema supervisório via *internet*.

Assim, primeiramente, para que dois computadores possam compartilhar informações entre si, é necessário que estes adotem as mesmas regras para o envio e recepção de informações. O conjunto dessas regras é denominado Protocolo de Comunicação [6]. Portanto, para que computadores em uma mesma rede possam se comunicar é necessário que estes utilizem o mesmo protocolo de comunicação.

No início da *internet*, os principais protocolos de comunicação utilizados eram os listados a seguir:

- TCP/IP.
- NETBEUI.
- IPX/SPX.
- Apple Talk.

Entretanto, com a popularização da *internet* e o aumento exponencial do número de usuários, o protocolo TCP/IP tornou-se um padrão mundial.

O protocolo TCP/IP não é formado somente pelo protocolo TCP e pelo protocolo IP, mas sim por um conjunto de protocolos sobre o qual a *internet* e a maioria das redes comerciais funcionam, sendo o TCP e o IP os mais conhecidos e,

por essa razão, o nome desse conjunto de protocolos acabou sendo batizado de protocolo TCP/IP [21].

Em uma rede que utiliza o protocolo TCP/IP, todos os equipamentos da rede (servidores, *hubs*, *switchs*, impressoras de rede, etc) precisam ser identificados na rede através de dois números, os quais são definidos por:

- Máscara da Sub-Rede
- Número IP

O número IP e a Máscara da Sub-Rede são formados por 4 octetos, conforme mostrado a seguir:

$$\text{Número IP} = X . Y . Z . W$$

$$\text{Mascara da Sub-Rede} = A . B . C . D$$

onde cada octeto (X, Y, Z, W, A, B, C e D) pode variar de 0 a 255, sendo, portanto, um número de 8 *bits*, mas que normalmente é representado na sua forma decimal [22].

O número IP serve para identificar a máquina na rede. Portanto, duas máquinas em uma mesma rede não podem possuir o mesmo número IP.

A Máscara de Sub-Rede serve para identificar a sub-rede a qual determinado computador pertence. Os valores convencionais para esses octetos são 0 ou 255, mas também podem ser usados valores intermediários entre esses dois valores, embora não seja o usual.

As classes dos números IP e as correspondentes configurações das Máscaras da Sub-Rede não serão abordadas nessa pesquisa, pois o foco desse

trabalho é explicar como o programa do servidor se comunica com o programa do cliente.

Portanto, pode-se dizer que esses dois parâmetros identificam o computador na rede, mas para que um determinado programa se comunique com outro por meio da *internet* é necessário atribuir a ele um número de identificação de porta. Este número serve para diferenciar o programa dos demais aplicativos do computador que acessam a *internet*, de modo que cada tipo de serviço possua um número exclusivo de identificação de porta.

Existe em cada computador uma quantidade de 65536 portas TCP, numeradas de 1 a 65536. Cada porta pode ser utilizada por um programa ou serviço diferente, de forma que em um mesmo servidor, poderia se ter até 65536 diferentes tipos de serviços ativos simultaneamente [6].

As portas TCP mais utilizadas são as portas de 1 a 1024, que são reservadas para os serviços mais conhecidos, como FTP, servidores de *e-mail*, HTTP, etc. Portanto, as portas de 1 a 1024 são em geral bloqueadas, não podendo assim, serem utilizadas pelo usuário [21].

Dessa forma, para o sistema supervisorio desenvolvido nessa pesquisa, pode-se utilizar qualquer porta na faixa de 1025 a 65536, de forma a se realizar a comunicação entre o programa do servidor e do cliente.

A topologia adotada para o sistema supervisorio consiste da Servidor/Multi-Clientes, pois como só se tem um sistema de aquisição e um único sistema a ser monitorado, não há razão para se utilizar um sistema com mais de um servidor.

Definido assim as configurações de rede e a topologia adotada, para que um cliente se conecte ao servidor de dados, é necessário que este conheça seu número IP e seu número de identificação de porta.

Para que o servidor se comunique com o cliente, ele precisará também conhecer seu número IP. Entretanto, este número é obtido quando um cliente faz uma solicitação de conexão ao servidor, pois nesse caso, o número IP do cliente ficará registrado no cabeçalho de solicitação do datagrama IP enviado.

Portanto, em um sistema supervisório via TCP/IP, apenas o cliente precisa conhecer o número IP e a porta de comunicação utilizada pelo programa do servidor, já que o servidor obtém essas informações do cliente, quando este faz uma solicitação de conexão.

3 Estrutura do Sistema Supervisório Desenvolvido

3.1 Introdução

Nesse capítulo será descrito o sistema supervisório desenvolvido, iniciando com uma visão geral do sistema, seguida de uma explicação detalhada de suas partes constituintes.

As estruturas que compõem o sistema supervisório são:

- Sistema de Aquisição de Dados.
- Sistema de Controle.
- Sistema de Transmissão e Recepção de Dados via TCP/IP.
- Sistema de Supervisão de Parâmetros.
- Sistema de Visualização de Parâmetros via Painel Eletrônico de Mensagens.
- Interface de Entrada via controle Remoto Infravermelho.

Na Seção 3.2 é apresentada a visão geral do sistema supervisório desenvolvido.

Na Seção 3.3 é abordado o Sistema de Aquisição de Dados, onde é apresentado o *hardware* projetado, bem como as metodologias adotadas para se realizar a aquisição de cada parâmetro supervisionado.

Na Seção 3.4 é descrito o Sistema de Controle utilizado para se controlar a velocidade de rotação do eixo do motor DC.

Na Seção 3.5 é apresentado o Sistema de Transmissão e Recepção de Dados via TCP/IP, descrevendo os programas do cliente e do servidor, responsáveis por fazer a supervisão do sistema, bem como o acionamento dos painéis eletrônicos.

O Sistema de Supervisão de Parâmetros é discutido na Seção 3.6, e enfoca a metodologia adotada para se realizar a supervisão dos parâmetros do sistema.

Na Seção 3.7 é descrito o Sistema de Visualização de Parâmetros via Painel Eletrônico de Mensagens, onde são abordados o *hardware* e *software* desenvolvidos para os dois tipos de painel eletrônico utilizados no projeto, de modo a permitir a visualização dos parâmetros monitorados através destes.

Por fim, na Seção 3.8 é apresentada a Interface de Entrada via Controle Remoto Infravermelho, empregada para se ajustar a velocidade de rotação ou tensão do motor DC, utilizando-se de um controle remoto infravermelho.

3.2 Visão Geral do Sistema Supervisório Desenvolvido

Na Figura 3.1 é ilustrado o diagrama em blocos do sistema de supervisão e exibição de parâmetros proposto nesse projeto.

Figura 3.1. Diagrama em blocos do sistema supervisório desenvolvido.

Como se pode visualizar por esse diagrama, o bloco mais à esquerda da figura corresponde ao sistema a ser monitorado, que para esse projeto consiste de um motor DC.

O bloco seguinte corresponde ao sistema de aquisição de dados, onde os parâmetros de interesse do sistema são coletados, digitalizados e enviados ao computador servidor.

Para a aquisição de cada parâmetro do sistema são utilizados sensores e circuitos especialmente projetados, de forma a transformar as grandezas a serem medidas em tensões analógicas passíveis de serem medidas.

Utilizando-se de circuitos de condicionamento de sinais, essas tensões são condicionadas de acordo com a máxima tensão de entrada permitida para o conversor A/D utilizado.

Na próxima seção será mostrado que dentro desse bloco há um circuito multiplexador, que faz com que os sinais de entrada sejam selecionados e digitalizados um de cada vez, de forma a se utilizar apenas um conversor A/D no sistema de aquisição.

Os parâmetros digitalizados do sistema são então armazenados temporariamente nos registradores do microcontrolador utilizado, e posteriormente enviados via porta serial ao computador servidor.

O computador servidor é responsável tanto por encapsular e enviar os dados via TCP/IP aos clientes conectados, como o de fazer o acionamento do sistema de ajuste de tensão do motor.

Esse sistema de ajuste é constituído por um microcontrolador, responsável por controlar um conversor Digital/Analógico (D/A) conectado a um *driver* de corrente. Dessa maneira, ao receber um valor de tensão a ser ajustado, o

microcontrolador acionará convenientemente o conversor D/A, regulando assim na saída do *driver* de corrente, a tensão requerida pelo cliente.

No caso do cliente desejar ajustar a velocidade de rotação do eixo do motor, foi projetado um controlador PID discreto no próprio programa do servidor, de modo que este último calcula a ação de controle (tensão a ser ajustada no motor) e a envia ao sistema de ajuste de tensão. Dessa forma, a cada período de amostragem é calculada uma nova ação de controle e enviada ao sistema de ajuste de tensão, regulando dessa forma, a velocidade de rotação do eixo do motor, de acordo com o valor desejado pelo usuário.

Além dos parâmetros monitorados do sistema, a imagem deste também é obtida utilizando-se de uma câmera digital. Nesse caso, os *bytes* da imagem são encapsulados via TCP/IP, e enviados aos clientes conectados, permitindo assim, que cada cliente possa realizar uma inspeção visual do sistema.

O computador remoto (Cliente) tem a função de verificar a faixa de supervisão pré-programada para cada parâmetro de entrada do sistema, bem como a indicação de seus valores nos painéis eletrônicos.

Por intermédio do programa do cliente, o usuário pode enviar comandos ao servidor solicitando o ajuste da tensão de alimentação do motor DC ou requerendo o ajuste da velocidade de rotação do seu eixo.

Em ambos os casos, ao receber esse tipo de comando, o computador servidor enviará ao microcontrolador que controla o conversor D/A, um valor de tensão a ser ajustado, de modo que este último regule a tensão requerida.

Os painéis eletrônicos são controlados pelo computador cliente, entretanto, podem também ser acessados pelo servidor. Estes foram projetados de forma a

possuírem um *hardware* dedicado à exibição de mensagens, não sobrecarregando assim o processamento do computador cliente.

O acesso a eles é realizado utilizando-se da porta serial do computador cliente, ou seja, via protocolo RS-232. Dessa forma, para a exibição de uma única mensagem no painel, basta o cliente enviar ao painel escolhido os *bytes* da mensagem a ser exibida, bem como os *bytes* que representam o efeito e velocidade da mensagem, conforme selecionados no programa do cliente.

Conforme será explicado na Seção 3.7, cada painel possui um único microcontrolador, o qual é responsável por fazer o acionamento das suas respectivas matrizes de *led*. Dessa forma, o programa de cada microcontrolador tem que se comunicar ao mesmo tempo com o computador cliente, como também fazer a exibição das mensagens no painel.

3.3 Sistema de Aquisição de Dados

Nessa seção serão descritas as técnicas e metodologias adotadas na aquisição de cada parâmetro sob supervisão do sistema. Entretanto, é preciso primeiramente mencionar quais são os parâmetros de interesse a serem supervisionados.

Como o sistema adotado consiste de um motor DC, foram então escolhidos parâmetros do motor, da rede elétrica, parâmetros externos e do próprio sistema de aquisição. Esses parâmetros são listados a seguir:

Parâmetros do Motor

- Tensão de Armadura do Motor.
- Corrente de Armadura do Motor.

- Velocidade de Rotação do Eixo do Motor.
- Temperatura do Motor.

Parâmetros da Rede Elétrica

- Tensão da Rede.
- Frequência da Rede.

Parâmetros Externos

- Temperatura Ambiente.
- Tensão de Alimentação do Sistema Microprocessado.
- Tensão DC Externa.

A Tensão DC Externa consiste de uma tensão DC variável, ajustada através de um potenciômetro, utilizada para calibrar o sistema de aquisição e para testar o sistema de supervisão de parâmetros descrito na Seção 3.6.

Uma observação referente aos parâmetros dinâmicos do motor é que não foi incluído seu torque elétrico, pois se pode obter mais facilmente esse valor através da Equação (3.1).

$$T_e = k_t \cdot \phi_P \cdot i_a \quad (3.1)$$

onde:

T_e – Torque Elétrico do Motor [N.m].

k_t – Constante de Torque Elétrico $\left[\frac{\text{N.m}}{\text{Wb.A}} \right]$.

ϕ_P – Fluxo Magnético por pólo [Wb].

i_a – Corrente de Armadura do Motor [A].

Como o motor DC utilizado nesse projeto consiste de um motor de imã permanente, é comum o fabricante embutir a constante de torque elétrico, que depende de fatores construtivos do motor, tendo o valor do fluxo magnético por pólo em uma única constante. Dessa maneira, a equação anterior se resume à Equação (3.2), ou seja :

$$T_e = K_t \cdot I_a \quad (3.2)$$

onde:

T_e – Torque Elétrico do Motor [N.m]

K_t – Constante de Torque Elétrico $\left[\frac{\text{N.m}}{\text{A}} \right]$

i_a – Corrente de Armadura do Motor [A]

Pela equação anterior, tem-se que o torque elétrico do motor será diretamente proporcional à sua corrente de armadura. Como o valor da corrente é um dos parâmetros adquiridos pelo sistema, necessita-se apenas conhecer o valor da sua constante de torque elétrico, a qual pode ser obtida diretamente a partir das especificações do fabricante do motor.

Por meio dessa metodologia determina-se o torque elétrico mais facilmente do que utilizando um circuito complexo para a medição do mesmo.

Para o sistema supervisório desenvolvido, utilizou-se um motor DC de imã permanente de 12V e corrente nominal de 200mA sem carga.

Na Figura 3.2 é ilustrada a conexão entre motor, sistema de aquisição de dados, sistema de ajuste de tensão do motor e computador servidor.

Figura 3.2. Ilustração da conexão entre motor, sistema de aquisição de dados, sistema de ajuste de tensão do motor e computador servidor.

Nessa figura, o Circuito de Aquisição Analógica e Condicionamento de Sinais tem a função de converter os parâmetros a serem supervisionados em tensões analógicas, passíveis de serem medidas. Para isso, são utilizados sensores e circuitos especialmente desenvolvidos para a aquisição de cada tipo de parâmetro a ser supervisionado.

Após essa conversão, o sistema de aquisição de dados faz a digitalização e posterior envio dos dados coletados ao computador servidor.

Como mencionado brevemente na introdução, o sistema de aquisição de dados consiste de um circuito *multiplex* conectado a um conversor A/D, e este último

a um microcontrolador. A saída de cada circuito de condicionamento de sinal é conectada a uma das entradas do circuito *multiplex*.

Como será explicado no fim dessa seção, para a aquisição da frequência da rede elétrica não foi utilizado o conversor A/D do sistema de aquisição, mas as entradas pulsadas do próprio microcontrolador. Para isso, foi projetado um circuito conveniente de forma a gerar pulsos diretamente proporcionais ao valor atual desse parâmetro.

Portanto, como se tem um total de nove parâmetros a serem monitorados, oito são coletados a partir do conversor A/D e um por intermédio das entradas pulsadas do próprio microcontrolador.

Como se pretende utilizar apenas um conversor A/D no sistema de aquisição, é necessário utilizar um circuito *multiplex* de 8 canais de entrada, de modo que este faça a seleção da entrada analógica a ser digitalizada.

Um circuito *multiplex* de 8 canais de entrada é representado na Figura 3.3:

Figura 3.3. Representação de um circuito *multiplex* de 8 canais de entrada.

Nessa figura, as entradas analógicas do circuito são selecionadas através das entradas digitais (A, B e C). Uma determinada entrada analógica é selecionada e

chaveada para a saída do circuito, conforme o valor atual desses 3 *bits*. Os possíveis valores de entrada e saída correspondente são mostrados na Tabela 3.1.

Tabela 3.1. Saída do circuito *multiplex* de 8 canais, em função de suas entradas digitais.

Entradas Digitais			Saída
A	B	C	
0	0	0	E ₁
0	0	1	E ₂
0	1	0	E ₃
0	1	1	E ₄
1	0	0	E ₅
1	0	1	E ₆
1	1	0	E ₇
1	1	1	E ₈

Assim, por exemplo, se os valores dos *bits* de seleção ABC forem respectivamente 010, a entrada selecionada seria a entrada E₃ e, portanto, qualquer sinal presente nessa entrada apareceria na saída do circuito *multiplex*.

Dessa forma, foi utilizado esse mesmo circuito *multiplex* no projeto, de modo a chavear os 8 sinais analógicos do sistema, sendo cada sinal selecionado e digitalizado por vez, necessitando assim, utilizar apenas um único conversor A/D.

O microcontrolador utilizado é o responsável tanto pelo acionamento do conversor A/D como do circuito *multiplex*, de forma a selecionar o sinal a ser digitalizado.

O esquema de ligação entre o circuito *multiplex*, conversor A/D e microcontrolador é mostrado na Figura 3.4, onde:

E₁ – Tensão de Armadura do Motor.

E₂ – Corrente de Armadura do Motor.

E_3 – Temperatura do Motor.

E_4 – Velocidade de Rotação do Eixo do Motor.

E_5 – Temperatura Ambiente.

E_6 – Tensão da Rede Elétrica.

E_7 – Tensão de Alimentação do Sistema Microprocessado.

E_8 – Tensão DC Externa.

Figura 3.4. Sistema de aquisição – entradas analógicas.

As entradas E_i correspondem à saída de cada circuito de condicionamento de sinal referente a cada parâmetro a ser supervisionado.

O conversor A/D utilizado no projeto é o ADS7805 da *Texas Instruments*. Esse conversor tem como características principais:

- Tempo de conversão de até 10 μ s.
- Resolução de 16 *bits*.
- Tensão de entrada de ± 10 V.
- Erro de ± 4 LSB.

Como o conversor A/D suporta tensões de entrada na faixa de $\pm 10V$, no projeto de cada circuito de condicionamento de sinal deve-se observar que a máxima tensão de saída deve estar também entre $\pm 10V$.

A resolução de uma tensão a ser medida pode ser melhorada, fazendo com que a excursão do sinal de saída de cada circuito de condicionamento de sinal, aproxime-se da excursão máxima da tensão de entrada permitida para o conversor A/D, ou seja, $\pm 10V$.

Embora o conversor A/D utilizado tenha uma resolução de 16 *bits*, 1 *bit* é de sinal e 4 *bits* de erro. Portanto, na melhor das hipóteses a resolução de uma tensão medida seria de 11 *bits*. Porém, testando-se esse conversor na prática observou-se que essa precisão não passava de 10 *bits*, o que permite diferenciar tensões de até 9,76 mV na entrada do conversor.

A razão de não se utilizar diretamente um conversor A/D de 8 canais de entrada, deve-se ao fato, que em geral os conversores disponíveis no mercado possuem uma resolução bem menor do que o conversor utilizado nesse projeto. E mesmo os que se aproximam dessa resolução, possuem a desvantagem do custo elevado.

Para o sistema de aquisição descrito anteriormente, o circuito *multiplex* de 8 canais tem a função de selecionar o canal de entrada a ser digitalizado, sendo o microcontrolador o responsável por controlá-lo através de seus pinos de seleção (A, B e C).

Portanto, o microcontrolador utilizado nesse projeto tem três funções principais, ou sejam: selecionar o canal a ser digitalizado, através do circuito *multiplex*; fazer o acionamento do conversor A/D; e transmitir os dados coletados ao computador servidor.

A transmissão dos dados é feita via protocolo RS-232 ao computador servidor, sendo esse último responsável por fazer o encapsulamento dos dados via TCP/IP, e posterior envio destes aos clientes conectados.

Todos os microcontroladores utilizados nesse projeto consistem do AT89S52, que é fabricado pela ATMEL, e tem como principal vantagem o baixo custo, principalmente em relação a alguns PIC's, e a uma grande parte dos microcontroladores disponíveis no mercado.

A razão de não se utilizar PIC's no projeto deve-se primeiramente ao fato de que os que possuem conversor A/D de alta resolução possuem custo muito elevado. Outro aspecto deve-se ao fato de que o *hardware* do painel eletrônico necessita de uma RAM externa de grande capacidade de armazenamento (pelo menos 32k bytes), e os PIC's em geral possuem uma RAM interna muito pequena.

Mesmo se fosse utilizada uma RAM externa conectada ao PIC, esta teria que se comunicar com ele via protocolo I2C, que é muito mais lento do que utilizando o acesso paralelo convencional, inviabilizando assim seu uso no *hardware* dos painéis eletrônicos.

Outra alternativa poderia ser a utilização do MSP-430 da *Texas Instruments*, o qual possui algumas versões com custo inferior ao AT89S52. Entretanto, um dos problemas desse microcontrolador é que além dele ser SMD, ele trabalha com tensão de alimentação de 3,6V, o que faz com que todos os circuitos digitais utilizados no sistema de aquisição necessitem trabalhar na mesma faixa de tensão.

Assim, o microcontrolador encontrado que possui o melhor custo benefício para esse projeto é o AT89S52-PC24. Além de ser de baixo custo, ele permite gravação ISP, dispensando assim o uso de gravadores universais.

Se fosse necessário fazer uma maior integração dos componentes do sistema de aquisição, ou se o sistema fosse produzido em escala industrial, poderia se optar pelo MSP-430, mas em se tratando de um protótipo, o AT89S52 supre bem os requisitos necessários para o funcionamento do sistema.

As principais características do AT89S52-PC24 são:

- Frequência de operação de até 24 MHz.
- RAM interna de 256 *bytes*.
- 3 contadores/temporizadores de 16 *bits*.
- Faixa de operação de 4,5 a 5,5V.
- 8K *Bytes* de Memória de Programa do tipo *Flash*.

Uma importante característica desse microcontrolador é a presença de 3 contadores/temporizadores de 16 *bits*, os quais são necessários para ajustar a taxa de transmissão serial dos dados (*baud-rate*) e para contar o número de pulsos gerados pelo circuito de aquisição da frequência da rede, como será visto na Seção 3.3.4.

Desses três contadores/temporizadores, um deles fica disponível para uso geral caso seja necessário monitorar uma nova entrada pulsada.

O *baud-rate* adotado nesse projeto foi de 57600 *bits*/segundo, que é a maior taxa de envio/recepção de dados que pode ser configurada para esse microcontrolador. Para obter essa taxa foi utilizado um cristal de frequência igual à 21.45666MHz.

A máxima frequência de operação permitida para o AT89S52-24PC é de 24MHz, mas a razão de não se utilizar um cristal com essa frequência, deve-se ao

fato, de não se conseguir ajustar um *baud-rate* de 57600 *bits/s* utilizando-se desse cristal. Portanto, o cristal mais rápido no qual se consegue ajustar essa taxa de transmissão é o cristal de 21,45666MHz.

Voltando ao sistema de aquisição ilustrado na Figura 3.4, observa-se que não está incluída nas entradas analógicas do sistema a saída do circuito de aquisição e condicionamento de sinal da frequência da rede.

Como mencionado anteriormente, o valor desse parâmetro é obtido a partir de um circuito convenientemente projetado para gerar pulsos diretamente proporcionais ao valor da respectiva medida. Sendo, a saída desse circuito conectada diretamente a uma das entradas pulsadas do microcontrolador. Utilizando-se dessa metodologia, determina-se com uma maior praticidade a frequência da rede elétrica.

Assim, para a aquisição da frequência da rede, foi projetado um circuito capaz de gerar um pulso de tensão a cada ciclo de onda da rede. Portanto, se a frequência da rede for igual a 60 Hz, serão então gerados 60 pulsos de tensão, no intervalo de 1 segundo, na saída de seu correspondente circuito de aquisição.

Pela definição de frequência de um sinal como sendo o número de ciclos deste em 1s, tem-se que para se obter a frequência da rede, bastará contar o número de pulsos gerados pelo correspondente circuito de aquisição no intervalo de 1s, conforme enunciado na Equação (3.3).

$$\text{Frequência Rede (Hz)} = N_F |_{1s} \quad (3.3)$$

onde:

N_F – número de pulsos de tensão gerados, na saída do circuito de aquisição da frequência da rede.

Como o microcontrolador necessita contar exatamente 1s, poderiam ser utilizadas duas alternativas. Uma delas é a de tentar contar esse intervalo de tempo através do *software* interno do microcontrolador, e a outra consiste em utilizar um oscilador externo de período igual à 1s.

Por razão de precisão de medida, preferiu-se utilizar um oscilador externo devido ao cristal utilizado ser de 21.45666MHz e, portanto, o ciclo de máquina do AT89S52 não seria múltiplo de 10, o que acarretaria em um erro maior na estimação desse intervalo.

Dessa forma, para se contar exatamente esse intervalo de tempo, foi projetado um circuito oscilador de período igual à 1s conectado ao pino de interrupção ($\overline{INT1}$) do microcontrolador.

Nesse caso, a cada 1s será gerada uma interrupção no programa principal do microcontrolador, fazendo com que este pare a execução normal de seu programa e atenda a interrupção solicitada.

Neste instante, o registrador do AT89S52 que armazena a contagem de pulsos conterá o valor real da frequência da rede, conforme enunciado pela Equação (3.3).

No diagrama em blocos ilustrado na Figura 3.5 é apresentado esse esquema descrito.

Figura 3.5. Sistema de aquisição – entrada pulsada.

Como no sistema supervisorio existem 9 parâmetros a serem supervisionados, 8 são obtidos a partir do conversor A/D, e 1 através das entradas pulsadas do microcontrolador.

Nas seções apresentadas a seguir serão descritas as metodologias adotadas para a aquisição de cada parâmetro sob supervisão do sistema e, no final dessa seção, será apresentado o circuito de aquisição final do sistema supervisorio desenvolvido.

3.3.1 Aquisição da Temperatura do Motor e Temperatura Ambiente

Para aquisição da temperatura do motor e da temperatura ambiente foram utilizados termistores, que são semicondutores cuja resistência possui alta sensibilidade à temperatura. O valor de sua resistência elétrica varia inversamente proporcional à variação de sua temperatura.

Na Figura 3.6 é ilustrada uma curva característica típica do termistor NTC-10k.

Figura 3.6. Curva característica do NTC-10k.

Como se pode observar por essa curva característica, o NTC-10k apresenta na temperatura de 25°C sua correspondente resistência nominal, ou seja, 10k Ω .

Entretanto, como os demais componentes do circuito, os termistores apresentam tolerância em relação ao seu valor nominal, o que torna difícil encontrar dois termistores iguais, possuindo assim, curvas características ligeiramente diferentes.

Assim, para se fazer a correspondência entre resistência e temperatura do termistor, é imprescindível levantar sua curva característica de forma a se diminuir o erro nessa estimativa.

Para o levantamento da curva característica de cada termistor, foram feitas medidas colocando-os em contato com superfícies quentes e frias na faixa de 0 a 100°C, e utilizando um instrumento adequado para medir temperatura, foram registradas as temperaturas da superfície, juntamente com a medição da resistência elétrica do termistor.

De posse desses dados, foram plotadas suas correspondentes curvas características, sendo gerado um polinômio por meio de extrapolação para cada termistor, obtendo assim, uma aproximação da temperatura da superfície em função da resistência do termistor.

A partir desse polinômio gerado, pode-se obter uma aproximação da temperatura da superfície desde que seja conhecido o valor de resistência do termistor.

O circuito para aquisição das temperaturas ambiente e do motor consiste de um divisor de tensão conectado a um amplificador de instrumentação, conforme ilustrado na Figura 3.7, onde:

$$k_T = 1 + \frac{R_3}{R_2} \quad (3.4)$$

Figura 3.7. Circuito para aquisição das temperaturas do motor e ambiente.

A tensão de interesse nesse caso é a tensão sobre o termistor, a qual é amplificada pelo amplificador de instrumentação, cujo ganho k_T é dado pela Equação (3.4). Nesse circuito, ao invés de utilizar um amplificador operacional convencional optou-se por empregar um amplificador de instrumentação, devido a esse último possuir uma alta impedância de entrada (da ordem de $10^{12} \Omega$) e assim praticamente eliminar o efeito de carregamento na tensão a ser medida.

Como a resistência do termistor varia com sua correspondente temperatura, a tensão sobre ele variará diretamente proporcional à variação de sua resistência. Utilizando a 1ª Lei de Ohm, obtém-se:

$$V_{Termistor} = R_{Termistor} \cdot I \quad (3.5)$$

$$I = \frac{V_F}{(R_1 + R_{Termistor})} \quad (3.6)$$

A tensão de saída do circuito (V_S) é dada pela Equação (3.7).

$$V_S = k_T \cdot V_{Termistor} \quad (3.7)$$

Portanto, uma variação na temperatura do termistor acarretará em uma correspondente variação na tensão de saída do circuito (V_S).

As saídas dos circuitos que medem as temperaturas do motor e ambiente (Figura 3.7), são conectadas às entradas do circuito multiplex, ilustrado na Figura 3.4.

Assim, essas tensões de saída serão selecionadas e digitalizadas uma de cada vez, de forma a se obter os parâmetros de interesse que nesse caso correspondem aos valores de temperatura.

Para se obter a temperatura de cada termistor, torna-se necessário determinar sua correspondente resistência e utilizar o polinômio característico gerado de forma a estimar sua correspondente temperatura.

Portanto, para determinar a relação da resistência do termistor em função da tensão de saída medida do circuito (V_S), deve-se substituir (3.6) em (3.5) e o resultado em (3.7). Isolando-se a variável $R_{Termistor}$ obtém-se a Equação (3.8).

$$R_{Termistor} = \frac{R_I}{\left[\frac{V_F}{V_S} \left(I + \frac{R_3}{R_2} \right) - I \right]} \quad (3.8)$$

Como V_F , R_1 , R_2 e R_3 são constantes conhecidas, as duas únicas variáveis na equação acima são a resistência do termistor ($R_{Termistor}$) e a correspondente tensão de saída do circuito (V_S).

Assim, medindo-se a tensão de saída do circuito (V_S), determina-se a correspondente resistência do termistor através da Equação (3.8). Substituindo esse

valor na equação do polinômio característico gerado, obtém-se sua correspondente temperatura.

Os valores das constantes V_F , R_1 , R_2 e R_3 (Figura 3.7) para os dois circuitos de aquisição de temperatura são os mesmos, sendo que a única diferença entre eles deve-se ao fato que para cada termistor haverá um correspondente polinômio característico.

3.3.2 Aquisição da Corrente e Tensão de Armadura do Motor

Para aquisição da corrente e tensão de armadura do motor, foi projetado o circuito ilustrado na Figura 3.8, sendo k_1 e k_2 dados por:

$$k_1 = 1 + \frac{R_3}{R_2} \quad (3.9)$$

$$k_2 = -\frac{R_5}{R_4} \quad (3.10)$$

Figura 3.8. Circuito para aquisição da corrente e tensão do motor.

Nesse circuito, a corrente do motor é medida utilizando-se um resistor de resistência nominal igual a 1Ω em série com o motor DC, sendo então medida a

tensão sobre ele. Como os resistores são componentes idealmente lineares, a corrente do motor será numericamente igual à tensão sobre o resistor (1ª Lei de Ohm), para esse valor de resistência. Como todo resistor apresenta tolerância em relação ao seu valor nominal, é preciso então medir seu valor real e compensar essa variação no cálculo final da corrente.

O motor utilizado para testes do sistema de supervisão consiste de um motor DC de imã permanente de 12V e corrente nominal de 200mA sem carga.

Observa-se assim, que se o motor for alimentado com 12V, uma corrente de aproximadamente 200mA percorreria o circuito, e seria gerada uma tensão de 0,2V sobre o resistor de 1Ω . Como a máxima tensão de entrada permitida para o conversor A/D é de 10V, perder-se-ia resolução se fosse utilizada diretamente a tensão lida sobre esse resistor.

Assim, é necessário amplificar esse valor de tensão de forma que a máxima tensão sobre o resistor aproxime-se da máxima tensão de entrada do conversor A/D, a qual é de 10V.

Portanto, conforme ilustrado na Figura 3.8 foi utilizado um amplificador não inversor para amplificar esse valor de tensão. O ganho de tensão nesse caso é dado pela Equação (3.9).

Escolhendo-se valores convenientes para os resistores R_2 e R_3 , ajusta-se o ganho de tensão k_1 conforme o desejado.

Já para se fazer a aquisição da tensão do motor, foi necessário utilizar um amplificador inversor conectado aos seus terminais. Este amplificador é necessário não para amplificar a tensão medida sobre o motor, mas sim para reduzi-la, pois a máxima tensão de entrada permitida para o conversor A/D é de 10V. Assim, quando a tensão de alimentação do motor for igual a 12V, é necessário que a tensão de

entrada do conversor A/D seja igual a 10V, portanto, esse amplificador inversor deve possuir um ganho máximo de 0,833.

Escolhendo-se adequadamente os valores de R_4 e R_5 , ajusta-se o ganho k_2 dado pela Equação (3.10).

Como se pode verificar pelo circuito de aquisição da Figura 3.8, a posição do terra do circuito foi escolhida de forma que a tensão medida sobre o resistor de 1Ω seja positiva, enquanto que a tensão sobre o motor seja negativa.

As escolhas das topologias dos amplificadores operacionais foram feitas de modo que as tensões de saída dos amplificadores sejam sempre positivas. Além disso, para o caso da medida da tensão sobre o motor é necessário utilizar um amplificador inversor, já que o ganho mínimo do amplificador não-inversor é igual a 1, o que faz com que essa topologia não possa ser utilizada para se reduzir uma tensão, mas apenas para amplificá-la.

Para o caso da aquisição da corrente, é necessário que se amplifique o valor de tensão lida sobre o resistor de 1Ω . Nesse caso, a amplificação poderia ser feita tanto com um amplificador inversor quanto com um não-inversor. Como é preferível trabalhar-se com tensões positivas nas saídas dos circuitos de condicionamento de sinal, deve-se utilizar um amplificador não-inversor para realizar essa amplificação.

Essa metodologia adotada justifica assim a posição dos componentes do circuito e o tipo de amplificador utilizado para a aquisição da corrente e tensão do motor DC.

3.3.3 Aquisição da Tensão da Rede Elétrica

Para aquisição da tensão da rede elétrica foi projetado um retificador em ponte conforme ilustrado na Figura 3.9.

Figura 3.9. Circuito para aquisição da tensão da rede elétrica.

O objetivo é fazer com que a tensão AC da rede seja convertida em um nível DC com um pequeno *ripple*, podendo assim obter-se mais facilmente o parâmetro de interesse que consiste da tensão RMS da rede.

Para o circuito anterior, as formas de onda no ponto A, com a chave S aberta e fechada, são mostradas na Figura 3.10.

Figura 3.10. Formas de onda no ponto A com a chave S aberta e fechada.

Supondo inicialmente os 4 diodos ideais, ou seja, que não possuam tensão de polarização, pode-se analisar o circuito anterior mais facilmente, como será apresentado a seguir.

Quando o circuito está com a chave S fechada, ou seja, está com carga, o capacitor C carregar-se-á até atingir o valor de pico V_p da tensão de entrada sobre ele. Neste instante, os diodos $D1$ e $D3$ irão parar de conduzir, e o capacitor se descarregará através da resistência R .

A descarga continuará até o instante em que a tensão de entrada exceder novamente a tensão sobre o capacitor. Nesse momento, o capacitor começará a se carregar até atingir novamente a tensão de pico de entrada, e assim, todo o processo se repetirá.

A variação de tensão sobre o capacitor considerando-o totalmente carregado, e o valor mínimo durante a descarga é chamada de *ripple*. Para esse projeto é desejado que esse valor seja o menor possível, portanto, a constante de tempo RC deve ser muito maior do que o intervalo de tempo de descarga.

Como a tensão sobre o capacitor corresponde a uma das entradas do circuito *multiplex*, conforme ilustrado na Figura 3.4, o *ripple* estará também presente na leitura dessa tensão. Assim, o que é proposto é fazer com que o microcontrolador faça 10 amostragens em um intervalo de meio ciclo de onda, ou seja, uma amostra a cada aproximadamente 833us, e dentre essas 10 amostras, ele determine o valor máximo de tensão, o qual estaria bem próximo da tensão de pico sobre o capacitor.

Para uma maior precisão da medida, poder-se-ia aumentar o número de amostras neste intervalo de meio-ciclo de onda. Entretanto, utilizando-se 10 amostras consegue-se uma boa aproximação para esse valor.

O valor de pico determinado corresponderia então à tensão de pico do secundário do transformador subtraído de 2 vezes a tensão de condução dos diodos (V_{D0}), que inicialmente foram considerados ideais. Quando o capacitor está se carregando, durante o semiciclo positivo da tensão da rede, a tensão sobre o mesmo é igual à tensão do secundário do transformador subtraída da tensão de condução dos diodos D_1 e D_3 . Para o semiciclo negativo, essa tensão medida corresponderia à tensão do secundário subtraída das tensões de condução dos diodos D_2 e D_4 .

Assim, as tensões de condução dos diodos devem ser inicialmente medidas para serem subtraídas da tensão medida pelo conversor A/D, de forma a diminuir o erro na estimação da tensão de pico do secundário do transformador.

De posse desse valor, deve ser utilizada a relação de espiras entre o primário e o secundário do transformador, para assim, por meio da Equação (3.11), determinar-se a tensão de pico da rede elétrica.

$$\frac{V_P}{N_P} = \frac{V_S}{N_S} \quad (3.11)$$

onde:

N_P – Número de espiras do primário do transformador.

N_S – Número de espiras do secundário do transformador.

V_P – Tensão do primário do transformador [V].

V_S – Tensão do secundário do transformador [V].

Conhecendo-se o valor da tensão de pico da rede, e supondo-se que a forma de onda desta seja puramente senoidal, pode-se determinar a tensão RMS da rede pela Equação (3.12).

$$V_{Re\ de}(rms) = \frac{V_{Re\ de}(pico)}{\sqrt{2}} \quad (3.12)$$

A razão de se utilizar um retificador em ponte ao invés de um retificador de meia onda, se deve ao fato de que para este último, seria necessário utilizar um maior número de amostras para se determinar o pico de tensão sobre o capacitor com a mesma precisão, já que a descarga sobre o capacitor ocorreria durante um

ciclo de onda completo, e não durante meio-ciclo, como é o caso do retificador em ponte.

Supondo-se constante o valor do capacitor C e variando-se o valor da resistência R , se observará que quanto menor for o valor da resistência utilizada, maior será o valor do *ripple*. Entretanto, independentemente do valor *ripple*, a tensão de pico sobre o capacitor será sempre a mesma.

Isso possibilita utilizar o próprio regulador de tensão do circuito como carga, facilitando assim, a implementação do circuito de aquisição da tensão da rede.

Embora as cargas ligadas ao regulador de tensão possam variar, e isso acarretaria em uma variação no valor do *ripple*, a tensão de pico sobre o capacitor será sempre a mesma, independentemente da carga ligada. Assim, o algoritmo para encontrar o valor máximo da tensão sobre o capacitor durante meio-ciclo de onda permaneceria inalterado, obtendo dessa forma o valor da tensão de pico sobre o capacitor.

Utilizando-se dessa metodologia, consegue-se estimar de forma mais simples e prática a tensão RMS da rede elétrica.

3.3.4 Aquisição da Frequência da Rede Elétrica

Para aquisição da frequência da rede elétrica, foi projetado um circuito capaz de gerar um pulso de tensão a cada ciclo de onda da rede. Dessa forma, se a frequência da rede for igual a 60 Hz, por exemplo, seriam gerados 60 pulsos/segundo na saída desse circuito, sendo, portanto, contados pelo microcontrolador.

Para se contar exatamente 1s pelo AT89S52, foi utilizado um oscilador externo conectado ao seu pino de interrupção $\overline{INT1}$.

No instante de cada interrupção ter-se-ia armazenado no registrador interno do contador do AT89S52 o valor da frequência da rede, conforme enunciado pela Equação (3.3).

Na Figura 3.11 é ilustrado o circuito projetado para se gerar um pulso de tensão a cada ciclo da tensão da rede.

Figura 3.11. Circuito para obtenção da frequência da rede.

Como se pode verificar por essa figura, o circuito consiste de um diodo em série com um divisor de tensão, conectado na entrada de um *shift trigger*. O diodo deixa passar apenas a parte positiva do sinal, enquanto que o divisor de tensão regula o pico máximo de tensão na entrada do *shift trigger* em 5V.

Quando a tensão de entrada do *shift trigger* passa de determinado valor (V_{TH}), o *shift trigger* é acionado fazendo com que sua saída passe para nível lógico “1”. Dessa forma, a cada ciclo de onda da rede será gerado um correspondente pulso de tensão na saída do *shift trigger*.

De modo a facilitar a implementação, foi utilizado no lugar do *shift trigger* um simples inversor lógico, mais especificamente o 74HC04. A tabela verdade para

esse inversor e seu diagrama esquemático são ilustrados respectivamente na Tabela 3.2 e Figura 3.12.

Tabela 3.2. Tabela verdade para o inversor lógico.

Entrada Digital (E)	Saída Digital (S)
0	1
1	0

Figura 3.12. Inversor lógico.

Para que o integrado 74HC04 reconheça um nível lógico “1” na sua entrada, é necessário que essa tensão seja superior a 3,15V, podendo variar levemente esse valor dependendo do fabricante.

Assim, o objetivo é fazer com que essa tensão de limiar corresponda à tensão V_{TH} ilustrada na Figura 3.11. Entretanto, diferentemente do funcionamento do *shift trigger* convencional, a saída desse circuito sairia invertida em relação à mostrada na Figura 3.11.

Supondo que a tensão de reconhecimento de nível lógico “1” seja exatamente 3,15V, a tensão de saída do circuito permanecerá em nível lógico “1” enquanto a tensão de entrada for menor do que 3,15V. Para tensões de entrada superiores a esse valor de tensão, a tensão de saída passaria para nível lógico “0”.

Como o contador do AT89S52 foi programado para reconhecer subidas de borda, então o pulso de tensão seria contado na transição de tensões superiores a 3,15V para tensões inferiores a esta.

O circuito final e as formas de onda de saída desse circuito são ilustrados na Figura 3.13.

Figura 3.13. Circuito final para obtenção da freqüência da rede elétrica.

Nessa figura, a cada passagem de nível lógico “0” para nível lógico “1” na saída do 74HC04 (ponto C), seria contado um pulso pelo microcontrolador. Portanto, pela definição de freqüência de um sinal como sendo o número de ciclos deste em um intervalo de 1s, bastaria contar o número de pulsos gerados por esse circuito para se obter sua correspondente freqüência, conforme enunciado na Equação (3.3).

Utilizando-se dessa metodologia, consegue-se estimar a freqüência da rede elétrica com uma resolução de 1Hz.

3.3.5 Aquisição da Velocidade de Rotação do Eixo do Motor

Para a aquisição da velocidade de rotação do eixo do motor foi utilizada primeiramente uma hélice conectada ao seu eixo e um sensor óptico de modo a detectar a passagem das pás da hélice através do sensor, conforme ilustrado na Figura 3.14.

Figura 3.14. Representação da hélice conectada ao eixo do motor.

Nesse caso, foi utilizada uma das entradas pulsadas do AT89S52 de modo a contar os pulsos gerados pelo circuito conectado ao sensor óptico. Assim, como há uma interrupção gerada pelo oscilador externo a cada exatamente 1 segundo, bastaria contar o número de pulsos gerados nesse intervalo de tempo e dividir o resultado pelo número de pás da hélice para se determinar a velocidade de rotação do eixo do motor, conforme enunciado na Equação (3.13).

$$Rotação_{Motor} (rad / s) = \frac{N_c}{N_p} \quad (3.13)$$

onde:

N_c – número de pulsos contados pelo microcontrolador no intervalo de 1s.

N_p – número de pás da hélice.

O circuito utilizado para essa metodologia é ilustrado na Figura 3.15.

Figura 3.15. Circuito projetado para aquisição da velocidade de rotação do eixo do motor.

Entretanto, utilizando-se dessa metodologia, a velocidade de rotação do eixo do motor seria atualizada a cada 1s, o que na maioria das aplicações seria um tempo muito longo caso a velocidade de rotação do eixo do motor variasse rapidamente.

Uma das alternativas poderia ser diminuir o período do oscilador externo, de forma a se gerar uma interrupção em um menor intervalo de tempo, só que nesse caso perder-se-ia uma grande resolução na medida da rotação se fosse utilizado o mesmo número de pás na hélice, pois o número de pulsos gerados seria o mesmo, apenas o intervalo de amostragem é que seria menor.

Portanto, para aumentar a resolução dessa medida utilizando essa metodologia, seria necessário aumentar o número de pás na hélice e empregar um circuito de alta sensibilidade para detectar a passagem de cada pá pelo sensor.

Além disso, um maior número de pás acarretaria em uma maior proximidade entre elas (caso seja utilizado o mesmo raio), o que poderia fazer com que mais de uma pá da hélice passasse pelo feixe de luz infravermelha do sensor ao mesmo tempo, gerando assim, leituras incorretas de velocidade.

Dessa forma, para essa metodologia seria necessário utilizar um sensor óptico menor e mais sensível, o que necessitaria de um circuito mais complexo para a detecção da passagem das pás.

Como não é o enfoque desse trabalho desenvolver um detector de alta sensibilidade, optou-se por utilizar um tacogerador acoplado mecanicamente ao eixo do motor.

O tacogerador consiste de um gerador DC que gera uma tensão de armadura diretamente proporcional a sua velocidade de rotação, já que o campo magnético é provido por ímãs permanentes no estator. Nesse caso, a tensão de saída é um nível DC com um pequeno *ripple* e, portanto, basta utilizar um filtro passa-baixas de modo a deixar apenas passar a componente DC do sinal.

Assim, medindo-se a tensão gerada pelo gerador DC pode-se estimar a velocidade de rotação do eixo do motor de forma mais prática e precisa. A vantagem dessa metodologia em relação à utilização do sensor óptico deve-se ao fato de que a cada período de amostragem consegue-se obter uma estimação para a velocidade de rotação do eixo do motor. A única desvantagem é que o gerador DC é uma carga a mais conectada ao eixo do motor.

Embora para o sistema supervisório desenvolvido seja utilizado um motor pequeno, e assim o gerador não poderia ser considerado uma carga a ser desprezada, em uma aplicação real com um motor de grande potência, a carga introduzida pelo gerador DC seria muito inferior à carga solicitada ao motor e assim poderia ser desconsiderada.

Na Figura 3.16 é ilustrado o acoplamento mecânico entre o motor e o gerador DC utilizado.

Figura 3.16. Acoplamento mecânico entre o motor e o gerador DC.

A tensão induzida no gerador DC é dependente de três fatores que são: a velocidade de rotação do eixo do gerador; o valor do fluxo magnético por pólo; e o valor da constante de força contra eletromotriz [23]. Assim, a tensão induzida no gerador pode ser enunciada pela Equação (3.14).

$$E_A = k_e \cdot \phi_p \cdot \omega \quad (3.14)$$

onde:

E_A – Tensão induzida na armadura do gerador DC [V].

k_e – Constante de força contra eletromotriz $\left[\frac{\text{V.s}}{\text{Wb.rad}} \right]$.

ϕ_p – Fluxo Magnético por pólo do gerador DC [Wb].

ω – Velocidade de rotação do eixo do gerador $\left[\frac{\text{rad}}{\text{s}} \right]$.

Como o gerador DC utilizado é de ímã permanente, tem-se que o fluxo por pólo ϕ_p será constante. Dessa forma, como k_e depende apenas de fatores construtivos do motor, é comum o fabricante juntar as constantes k_e e ϕ_p em uma única constante, conforme enunciado na Equação (3.15).

$$E_A = K_e \cdot \omega \quad (3.15)$$

onde:

E_A – Tensão induzida na armadura do gerador DC [V].

K_e – Constante de força contra eletromotriz $\left[\frac{\text{V.s}}{\text{rad}} \right]$.

ω – Velocidade de rotação do eixo do gerador $\left[\frac{\text{rad}}{\text{s}} \right]$.

Essa relação mostra a linearidade entre a tensão induzida na armadura do gerador e a velocidade de rotação do seu eixo. Dessa forma, medindo-se a tensão induzida no gerador, determina-se através da Equação (3.15) a velocidade de rotação do seu eixo.

É importante salientar que essa linearidade é distorcida para velocidades muito altas devido ao efeito de saturação. Nesse caso, quando se chega em tal ponto, um grande incremento na velocidade pode resultar em um incremento insignificante na tensão induzida na armadura.

O circuito projetado para aquisição da velocidade de rotação do eixo do gerador é ilustrado na Figura 3.17, onde:

$$k_1 = 1 + \frac{R_3}{R_2} \quad (3.16)$$

Figura 3.17. Circuito para aquisição da velocidade de rotação do eixo do gerador.

Como se pode verificar pela Figura 3.17, o gerador é conectado a um filtro passa-baixas de modo a deixar passar apenas a componente DC do sinal. A tensão de saída do capacitor é então conectada à entrada de um amplificador operacional em configuração não inversora.

Dessa forma, a tensão sobre o capacitor é amplificada por um ganho k_1 , de modo que a máxima tensão de saída do circuito aproxime-se da máxima tensão de entrada permitida para o conversor A/D, não perdendo assim resolução na medida dessa tensão.

O ganho k_1 do circuito é dado pela Equação (3.16) e deve ser ajustado de acordo com a máxima velocidade de rotação do eixo do gerador.

Como a relação entre a tensão induzida na armadura do gerador e a velocidade de rotação do mesmo é linear (Equação (3.15)), foram feitas medidas variando-se a velocidade de rotação do eixo do gerador e anotada a correspondente tensão de saída do circuito.

De posse desses dados, foi confeccionado o gráfico da velocidade de rotação versus tensão de saída do circuito e, por meio de extrapolação, foi

determinada a equação de reta que mais se aproximou do comportamento real do gerador.

Dessa forma, para determinar a velocidade de rotação do eixo do gerador basta medir a tensão de saída do circuito da Figura 3.17, e substituir seu valor na equação de reta obtida. Como há um acoplamento mecânico entre o gerador e o motor DC, a velocidade de rotação do eixo do motor DC será igual à velocidade de rotação do eixo do gerador.

3.3.6 Aquisição dos Demais Parâmetros DC

Para aquisição dos demais parâmetros DC do sistema, tais como tensão de alimentação do sistema microprocessado e tensão DC externa, essas foram conectadas diretamente as entradas E_7 e E_8 do circuito *multiplex* ilustrado na Figura 3.4.

Este procedimento é válido desde que seus valores estejam dentro da faixa de tensão de entrada permitida para o conversor A/D utilizado, que nesse caso é de $\pm 10V$.

3.3.7 Circuito Final de Aquisição

Na Figura 3.18 é ilustrado o circuito de aquisição final para o sistema supervisorio desenvolvido.

Como se pode observar nessa figura, foi utilizado o amplificador de instrumentação TL071 como amplificador operacional.

A razão de se utilizar esse amplificador se deve a este possuir entrada J-FET, o que praticamente não acarreta efeito de carregamento na tensão a ser medida, já que sua resistência de entrada é da ordem de $10^{12}\Omega$.

Foram testados amplificadores operacionais convencionais, tais como o LM741 e o LM324, mas estes além de possuírem problemas de *offset*, possuem resistência de entrada muito pequena quando comparada com amplificadores de instrumentação.

Para a escolha do amplificador operacional, foi então feita a comparação entre os amplificadores de instrumentação TL071 e TL062, que são uns dos mais baratos e fáceis de se encontrar no mercado.

Por razão de desempenho e custo, optou-se pelo amplificador de instrumentação TL071, sendo então este utilizado como amplificador operacional padrão em todos os circuitos do sistema supervisor.

Para se ajustar a tensão DC externa do sistema, foi utilizado um potenciômetro em série com um resistor formando um simples divisor de tensão. Nesse caso, alterando-se o potenciômetro ajusta-se essa tensão DC conforme o desejado. Essa tensão de referência serve para calibrar o sistema de aquisição e para testes do sistema de supervisão de parâmetros.

Figura 3.18. Circuito de aquisição final projetado.

Na Figura 3.18, o único circuito cujo funcionamento ainda não foi detalhado é o oscilador externo, mostrado de forma ampliada na Figura 3.19.

Figura 3.19. Oscilador externo projetado.

O oscilador externo projetado é composto por um circuito oscilador de frequência igual a 32.768kHz conectado a dois contadores de 14 *bits* (CD 4060).

Como é necessário obter na saída desse circuito oscilador uma forma de onda quadrada de período igual a 1s, foi então utilizado um cristal de frequência 32.768kHz, e contadores de modo a dividir esse valor de frequência 32768 vezes, obtendo assim na sua saída uma forma de onda quadrada de frequência igual a 1 Hz.

Como não é possível fazer essa divisão utilizando um único CD4060, foram ligados em cascata dois CD4060, de modo que a saída Q7 do primeiro contador fosse conectada à entrada de *clock* do contador seguinte.

Nesse caso, na saída Q7 do primeiro contador ter-se-á uma forma de onda quadrada com frequência igual a :

$$Q7_{Contador_1} = \frac{32768kHz}{2^7} = 256Hz \quad (3.17)$$

Como a saída Q7 do primeiro contador corresponde à entrada de *clock* do segundo contador, ter-se-á na saída Q8 deste último contador uma forma de onda quadrada com frequência igual a :

O computador servidor comunica-se com o sistema de aquisição por meio de sua porta serial. Os únicos pinos utilizados do conector DB9 são os pinos 2, 3 e 5, que correspondem respectivamente aos pinos de recepção, transmissão e terra, da porta serial do computador.

A necessidade de utilização do *driver/receiver* MAX232 se deve ao fato que os níveis de tensão, presentes nos pinos da porta serial do computador, não são níveis de tensão TTL. Assim, é necessário utilizar esse integrado para se compatibilizar os níveis de tensão tanto do computador servidor para o microcontrolador como vice-versa.

3.4 Sistema de Controle

3.4.1 Controle da Velocidade de Rotação do Eixo do Motor DC

Nos últimos 50 anos, diversas metodologias inovadoras na área de controle foram criadas de modo a se resolver problemas cada vez mais complexos e atingir melhores desempenhos. Os sistemas de controle sofreram enormes mudanças após o advento dos computadores, que tem se tornado dominante em praticamente todas as áreas da engenharia. Embora um vasto número de controladores tenha sido introduzido, a grande maioria dos processos industrial é ainda controlada por meio de simples controladores PID [24].

Isso não se deve somente à sua simplicidade, mas também a garantia de um desempenho aceitável para uma ampla faixa de plantas industriais e seu uso (*tuning* dos parâmetros) bem conhecido pela maioria dos operadores industriais. Dessa forma, o controlador PID fornece um custo/benefício que é bem difícil de se superar em comparação a outros tipos de controladores [25].

Entretanto, devido a simplicidade na sua estrutura, os controladores PID são convenientes para se controlar processos de primeira e segunda ordem. Assim, para plantas de ordem superior ou que apresentem problemas de não linearidade, é preferível utilizar outros tipos de controladores.

Exemplo desses controladores são os que mesclam o PID com regras de inferência Fuzzy, os que utilizam redes neurais, os controladores dead-beat, dentre outros.

Em [26] é proposto um sistema de controle de velocidade para motores de indução utilizando redes neurais artificiais, sendo esse controlador de 2 graus de liberdade (2-DOF).

Trabalho semelhante a esse é apresentado em [27], entretanto, ao invés de utilizar redes neurais é proposto um controlador Fuzzy para se regular a velocidade de rotação do motor de indução.

Para o controle da velocidade de motores DC, é proposto em [28] um controlador Fuzzy implementado em hardware, de modo a contornar os problemas na obtenção dos parâmetros do sistema e em situações que requeiram modelos precisos para se projetar um bom controlador.

Modificações do PID convencional são apresentados em [29] e [30]. Em [29], é utilizado sistemas de inferência Fuzzy juntamente com um PID para calcular a trajetória de mísseis terrestres.

Um controlador similar é apresentado em [30], só que nesse caso o controlador Fuzzy-PID proposto é utilizado no braço mecânico de um robô, e segundo os autores produz um melhor desempenho do que o PID convencional, tratando melhor os problemas de incerteza com cargas variantes no tempo aplicadas no braço do robô.

Um dos principais problemas ao se projetar um controlador PID é fazer o *tuning* dos seus parâmetros. Em [31], J.G. Ziegler e N.B. Nichols propõem duas metodologias para se realizar o ajuste dos coeficientes do PID. Na primeira, leva-se em consideração à resposta ao degrau do sistema, projetando o PID de forma a possuir um *damping* (ε) de 0.21, o que corresponde a uma taxa de decaimento de $\frac{1}{4}$. Segundo os autores esse valor corresponde a um compromisso razoável entre rapidez de resposta e margem de estabilidade.

Para a segunda metodologia desse trabalho, é utilizado o sistema em malha fechada com um controlador do tipo proporcional, e é aumentado o ganho deste até que o sistema comece a oscilar. Nesse instante são anotados o ganho crítico do controlador e o correspondente período de oscilação do sistema. De posse desses dois valores, determina-se através de tabelas os coeficientes do PID.

Outras metodologias para se fazer o *tuning* do PID são propostas em [25] e [32]. Nessas duas metodologias são propostos sistemas de inferência Fuzzy para realizar o *tuning* do PID, melhorando assim segundo os autores, o método apresentado por Ziegler-Nichols [31].

Como o sistema nesse projeto consiste de um motor DC e o enfoque desse trabalho não é fazer um sistema de controle de alto desempenho, optou-se por utilizar um PID convencional, o qual é bastante adequado para o controle de motores DC, já que esse sistema pode ser modelado por equações de primeira ou segunda ordem.

Para realizar o *tuning* do PID, optou-se pelo método proposto por Ziegler-Nichols pela sua grande simplicidade em se obter os parâmetros do controlador.

Segundo [33], os sistemas de controle podem ser classificados como sendo de dois tipos que são:

Sistemas Reguladores: A saída do sistema deve ser mantida tão próxima quanto possível de um valor constante desejado, mesmo na presença de perturbações.

Servomecanismos: Uma entrada variável no tempo deve ser seguida pela saída, o mais próximo quanto possível.

Segundo essa classificação, o motor DC utilizado nesse projeto constitui de um servomecanismo, pois a saída do sistema (velocidade de rotação do eixo do motor) deve seguir a entrada de referência, que corresponde à rotação desejada pelo operador do sistema.

Para se evitar problemas na obtenção das constantes do sistema, como mencionado em [28], é proposta a utilização do aplicativo MATLAB da Mathworks®, o qual através da ferramenta “ident” pode-se calcular a função de transferência que mais se aproxima do comportamento real do sistema. Nesse caso, é necessário primeiramente fazer um ensaio ao degrau do sistema, de modo a se utilizar os dados coletados como entrada no *toolbox* do MATLAB.

Entretanto, para tratar a função de transferência do motor DC é necessário primeiramente apresentar sua modelagem dinâmica. Na Figura 3.21 é ilustrado o diagrama eletromecânico do motor DC.

Figura 3.21. Diagrama eletromecânico do motor DC.

onde :

R_a – Resistência de armadura do motor [Ω].

L_a – Indutância de armadura do motor [H].

V_a – Tensão de armadura do motor [V].

K_e – Constante de força contra-eletromotriz [V.s/rad].

K_t – Constante de torque elétrico do motor [N.m/A].

ω – Velocidade de rotação do eixo do motor [rad/s].

J – Momento de Inércia do Motor [N.m.s²/rad].

B – Coeficiente de atrito viscoso [N.m.s/rad].

F – Coeficiente de atrito estático [N.m].

Por intermédio do diagrama eletromecânico mostrado na Figura 3.21, obtém-se o seguinte equacionamento para o sistema:

$$V_a(t) = R_a \cdot i_a(t) + L_a \cdot \frac{\partial}{\partial t} i_a(t) + K_e \cdot \varpi(t) \quad (3.19)$$

$$T_e(t) = K_t \cdot i_a(t) \quad (3.20)$$

$$T_R(t) = T_e(t) = J \cdot \frac{\partial}{\partial t} \varpi(t) + B \cdot \varpi(t) + F \quad (3.21)$$

Como o sistema em questão é um motor DC, em [24] foram calculadas as funções de transferência de primeira e segunda ordem para um motor DC genérico. Essas funções de transferências correspondem às Equações (3.22) e (3.23), respectivamente.

$$G_1(s) = \frac{\frac{K_e}{\tau_m}}{\left(s + \frac{1}{\tau_m}\right)} \quad (3.22)$$

$$G_2(s) = \frac{\frac{1}{K_e \cdot \tau_m \cdot \tau_a}}{\left[s^2 + \left(\frac{1}{\tau_a} + \frac{1}{\tau_b}\right)s + \left(\frac{1}{\tau_m \cdot \tau_a} + \frac{1}{\tau_a \cdot \tau_b}\right)\right]} \quad (3.23)$$

onde:

$$\tau_a = \frac{L_a}{R_a} \quad (3.24)$$

$$\tau_b = \frac{J}{B} \quad (3.25)$$

$$\tau_m = \frac{R_a \cdot J}{(K_e \cdot K_t)} \quad (3.26)$$

Na modelagem de primeira ordem é desprezado o efeito da indutância de armadura do motor. Para esse projeto, foi adotada a equação de segunda ordem (Equação (3.23)), pois corresponde a um modelo mais próximo do comportamento real do sistema.

Para a ferramenta “ident” do programa MATLAB, não é necessário conhecer os parâmetros do motor para que este faça a identificação da função de transferência. Basta que o usuário conheça o número de pólos e zeros do sistema, de modo que o programa determina por meio de extrapolação a função de transferência que mais se aproxima do comportamento do sistema. Nesse caso, devem ser entrados no *toolbox* os valores obtidos do ensaio ao degrau do sistema,

que são: dados de entrada (tensão de armadura do motor); dados de saída (velocidade de rotação do eixo motor); e o correspondente período de amostragem do sistema, que para esse projeto é de 25ms.

Após esse procedimento, o programa retorna a função de transferência que mais se adequa ao sistema em questão. Dessa forma, não é necessário medir as constantes do motor individualmente, o que representa uma grande vantagem, pois praticamente elimina erros na medição dos mesmos.

De posse da função de transferência obtida, foi projetado o controlador PID utilizando a metodologia proposta por Ziegler-Nichols [31], de modo a realizar o *tuning* das constantes do PID.

Como o controlador utilizado é digital, em [34] é proposta uma condição prática para se escolher o máximo tempo de amostragem do sistema. Nessa metodologia, é proposto que o tempo de amostragem deve ser pelo menos 6 vezes menor que o tempo de subida durante a resposta ao degrau do sistema.

Como essa condição é satisfeita para o tempo de amostragem adotado ($T_a = 25\text{ms}$), resta examinar a segunda condição que sugere que o conversor A/D utilizado tenha pelo menos uma precisão de 10 a 12 bits, já que o processo de quantização adiciona um ruído extra ao sistema. Como o conversor A/D utilizado é de 16 bits, essa condição também é satisfeita.

Analisando as duas metodologias propostas por [31] para realizar o ajuste dos coeficientes do PID, optou-se pelo método da última sensibilidade. Nesse método é utilizado um controlador do tipo proporcional com ganho K_{CRIT} , e é aumentado seu ganho até que o sistema comece a oscilar. Esse esquema proposto é ilustrado na Figura 3.22.

Figura 3.22. Método da última sensibilidade.

No instante em que o sistema entra em oscilação, são anotados o ganho crítico (K_{CRIT}) e o correspondente período crítico de oscilação do sistema (T_{CRIT}).

Para o sistema supervisorio desenvolvido, o controlador foi implementado no próprio programa do servidor. Nesse caso, o programa calcula a tensão a ser ajustada no motor e envia o valor calculado ao sistema de ajuste de tensão, via porta serial. Ao receber o valor de tensão a ser ajustado, o microcontrolador que controla o conversor D/A ajusta na saída do seu *driver* de corrente o valor requerido pelo servidor.

Dessa forma, a cada período de amostragem é calculada a ação de controle e enviada ao sistema de ajuste de tensão. Essa metodologia simplifica a implementação do controlador, pois o cálculo da equação do PID é efetuado pelo computador servidor e não pelo microcontrolador.

Como é utilizado um controlador discreto, torna-se necessário utilizar a forma discreta do equacionamento do PID. A equação de recorrência desse controlador é mostrada na Equação (3.27).

$$u(k) = u(k-1) + K \left(1 + \frac{T_D}{T_0} \right) \cdot [\varpi_{ref}(k) - \varpi(k)] + K \left(\frac{T_0}{T_I} - 1 - \frac{2T_D}{T_0} \right) \cdot [\varpi_{ref}(k-1) - \varpi(k-1)] + K \left(\frac{T_D}{T_0} \right) \cdot [\varpi_{ref}(k-2) - \varpi(k-2)] \quad (3.27)$$

onde :

u – Tensão de Controle [V].

ϖ – Velocidade de Rotação do Eixo do Motor [rad/s].

ω_{ref} – Velocidade de Rotação de Referência do Motor [rad/s].

T_0 – Período de amostragem [s].

T_I – Coeficiente de Integração [rad/V].

T_D – Coeficiente de Derivação [rad/V].

K – Fator de Amplificação [V/V]

Os valores dos coeficientes K , $\frac{T_0}{T_I}$ e $\frac{T_D}{T_0}$, em função de K_{CRIT} e T_{CRIT} medidos,

são dados pela Tabela 3.3.

Tabela 3.3. Valores tabelados para o PID discreto, utilizando o método proposto por Ziegler-Nichols.

Coeficiente	Valor
K	$0,6 \left(1 - \frac{T_0}{T_{CRIT}} \right)$
$\frac{T_0}{T_I}$	$\frac{1,2 \cdot K_{CRIT} \cdot T_0}{K \cdot T_{CRIT}}$
$\frac{T_D}{T_0}$	$\frac{3 \cdot K_{CRIT} \cdot T_0}{40 \cdot K \cdot T_{CRIT}}$

Portanto, após a determinação das constantes K_{CRIT} e T_{CRIT} do sistema, basta calcular os coeficientes do PID utilizando-se da Tabela 3.3, e substituir seus valores na Equação (3.27).

As variáveis $\omega(k)$, $\omega_{ref}(k)$ e $u(k)$ dessa equação, são os valores coletados pelo sistema de aquisição no instante $t=k \cdot T_0$, e representam respectivamente a velocidade de rotação do eixo do motor, a velocidade de rotação de referência ajustada pelo cliente, e a tensão de controle.

Assim, a cada período de amostragem é calculada a tensão a ser ajustada no motor e enviada ao sistema de ajuste de tensão. Dessa forma, consegue-se fazer com que a velocidade de rotação do motor siga a velocidade de rotação de referência, a qual é definida pelo usuário. Como sempre existe na prática um erro na determinação exata dos parâmetros do PID, efetuou-se um pequeno ajuste manual nos parâmetros do PID, de maneira que este apresentasse uma taxa de decaimento de $\frac{1}{4}$, conforme a proposta inicial do método de Ziegler-Nichols.

3.4.2 Circuito do Sistema de Ajuste de Tensão

O circuito do sistema de ajuste de tensão é composto por um microcontrolador conectado a um conversor D/A, e este último a um *driver* de corrente convenientemente projetado.

O conversor D/A utilizado no projeto é o DAC715 da *Texas Instruments*, e tem como características principais:

- Resolução de 16 bits.
- Tensão de saída de 0 a 10V.
- Ganho e *offset* ajustáveis através de potenciômetros de compensação.

Como o motor DC utilizado é de 12V e a máxima tensão de saída do conversor D/A é 10V, se torna necessário utilizar um amplificador operacional na saída desse conversor, de forma a amplificar esse valor de tensão, e se obter na saída do amplificador uma faixa de variação de tensão de 0 a 12V.

A comunicação entre o sistema de ajuste de tensão e o computador servidor é feita via protocolo RS-232, assim, é utilizado novamente o integrado MAX 232 para

compatibilizar os sinais TTL do microcontrolador com a porta serial do computador servidor.

Na Figura 3.23 é ilustrado o circuito do sistema de ajuste de tensão desenvolvido para o sistema supervisorio.

Figura 3.23. Circuito do sistema de ajuste de tensão.

Como se pode visualizar nessa figura, na saída do amplificador operacional utilizado (TL071) é conectado um *driver* de corrente, de modo a fornecer corrente suficiente para o acionamento do motor.

O *driver* projetado consiste de um amplificador classe A, e é composto por um circuito *Darlington* em topologia coletor-comum. O ganho deste *driver* é aproximadamente igual a 0,9 V/V.

Dessa forma, o ganho do amplificador de instrumentação (TL071) foi ajustado de modo a amplificar a tensão de saída do conversor D/A e ao mesmo tempo compensar a redução do ganho devido ao *driver* de corrente.

Para uma tensão de saída do conversor D/A igual à 10V, deve-se ter na saída do *driver* de corrente uma tensão igual à 12V, conforme o projetado.

A utilização desse circuito de ajuste de tensão só é válida para motores DC de baixa potência. Para motores DC de alta potência não é possível utilizar amplificadores classe A devido à grande perda de potência nesse tipo de amplificador.

Uma solução para motores DC de alta potência seria substituir o amplificador classe A utilizado por um amplificador chaveado. Nesse caso, a tensão de saída do conversor D/A poderia ser utilizada para ser a tensão de referência do amplificador chaveado, e assim regular a frequência dos pulsos de saída desse tipo de amplificador.

3.5 Sistema de Transmissão e Recepção de Dados via TCP/IP

3.5.1 Programa do Servidor

Após a obtenção dos parâmetros de interesse pelo sistema de aquisição, estes são enviados ao computador servidor, de modo a serem encapsulados via TCP/IP, e posteriormente transmitidos aos clientes conectados.

A comunicação entre o servidor e o sistema de aquisição de dados é feita via protocolo RS-232. Como se deseja enviar os dados com a maior taxa de transmissão possível, optou-se por uma taxa de transmissão de 57600 *bits/s*, que corresponde à maior taxa padrão que pode ser ajustada no microcontrolador utilizado.

Entretanto, a velocidade de transmissão dos dados não é função apenas do *baud-rate* ajustado, mas sim da velocidade com que o computador servidor consegue fazer uma leitura da sua porta serial.

Se fosse feito, por exemplo, uma sub-rotina no programa do servidor que fizesse uma leitura da sua porta serial o mais rápido possível, o processamento do computador servidor chegaria em poucos segundos ao seu máximo e este provavelmente interromperia totalmente sua execução.

Assim, o que é feito no programa do servidor é a utilização de *timers* pré-programados, que executam determinada função a cada determinado intervalo de tempo. Nesse caso, um *timer* seria utilizado para ler a porta serial do computador a cada intervalo regular específico.

O intervalo de tempo ajustado para esse *timer* foi feito de modo a não deixar o processamento do servidor muito sobrecarregado, pois além da aquisição dos dados, o servidor tem como função: gerenciamento das conexões dos clientes; acionamento do circuito de ajuste de tensão do motor; gerenciamento da câmera digital; e encapsulamento e envio dos dados e imagem do sistema via TCP/IP.

Assim, foi estabelecida por meio de muitos testes, a taxa de aquisição máxima de cada canal como sendo igual a 40 Hz. Como se tem um total de 9 parâmetros a serem supervisionados, o sistema de aquisição projetado tem uma capacidade máxima de 360 Hz.

Como quase todos os parâmetros supervisionados consistem praticamente de tensões DC, o sistema de aquisição projetado é suficiente para fazer a aquisição dos parâmetros a serem supervisionados, utilizando-se dessa taxa de amostragem.

O único parâmetro que poderia representar discrepância seria a corrente do motor, que além do nível DC, apresenta uma pequena variação em torno de seu valor médio, devido à comutação das escovas coletoras do motor.

A frequência desses pulsos depende da velocidade de rotação do eixo do motor, podendo chegar na faixa de alguns quilohertz para o motor utilizado.

Portanto, o sistema de aquisição projetado não é capaz de fazer a aquisição desses pequenos pulsos, pois pelo critério de *Nyquist*, a frequência de amostragem deve ser no mínimo igual a duas vezes a frequência máxima do sinal.

Entretanto, como no projeto é desejado apenas medir o valor médio da corrente, foram desprezados os pequenos pulsos gerados em torno de seu valor DC, utilizando para isso um filtro passa-baixas conectado à saída do amplificador não inversor ilustrado na Figura 3.8.

A partir do programa do servidor, é possível ajustar o número de pontos a serem enviados aos clientes conectados, diminuindo (se necessário) o processamento do computador servidor.

Assim, por exemplo, pode-se optar pelo envio de 20 pontos/segundo para cada parâmetro sob supervisão, ao invés do normal que é de 40 pontos/segundo.

Pelo programa do servidor é possível também regular o intervalo de envio dos blocos de dados, permitindo dessa forma ajustar a velocidade com que os parâmetros coletados são enviados aos clientes conectados.

Os dados coletados são enviados em blocos, de modo não seqüencial, conforme sua leitura pelo sistema de aquisição. Embora o envio seqüencial seja possível, esse causa problemas por congestionar rapidamente a rede caso algum dos pacotes seja perdido, fazendo com que aconteça uma realimentação positiva na rede.

Uma realimentação positiva é causada quando a rede está muito sobrecarregada e o roteador acaba perdendo pacotes. Nesse caso, como o protocolo TCP é um protocolo voltado à conexão e à transmissão confiável de dados, ocorrerá uma retransmissão dos pacotes perdidos, aumentando ainda mais o congestionamento da rede, o que a degradaria ao longo do tempo [35].

Portanto, o envio de um bloco maior de dados é preferível ao envio ponto a ponto dos dados coletados.

O tempo mínimo escolhido para a transmissão desses blocos foi de 250 ms. Tal valor foi baseado em testes funcionais do projeto em uma rede local.

O intervalo de envio pode ser ajustado para valores superiores a essa taxa mínima, caso a rede esteja muito sobrecarregada, ou de forma a diminuir o processamento do computador servidor, caso haja um grande número de clientes conectados.

Além dos dados, a imagem do sistema supervisionado também é obtida utilizando-se de uma câmera digital.

Pelo programa do servidor, é possível configurar a resolução da imagem, bem como parâmetros de brilho, contraste, dentre outros.

O tamanho em disco da imagem é função da resolução escolhida. Assim, uma maior resolução implicará em um maior número de *bytes* da imagem e, conseqüentemente, um maior intervalo de tempo para enviá-la ao cliente.

Assim, o que é proposto para a transmissão da imagem no programa do servidor, é permitir que o usuário possa ajustar o intervalo de envio da imagem ou optar pelo ajuste automático.

O ajuste automático de envio consiste em medir o intervalo de tempo que dado cliente leva para receber todos os *bytes* de uma imagem, e assim fazer o ajuste automático de sua taxa de transmissão. Nesse caso, se um cliente demora 500 ms, por exemplo, para receber a imagem, não se tem razão para enviá-la a cada 250 ms. Portanto, o programa tenta encontrar a taxa de envio que mais se enquadra nas taxas disponíveis no programa. Nesse caso, há uma taxa individual de

envio para cada cliente, que é obtida de acordo com o tempo que este demora a receber todos os *bytes* de uma imagem.

No caso do ajuste manual, a taxa de envio regulada seria a mesma para todos os clientes, independentemente do tempo que este leva para receber uma foto.

Nesse sistema, o número de clientes que podem acessar o servidor ao mesmo tempo é função do processamento do servidor e do tipo de serviço que é solicitado por cada cliente.

Assim, o sistema foi projetado de forma que os dados e imagens do processo apenas sejam enviados aos clientes que solicitarem o serviço, utilizando-se para isso de botões de requisição, conforme será visto nos resultados do programa do cliente.

Isso reduz o processamento do servidor, pois nem todo cliente que se conecta ao servidor precisa de todos os tipos de serviços disponíveis.

Outra implementação feita no programa do servidor foi o *chat* de comunicação, que permite que clientes conectados possam se comunicar entre si, bem como com o próprio servidor.

Todas as conversas entre clientes passam pelo servidor, que é responsável por fazer o gerenciamento das conexões.

Para um exemplo de 3 clientes conectados ao servidor, o gerenciamento das conexões poderia ser ilustrado conforme a Figura 3.24.

Figura 3.24. Diagrama das conexões de 3 clientes ao servidor.

Assumindo primeiramente que nenhum cliente está conectado, após a conexão do Cliente 1 ao servidor, este será adicionado à sua lista de usuários conectados.

Na conexão do Cliente 2, além da adição deste na lista de clientes conectados do servidor, é necessário enviar uma mensagem ao Cliente 1 informando que o Cliente 2 se conectou, e a este último a informação que o Cliente 1 já estava conectado. Dessa forma, cada cliente que se conecta sempre sabe quais são os demais usuários *on-line* e vice-versa, permitindo assim, a troca de mensagens e informações entre estes.

Processo análogo acontecerá quando um terceiro cliente se conectar (Cliente 3), ou seja, deverá ser enviada aos Clientes 1 e 2 a informação que o Cliente 3 se conectou, e ao Cliente 3 a informação que os Clientes 1 e 2 já estavam conectados.

Se após a entrada do Cliente 3, o Cliente 2 resolvesse desconectar, por exemplo, o servidor teria que enviar mensagens aos Clientes 1 e 3, informando que o Cliente 2 desconectou-se, sendo este último removido da lista de usuários *on-line* dos Clientes 1 e 3.

Dessa forma, o servidor faz o gerenciamento das conexões, permitindo que cada cliente sempre saiba quais são os demais usuários *on-line*.

3.5.2 Programa do Cliente

O programa do cliente tem três principais finalidades, ou seja, fazer a recepção via TCP/IP dos dados e imagens do sistema; fazer a exibição dos parâmetros monitorados e informações do sistema através dos painéis eletrônicos projetados; e enviar ações de controle ao servidor.

O acionamento dos painéis eletrônicos é feito utilizando-se as portas seriais do computador cliente, sendo necessário uma porta serial para cada painel.

Foram utilizados no projeto dois painéis eletrônicos, sendo um para exibição dos dados (painel tricolor) e um para exibição de informações de *status* do sistema (painel monocromático), tais como tempo de conexão, *delay* dos dados, latência da rede, horário, e mensagens informativas.

Tanto o programa do cliente quanto o programa do servidor permitem a visualização dos parâmetros supervisionados através de gráficos. No programa do servidor, conforme o dado for sendo coletado, este é plotado em tempo real em sua respectiva janela gráfica. Já para o caso do cliente, tem-se como mencionado anteriormente, a recepção dos dados em blocos a cada determinado intervalo de tempo. Assim, quando um determinado bloco é recebido pelo computador cliente, todos os dados desse bloco são plotados de uma só vez na sua janela gráfica.

Se, por exemplo, for ajustada no programa do servidor uma taxa de envio dos blocos igual à 250ms com uma taxa de amostragem de 40 pontos/segundos, ter-se-á no programa do cliente a recepção de 10 pontos a cada 250ms, de forma que a cada intervalo de 1s se tenha recebido os 40 pontos de cada parâmetro, conforme ajustado no programa do servidor.

Através da utilização de botões de requisição, o cliente pode fazer a solicitação dos serviços disponíveis do servidor, tais como:

- Serviço de Dados.
- Serviço de Vídeo.
- Serviço de Controle do Motor.

Os dois primeiros serviços foram explicados anteriormente, e se referem à solicitação ao servidor dos dados e imagem do sistema.

O serviço de controle serve para centrar no servidor o controle do motor a um único usuário (controlador). Após este ser travado, os demais clientes só poderão atuar no sistema, quando o controlador destravar o sistema.

Através do programa do cliente, o usuário pode regular a tensão de alimentação do motor ou ajustar a velocidade de rotação do seu eixo, enviando para isso um comando ao servidor contendo o tipo de controle desejado e seu valor. Ao receber esse comando, o servidor acionará o sistema de ajuste de tensão regulando o valor desejado pelo usuário.

Quanto ao *chat* visto pelo lado do cliente, toda vez que este se conecta ao servidor, este último faz o envio de sua lista de usuários *on-line*. Dessa forma, o cliente sempre sabe quais são os demais usuários *on-line*.

Além da exibição dos dados e parâmetros do sistema através de painéis eletrônicos, o programa do cliente permite também a edição de faixas de supervisão para cada parâmetro, conforme será descrito na seção a seguir.

3.6 Sistema de Supervisão de Parâmetros

Os programas do servidor e do cliente permitem a edição de faixas de supervisão para cada parâmetro monitorado do sistema.

No lado do cliente, o programa de supervisão tem o objetivo de verificar se os valores dos parâmetros coletados estão dentro de suas faixas de supervisão, pré-programadas. Assim, o usuário pode estabelecer limites de operação para cada parâmetro monitorado.

No lado do servidor, pode-se programar a mesma faixa de supervisão, entretanto, essa faixa tem a função de estabelecer limites máximos de operação do sistema, os quais se ultrapassado por muito tempo poderia causar algum tipo de dano ao próprio sistema. Assim, o monitoramento no lado do servidor tem a função de proteger o sistema, enquanto que no lado do cliente tem a função do seu supervisionamento.

Para a monitoração do sistema, foram definidos três tipos de faixas de supervisão, conforme ilustradas na Figura 3.25. Estas podem ser selecionadas e editadas através dos programas do servidor e do cliente, onde:

NI – Nível Inferior.

NS – Nível Superior.

NI1 – Nível Inferior 1.

NI2 – Nível Inferior 2.

NS1 – Nível Superior 1.

NS2 – Nível Superior 2.

Figura 3.25. Tipos de faixas de supervisão: (a) faixa de supervisão máxima; (b) faixa de supervisão mínima; (c) faixa de supervisão intermediária.

Para a faixa de supervisão mostrada na Figura 3.25(a), é monitorado o valor máximo de determinado parâmetro, sendo os patamares NS e NI programados pelo usuário. Tais patamares estabelecem o *status* do parâmetro de acordo com sua faixa de supervisão.

Se para esse tipo de faixa, o valor corrente do parâmetro estiver abaixo do nível inferior (NI) pré-programado, o *status* do parâmetro estará normal (área verde da faixa).

Se o valor do parâmetro estiver entre os patamares NI e NS (área amarela da faixa), o *status* do parâmetro indicará atenção, pois o parâmetro estará próximo de seu limite de operação pré-estabelecido.

No último caso, se esse valor estiver acima do nível NS (área vermelha da faixa), indicará que o determinado parâmetro ultrapassou sua faixa de supervisão permitida, indicando assim ao usuário, que uma ação de controle poderá ser necessária.

Para o tipo de faixa mostrada na Figura 3.25(b), é monitorado o valor mínimo do parâmetro. Nesse caso, se seu valor atual estiver acima do nível NS, o *status* do parâmetro estará normal, pois corresponde à área verde da faixa de supervisão.

Valores entre NI e NS indicarão atenção ao usuário, pois corresponde a área amarela da faixa, estando assim, próximo do limite de operação pré-estabelecido.

Para o último caso, se o valor corrente do parâmetro monitorado estiver abaixo do nível NI, este estará fora de sua faixa de supervisão e, assim, uma ação de controle poderá ser necessária ser efetuada pelo cliente.

O último tipo de faixa é mostrado na Figura 3.25(c), e corresponde à combinação das faixas mínima e máxima em uma só. Nesse caso, deseja-se que determinado parâmetro opere não somente acima ou abaixo de determinado valor, mas sim dentro de determinada faixa de variação do parâmetro.

Nesse caso, o *status* do parâmetro estará normal se seu valor atual estiver entre os níveis NI2 e NS1 (área verde da faixa), conforme mostrado na Figura 3.25(c).

Para valores entre NS2 e NS1 ou entre NI2 e NI1, ter-se-á o valor do parâmetro dentro da área amarela da faixa, indicando assim, que o determinado parâmetro estará próximo de seu limite de operação.

Finalmente, para valores acima de NS2 ou abaixo de NI1 (área vermelha da faixa), o valor do parâmetro terá ultrapassado sua faixa normal de operação.

Como será descrito na próxima seção, se o valor de determinado parâmetro ultrapassar sua faixa de operação máxima, tanto no lado do cliente como no lado do servidor, será enviada para o painel eletrônico de *status* a informação de que o dado parâmetro ultrapassou sua faixa de operação pré-estabelecida sendo, portanto, exibida no painel. Ao mesmo tempo em que essa informação for exibida no painel de *status*, no painel de dados será exibido o valor corrente desse parâmetro.

3.7 Sistema de Visualização de Parâmetros via Painel Eletrônico de Mensagens

Há décadas atrás, *leds* de alta intensidade começaram a entrar no mercado a um custo acessível. Isso levou muitos pesquisadores a começarem a utilizá-los como meio de transmitir e compartilhar informações.

No trabalho apresentado em [36], *leds* de alta intensidade foram utilizados para modular o sinal de semáforos com informações de áudio. Dessa forma, se um motorista tivesse em seu carro um receptor específico instalado, informações de trânsito ou de qualquer tipo poderiam ser transmitidas aos motoristas, de forma a alertá-los sobre acidentes ocorridos nas imediações ou indicando as melhores rotas a serem seguidas, auxiliando-os assim, na tomada de decisões.

Nos dias de hoje é comum a utilização de painéis eletrônicos nas rodovias, de forma a alertar os motoristas sobre acidentes, tráfego ou indicando rotas a serem seguidas pelos motoristas.

A utilização de painéis eletrônicos em um sistema supervisorio é a principal inovação proposta nesse trabalho. Além de dispensar o usuário do trabalho de ter que verificar de tempo em tempo os parâmetros do sistema na tela do computador, os painéis permitem que o sistema possa ser monitorado não apenas pelo usuário que possui o programa, como também, por usuários comuns que possam visualizar as mensagens exibidas no painel.

Dessa forma, o programa supervisorio pode exibir continuamente os parâmetros do sistema monitorado, e no caso de uma pane ou falha do sistema, o painel pode exibir frases de alerta, indicando as possíveis falhas ocorridas, ou requisitando uma ação de controle do usuário.

Com a utilização de efeitos visuais no painel, tais como o de fazer a mensagem piscar, ou aumentar a velocidade de exibição, pode-se alertar mais facilmente o usuário que determinada falha ocorreu no sistema.

Dessa forma, os dados e informações do sistema podem ser exibidos não somente de maneira estática, mas também utilizando efeitos visuais, tornando assim o monitoramento do sistema muito mais agradável de ser visualizado pelo usuário.

Portanto, os painéis eletrônicos podem ser empregados como uma ferramenta de auxílio na transmissão de informações, para as mais diversas áreas e aplicações.

Existem disponíveis no mercado diferentes tipos de painéis eletrônicos, cuja aplicação vai desde o emprego em bolsas de valores até a utilização em propagandas publicitárias. Dentre os principais tipos, pode-se destacar os painéis segmentados e os de matrizes de *led*.

A representação de dados em displays segmentados ou utilizando matrizes de *led*, tem sido o foco de muitas pesquisas [37-42]. Isso se deve ao fato de que nem todos os países possuem a grafia convencional, utilizando os caracteres alfanuméricos.

Tal representação depende da cultura do país, e não pode ser desprezada, já que países como a China, por exemplo, possui cerca de 1/5 da população mundial, bem como a Índia que possui mais de 900 milhões de habitantes. Os países do Oriente Médio são outros exemplos de nações que não possuem a grafia convencional. Assim, a escolha de um *display* que possa levar em consideração tais grafias é de suma importância.

Surakamponporn et. al. em [37] estudaram uma forma de representar os números em formato BCD para uma representação em números tailandeses utilizando-se de um display de 9 segmentos.

Para isso, foi montada uma tabela verdade com as entradas BCD, e anotadas as saídas desejadas do display segmentado.

Utilizando-se de lógica booleana e do teorema de Morgan foram obtidas as correspondentes saídas para os segmentos do display em função das entradas BCD.

Da mesma forma, Lau e Jawalekar em [38] utilizando-se de um display de 13 segmentos, projetaram um circuito digital capaz de fazer a conversão de números em formato BCD para números em formato hindu. Os procedimentos utilizados nesse conversor foram os mesmos utilizados em [37], ou seja, foi montada uma tabela verdade com as entradas em formato BCD e anotadas as correspondentes saídas desejadas dos segmentos do display. Utilizando-se novamente de lógica booleana, chegou-se a um circuito conversor BCD para o formato hindu.

Lakeou et. al. em [39] estudaram uma forma de representar números BCD em aramaicos. Entretanto, ao invés de utilizar displays do tipo segmentado, optou-se por fazer tal representação em uma matriz de *led* de 5x7, ou seja, a mesma que será utilizada no painel monocromático desse projeto.

Utilizando-se de portas lógicas e registradores, foi desenvolvido um sistema capaz de fazer essa conversão. Tal representação resultou em um circuito relativamente complexo e oneroso, pois envolvia uma grande quantidade de portas lógicas. E mesmo utilizando componentes de lógica programável, o qual possuem centenas de portas lógicas, o custo ainda seria relativamente elevado para a função

que esse circuito teria, ou seja, apenas fazer a conversão de números em formato BCD para a representação em aramaico.

Um dos principais problemas desses três tipos de implementações, é que estas requerem circuitos digitais relativamente grandes, os quais seriam praticamente inviáveis utilizando circuitos digitais discretos. Mesmo se a implementação fosse feita utilizando circuitos de lógica programável, o custo seria alto para uma simples função de conversão.

Outro ponto a se destacar é que se esses conversores fossem implementados em um sistema multi-numérico seria necessário um conversor para cada linguagem, o que inviabilizaria o sistema a baixo custo, a não ser que fosse produzido em escala industrial.

Uma solução encontrada para sistemas multi-numéricos foi apresentada por [40]. Nesse trabalho é apresentado um sistema que utiliza PROM's para solucionar esse problema. Para tal sistema, as entradas BCD são ligadas diretamente ao barramento de endereço das PROM's. Assim, para cada endereço na memória, ter-se-ia um correspondente dado armazenado, o qual seria diretamente ligado a um display de 16 segmentos. Dessa forma, selecionando-se a linguagem desejada através de uma chave, e aplicando-se as entradas BCD no barramento de endereço, obter-se-ia no display segmentado o número BCD previamente armazenado na linguagem desejada.

Este sistema é bem mais robusto que os demais por requerer apenas duas PROM's e por possibilitar a programação de qualquer grafia que possa ser representada em displays de 16 segmentos.

Embora todos os sistemas apresentados realizem a conversão de números BCD para a grafia apropriada, nenhum deles prevê a utilização de letras. Uma

solução para esse problema é apresentada por [41]. Nesse artigo é apresentada uma proposta para a exibição de caracteres arábicos através da utilização de matrizes de *led*.

Similarmente à [40], foi utilizado um sistema de ROM's, o qual armazena em sua memória todos os caracteres do alfabeto, e conforme a entrada aplicada obtém-se na matriz de *led* correspondente o mapeamento do caractere. Nesse caso, o caractere poderia ser tanto um número como uma letra.

Embora esse sistema apresente mais opções que os demais, pois poderia ser adaptado para mostrar caracteres em outras grafias, é necessário utilizar uma ROM de 64x8 para cada coluna da matriz de *led*. Assim, para se mostrar uma mensagem com 16 caracteres, onde cada caractere é formado por 5 colunas, seriam necessárias $5 \text{ (colunas/Matriz de Led)} \times 16 \text{ (Matriz de Led)} = 80 \text{ (colunas)} = 80 \text{ (ROM's)}$, o que teria um custo muito elevado, além de tornar o sistema de acionamento altamente complexo.

Uma forma de contornar esse problema foi apresentada por [42]. Utilizando-se de um sistema composto por EPROM's, *latches* e contadores em anel, foi montado um painel capaz de mostrar em quatro matrizes de *led* os dados previamente armazenados em código BCD. Através do contador em anel, é feita a varredura das linhas das matrizes de *led*. Assim, as linhas são amostradas a uma taxa tão alta que para o usuário torna imperceptível que estas são amostradas seqüencialmente, dando a impressão visual de que os números estão estáticos nas matrizes de *led*.

Esse sistema pode ser considerado o mais robusto de todos os apresentados, por ser constituído por um *hardware* mais simples. Entretanto, algumas das limitações desse projeto é que ele foi projetado apenas para exibir números nas

matrizes de *led*. Alterando-se o mapeamento de memória, como em [41], é possível adaptar esse sistema para mostrar letras, mas mesmo assim, tanto as letras como os números apareceriam estáticos nas matrizes de *led*.

Para se contornar esse problema, é proposto nesse projeto a utilização de um microcontrolador ao invés de EPROM's no controle das linhas e colunas das matrizes. Dessa forma, podem ser adicionados efeitos na exibição de mensagens, sendo, portanto, ainda mais robusto e possuindo um *hardware* mais simples, como será visto adiante, do que todas as metodologias mencionadas anteriormente.

3.7.1 Painel Eletrônico Monocromático

A metodologia utilizada nesse projeto, tanto para o painel eletrônico monocromático como para o tricolor aproxima-se da apresentada em [42], ou seja, utiliza a amostragem dos *leds* para a exibição das frases. Entretanto, ao invés de utilizar EPROM's foi utilizado um microcontrolador juntamente com registradores de deslocamento de modo a se fazer o acionamento das matrizes de *led*. Nesse caso, diferentemente da metodologia apresentada em [42], a qual faz a amostragem das linhas, é feita a amostragem das colunas, o que facilita seu acionamento conforme será descrito a seguir.

Nesse sistema, para se mostrar um único caractere no painel, é necessário amostrar coluna por coluna em cada varredura do display. Como existem 16 matrizes de *led* com cinco colunas cada, tem-se um total de 80 colunas a serem amostradas.

O diagrama esquemático de uma única matriz de *led* monocromática é mostrado na Figura 3.26.

Figura 3.26. Matriz de *led* catodo comum C-5570X.

As matrizes de *led* são construídas de modo a se utilizar o mínimo número de fios. Devido a essa razão suas colunas e linhas são multiplexadas e, portanto, não é possível acender os *leds* desejados de duas colunas ao mesmo tempo, caso seja adotada a amostragem por coluna. Assim, utilizando-se esse tipo de amostragem, as colunas da matriz devem ser acionadas uma de cada vez.

Para selecionar uma coluna da matriz de *led* do tipo catodo comum, é necessário colocar nível lógico “0” no pino correspondente da coluna, de modo a habilitá-la, e permitir que seus *leds* possam ser acesos. Para acender um *led* da coluna habilitada, basta colocar nível lógico “1” na linha correspondente.

Enquanto uma coluna está habilitada, as demais devem permanecer inibidas. Para isso, é necessário colocar nível lógico “1” nessas colunas.

Esse é o procedimento utilizado para o acionamento de uma única matriz de *led*. Para esse projeto que é composto por 16 matrizes, a idéia foi utilizar um banco de registradores de deslocamento em cascata, de modo que estes façam a seleção das colunas das matrizes, diminuindo assim, o número de pinos necessários para o seu acionamento pelo microcontrolador.

Nesse caso, aplicando-se nível lógico “0” na entrada do banco de registradores, e sucessivos pulsos de *clock* no mesmo, o *bit* de entrada deslocar-se-ia de saída em saída do banco, fazendo assim a seleção seqüencial das colunas das matrizes de *led*.

Tanto para as linhas como para as colunas das matrizes de *led*, é necessário adicionar *drivers* de corrente, pois nem o AT89S52, nem os registradores de deslocamento são capazes de fornecer corrente suficiente para que os *leds* acendam com intensidade suficiente, de modo a serem visualizados com clareza no painel.

Na Figura 3.27 é ilustrado o diagrama esquemático do *hardware* de acionamento para 3 matrizes de *led*. O circuito para as 16 matrizes seria praticamente o mesmo, apenas adicionando mais 13 matrizes nas linhas e aumentando-se o número de saídas do banco de registradores, adicionando-se então mais registradores em cascata.

Figura 3.27. Circuito de acionamento para 3 matrizes de *led*.

Pela figura anterior, para se selecionar determinada coluna do painel, basta dar um número suficiente de pulsos de clock no pino CLK do banco de registradores, selecionando assim a coluna desejada.

Para se acender os *leds* dessa coluna, basta escrever o *byte* desejado na porta P1 do AT89S52, fazendo assim acender os *leds* das linhas correspondentes.

Todas as linhas das matrizes de *led* são ligadas em paralelo, e dessa forma, para se fazer o acionamento das 16 matrizes, são necessários apenas 10 bits das portas do AT89S52, conforme se pode verificar pela Figura 3.27.

Utilizando-se dessa metodologia, para se escrever uma mensagem no painel, basta selecionar coluna por coluna e escrever seu correspondente *byte* na linha.

Para se produzir a ilusão de que todas as colunas estão acesas ao mesmo tempo, é necessário fazer o acionamento das 80 colunas de forma tão rápida que o olho humano não perceba a amostragem de cada coluna do painel.

Adotou-se assim a frequência de amostragem de 75Hz para o painel, de modo que este fosse visualizado com mais naturalidade.

Utilizando-se de um cristal de 21.45666 MHz no AT89S52, cada instrução de escrita em qualquer uma de suas portas levará cerca de 0,56us para ser executada. Assim, para se exibir uma frase no painel será necessário ter uma taxa de amostragem de cada coluna de no mínimo:

$$\text{Período Mínimo (s)} = 1/(80 \text{ colunas} \times 75 \text{ Hz}) = 1,67 \cdot 10^{-4} \text{s} \quad (3.28)$$

$$\text{Frequência mínima (Hz)} = 1/(\text{Período Mínimo}) = 6 \text{ kHz} \quad (3.29)$$

Conhecendo-se o tempo de escrita em cada porta do AT89S52, tem-se que esse microcontrolador é suficiente para controlar os registradores de deslocamento que farão a seleção das colunas das matrizes de *led*.

Entretanto, deve-se ressaltar que o programa de exibição de mensagens no painel possui diversas outras sub-rotinas que consomem tempo e, portanto, o tempo máximo que todo esse conjunto de sub-rotina deve levar, deverá ser inferior ao período mínimo de $167\mu s$.

Na Figura 3.28 é ilustrado um exemplo de seleção e escrita em uma das colunas do painel. A cada pulso de *clock* gerado no pino CLK do banco de registradores, habilita-se uma nova coluna, na qual podem ser acesos seus correspondentes *leds*. Tanto o controle do pino CLK como o de suas linhas são realizados pelo microcontrolador.

Figura 3.28. Seleção e escrita em uma das colunas do painel monocromático.

Aplicando-se sucessivos pulsos de *clock* na entrada CLK do banco de registradores, e escrevendo o correspondente valor na linha, consegue-se fazer com que o painel seja varrido coluna por coluna até se chegar na última coluna da última

matriz de *led*. Utilizando-se uma taxa de *clock* de no mínimo 6kHz, a varredura torna-se tão rápida que o olho humano não consegue perceber que as letras que aparecem nas matrizes são uma varredura contínua de suas colunas.

A vantagem de se utilizar um microcontrolador para se controlar os registradores de deslocamento ao invés de EPROM's como em [42], se deve ao fato de que no primeiro caso se pode fazer com que a mensagem seja exibida com efeitos dinâmicos, o que não pode ser realizado com a metodologia adotada utilizando EPROM's.

Através desse sistema, também se consegue exibir mensagens em qualquer grafia que possa ser representada em uma matriz de led de 5x7, bastando para isso, programar na memória do microcontrolador as colunas de cada caractere na grafia desejada. Dessa forma, esse sistema apresenta-se muito mais robusto e eficiente do que os apresentados anteriormente.

Outra grande vantagem desse sistema é o custo, que é inferior aos das demais metodologias, utilizando-se um mesmo número de matrizes de *led*. O custo do painel é função principalmente das matrizes de *led*, *drivers* de corrente e registradores de deslocamento.

Os *drivers* de corrente são necessários já que tanto o microcontrolador como os registradores de deslocamento não fornecem corrente suficiente para acionar as matrizes de *led*.

Portanto, para o acionamento das linhas das matrizes de *led*, foram projetados os *drivers* de corrente ilustrados na Figura 3.29.

Figura 3.29. *Driver* de acionamento das linhas do painel eletrônico monocromático.

Quando o programa do microcontrolador escrever, por exemplo, nível lógico “0” na porta P1.0, o transistor Q1 irá saturar, fazendo assim com que o *led* da linha 1 acenda se sua respectiva coluna estiver habilitada.

Se um nível lógico “1” na mesma porta for aplicado, o mesmo transistor entrará em corte, fazendo assim com que uma tensão próxima de 5V apareça entre seu coletor e emissor, não permitindo então que o *led* da linha 1 acenda, independentemente se a coluna está habilitada ou não.

Para se fazer o acionamento das colunas das matrizes de *led* é necessário também utilizar *drivers* de corrente. O circuito projetado para o acionamento das colunas das matrizes é ilustrado na Figura 3.30.

Figura 3.30. Circuito de acionamento das colunas do painel eletrônico monocromático.

Essa figura mostra o circuito de acionamento das colunas de uma única matriz de *led*. O circuito é composto por um registrador de deslocamento conectado a um *driver* de corrente. O *driver* utilizado consiste do ULN2803, que é um integrado composto por 8 circuitos *darlington* NPN independentes, enquanto que para o registrador de deslocamento foi utilizado o integrado 74LS164.

Nesse circuito, as saídas Q_i do 74LS164 são conectadas às entradas IN_i do ULN2803. As saídas desse último são ligadas diretamente nas colunas das matrizes de *led*.

O banco de registradores de deslocamento é obtido conectando-se a saída Q7 de cada registrador de deslocamento na entrada A do registrador seguinte.

Como cada matriz monocromática possui 5 colunas, restam 3 saídas do ULN2803 para cada matriz ligada. Nesse caso, as saídas que restam são ligadas nas colunas da matriz de *led* seguinte. Assim, para o primeiro caso se ligaria as 3 saídas que sobram do ULN2803 nas colunas 1, 2 e 3 da próxima matriz de *led*.

Como se tem 80 colunas para serem acionadas, e como os ULN2803 possuem 8 saídas cada, serão necessários dez ULN2803, bem como dez 74LS164 para se fazer o acionamento das 16 matrizes de *led*.

3.7.2 Painel Eletrônico Tricolor

O painel eletrônico tricolor projetado possui um funcionamento similar ao painel monocromático, com a diferença que ele exibe frases em três cores diferentes (vermelha, amarela e verde).

Cada ponto da matriz de *led* colorida é composto por dois tipos de *led* (verde e vermelho), sendo que a cor amarela é obtida ao se acender os *leds* de cor verde e vermelha ao mesmo tempo.

A matriz colorida utilizada nesse projeto consiste da matriz XMURG60C8, que é composta de 8 linhas x 8 colunas.

O diagrama esquemático dessa matriz é mostrado na Figura 3.31.

Figura 3.31. Diagrama esquemático da matriz colorida XMURG60C8.

Como pode ser visto nessa figura, as colunas e linhas dessa matriz são duplas, sendo uma para cada tipo de *led*. As linhas L_iR e colunas C_iR representam os *leds* de cor vermelha, enquanto que as Linhas L_iG e colunas C_iG representam os *leds* de cor verde.

Para facilitar o *hardware*, foram conectadas as colunas das duas cores em um mesmo ponto, de modo que ao se selecionar uma das colunas, estará se selecionando as colunas das duas cores ao mesmo tempo, tornando assim, o acionamento das colunas similar ao da matriz monocromática.

O diagrama esquemático de acionamento das colunas de uma única matriz de *led* colorida é mostrado na Figura 3.32.

Figura 3.32. Diagrama esquemático do acionamento das colunas de uma matriz de *led* colorida.

Como se pode observar nessa figura, o acionamento é similar ao da matriz monocromática, entretanto, o número de pinos de saída do ULN2803 é igual ao número de colunas da matriz de *led* colorida. Dessa forma, não restam pinos de saída desse *driver*, facilitando assim as ligações do *hardware* em relação ao painel monocromático.

Como as colunas das duas cores são ligadas ao mesmo ponto, é necessário separar as linhas; caso contrário, sempre as mensagens seriam mostradas na cor amarela.

No esquemático anterior, as linhas do tipo L_iG , correspondem às linhas dos *leds* de cor verde, enquanto que as linhas L_iR correspondem às linhas dos *leds* de cor vermelha.

As linhas L_iG e L_iR são conectadas aos *drivers* das linhas conforme será explicado a seguir.

A matriz de *led* colorida utilizada possui uma linha a mais do que a matriz de *led* monocromática e, além disso, o número de linhas é dobrado, pois as linhas das cores verde e vermelha são separadas.

Nesse caso é necessário utilizar 16 saídas do AT89S52 para se acessar as linhas das matrizes de *led*.

Como o microcontrolador utilizado não possui porta suficiente disponível, pois a ele é conectada uma RAM externa de 32kbytes, é necessário multiplexar uma de suas portas de 8 *bits*. Para isso são utilizados 2 *latch* tipo D, de forma a se duplicar a porta de saída P1. O *latch* escolhido foi o 74HC373, que é composto de oito *latches tri-state* tipo D.

O diagrama esquemático da Figura 3.33 ilustra a multiplexação da porta P1 do microcontrolador.

Figura 3.33. Multiplexação da porta P1 do AT89S52.

As linhas L_iG e L_iR ilustradas na figura anterior são conectadas às linhas das matrizes de *led* mostradas na Figura 3.32.

Para se escrever um *byte* na saída do *latch* U1, por exemplo, deixa-se o *latch* U2 inibido, e habilita-se o *latch* U1, de modo que todos os *bits* presentes na entrada desse *latch* sejam transferidos para sua correspondente saída. Feito isso, trava-se esse mesmo *latch*, de modo que em sua saída se mantenha os valores dos *bits* entrados até a chegada do próximo *byte* a ser escrito.

Assim, através do pino de habilitação de cada *latch* (pino $\overline{LE}$), seleciona-se o *latch* no qual se deseja escrever o *byte*, fazendo com que os *leds* da coluna habilitada possam acender conforme o *byte* de saída de cada *latch*.

No caso de se desejar acender determinado *led* na cor amarela, basta acender ao mesmo tempo os *leds* das cores vermelha e verde através dos *bits* de saídas dos seus respectivos *latches*.

Dessa forma, o *hardware* de acionamento das colunas da matriz de led tricolor será similar ao da matriz monocromática. A única diferença de *hardware* será no acionamento das linhas. Nesse caso, é necessário utilizar *latches* para duplicar a porta P1 e utilizar o dobro de transistores nas linhas, devido a essas serem separadas para cada tipo de *led*.

Na Figura 3.34 é ilustrado o diagrama simplificado para o acionamento de uma única matriz de led colorida.

Figura 3.34. Diagrama simplificado de acionamento de uma única matriz de led colorida.

Nessa figura as colunas C_i' correspondem às colunas C_iG e C_iR que são conectadas no mesmo ponto de modo a simplificar o hardware.

Tanto para o painel monocromático quanto para o painel tricolor, foi utilizada uma RAM externa de 32 kbytes de capacidade, de modo a armazenar as frases enviadas pelo computador cliente para posterior exibição.

Após a recepção de todos os caracteres de uma frase, o programa do microcontrolador monta em sua RAM externa o mapeamento das colunas de cada caractere, e em seguida, através de uma sub-rotina para cada tipo de efeito, faz a exibição da mensagem conforme o efeito selecionado pelo usuário.

O sistema de visualização de parâmetros via painel eletrônico é a principal inovação proposta neste sistema supervisorio.

Para a exibição dos parâmetros coletados nos painéis eletrônicos, basta o usuário conectá-los ao computador do cliente e habilitar a exibição das mensagens pelo programa, utilizando-se de um botão de requisição, conforme será mostrado nos resultados dessa seção.

Ao habilitar os painéis eletrônicos, os dados sob supervisão e informações de *status* do sistema serão exibidos automaticamente.

O computador cliente é o responsável por fazer o gerenciamento das mensagens nos painéis. A comunicação entre computador e painéis é feita via porta serial, ou seja, utilizando o protocolo RS-232.

Para a exibição das mensagens nos painéis, foi primeiramente feito um programa para apenas exibir mensagens digitadas em uma caixa de texto, sendo estas exibidas no painel escolhido pelo usuário. Após essa etapa, o programa foi adaptado para a exibição automática de parâmetros e informações de *status* do sistema, conforme a proposta do projeto.

Como mencionado anteriormente, o painel monocromático é responsável por fazer a exibição das mensagens de *status* do sistema. As informações que foram programadas para serem exibidas nesse painel foram:

- *Delay* dos dados recebidos.
- Latência da rede.
- Tempo de Conexão.
- Horário.
- Mensagens de supervisão.

O *delay* dos dados recebidos corresponde ao tempo médio que determinado cliente demora a receber um bloco de dados coletados.

A latência da rede corresponde ao tempo médio que um datagrama leva para ir do servidor ao cliente.

O tempo de conexão corresponde ao intervalo de tempo que o cliente está conectado no servidor monitorando o sistema.

As mensagens de supervisão são as frases exibidas no painel, que informam ao cliente que determinado parâmetro monitorado ultrapassou sua faixa de supervisão pré-programada, ou que determinado controlador assumiu o controle do sistema.

Como comentado anteriormente, se o valor de determinado parâmetro sair de sua faixa normal de operação, uma mensagem no painel de *status* será exibida, alertando o usuário da ocorrência desse evento.

No caso do parâmetro ultrapassar a faixa de supervisão no lado do servidor, este enviará aos clientes conectados, via TCP/IP, a mensagem que determinado

parâmetro saiu de suas condições normais de operação, e assim, cada cliente exibirá em seu respectivo painel de *status* a mensagem de alerta.

Outra funcionalidade adicionada foi permitir ao cliente escrever mensagens complementares a serem exibidas em qualquer um dos painéis eletrônicos. Esse procedimento é feito também no lado do servidor, permitindo que este também tenha acesso aos painéis eletrônicos. Nesse caso, as mensagens do servidor seriam difundidas como uma espécie de *broadcast* a todos os painéis dos clientes conectados.

Dessa forma, se o servidor desejar exibir uma mensagem adicional de alerta no painel de *status* dos clientes, bastará no seu próprio programa selecionar o painel que deseja escrever a mensagem, escolher o efeito e velocidade, e pressionar o botão de envio. Feito isso, a mensagem será enviada via TCP/IP a todos os clientes conectados, sendo, portanto, exibidas em seus respectivos painéis de *status*.

O painel de *status*, por apenas exibir mensagens informativas, foi projetado de modo a utilizar matrizes de *led* monocromáticas, ou seja, matrizes de uma única cor. Nesse caso foram escolhidas matrizes de *led* de cor vermelha.

O painel de dados foi projetado para utilizar matrizes de *led* coloridas. Nesse caso, a função das cores é a de representar o *status* do parâmetro monitorado de acordo com sua faixa de supervisão pré-estabelecida.

Se o parâmetro monitorado, por exemplo, for a tensão do motor e o tipo de faixa escolhida for a intermediária, conforme mostrada na Figura 3.25(c). Se fosse configurado os valores dos patamares mostrados nessa figura como NI1=4V, NI2=5V, NS1=6V e NS2=7V, o parâmetro seria mostrado na cor verde no painel, caso a tensão medida estivesse entre 5 e 6V (área verde da faixa), indicando assim que o *status* do parâmetro está normal.

Se esse valor estivesse entre 4 e 5V ou entre 6 e 7V (área amarela da faixa), o parâmetro seria mostrado na cor amarela, indicando atenção ao usuário, pois o valor do parâmetro está próximo do limite de operação pré-estabelecido.

No último caso, se o valor monitorado estivesse abaixo de 4V ou acima de 7V (área vermelha da faixa) o parâmetro seria mostrado na cor vermelha, indicando que o parâmetro saiu de sua faixa normal de operação.

A exibição de mensagens não é feita somente utilizando mensagens estáticas, mas também através de efeitos dinâmicos. O usuário do programa pode selecionar o tipo de efeito que deseja para cada parâmetro do sistema, bem como para cada informação de *status*.

Foram implementados 14 efeitos diferentes para cada painel, sendo estes listados a seguir:

- 1) Mensagem Parada.
- 2) Mensagem Piscando.
- 3) Mensagem Rolando de Cima para Baixo.
- 4) Mensagem Rolando de Baixo para Cima.
- 5) Mensagem Montada em Tiras para Baixo.
- 6) Mensagem Montada em Tiras para Cima.
- 7) Rolagem Horizontal da Esquerda para Direita.
- 8) Rolagem Horizontal da Direita para Esquerda.
- 9) Mensagem Dividida ao Meio.
- 10) Mensagem Exibida das Extremidades para o Meio.
- 11) Mensagem Exibida do Meio para as Extremidades.
- 12) Tiras Exibidas Aleatoriamente.

13) *Pacman*.

14) Faixa Horizontal de Fora para Dentro.

Na Figura 3.35 é ilustrada uma seqüência de imagens, representando o Efeito 3, ou seja, o efeito de Rolagem de Cima para Baixo da mensagem.

Figura 3.35. Representação do efeito de rolagem de cima para baixo de uma dada mensagem.

Os efeitos não foram implementados no programa do cliente, e sim no programa do microcontrolador, de forma que para se exibir uma mensagem com determinado efeito e velocidade, basta o computador cliente enviar via porta serial ao microcontrolador os *bytes* da frase, efeito e velocidade selecionados.

Ao receber essa cadeia de *bytes*, o AT89S52 monta em sua memória RAM os *bytes* das colunas de cada caractere da mensagem, para posterior exibição.

O *hardware* do AT89S52 foi projetado de forma a se mostrar não apenas frases curtas, mas permite que sejam exibidas frases longas de até 4000 caracteres, podendo assim ser utilizado até na exibição de pequenos textos.

A partir do programa do cliente, o usuário pode também selecionar para o painel tricolor, as cores das linhas das matrizes de *led*. Isso permite, por exemplo, que o usuário escreva uma mensagem com as duas primeiras linhas na cor vermelha, as duas seguintes na cor verde, e as restantes na cor amarela, dando assim um maior apelo visual à mensagem.

Por intermédio de tal programa é possível também configurar individualmente o efeito de cada parâmetro sob supervisão.

Todas as configurações feitas podem ser salvas em arquivo e restauradas conforme desejado. Esse procedimento também é feito para as faixas de supervisionamento pré-programadas pelo usuário, permitindo assim, que o usuário não tenha que defini-las novamente toda vez que executar o programa.

3.8 Interface de Entrada via Controle Remoto Infravermelho

Para esse projeto, foi desenvolvida uma interface de entrada via controle remoto infravermelho, de modo que o cliente possa configurar a tensão ou velocidade de rotação do eixo do motor DC diretamente no painel eletrônico de *status*, utilizando-se para isso, de um controle remoto infravermelho e um circuito receptor convenientemente projetado.

Como o sistema pode ser monitorado diretamente através dos painéis eletrônicos, uma interface de entrada que permita enviar ações de controle ao sistema, sem a necessidade de utilizar o teclado do computador é de suma importância, pois não se teria muito sentido fazer a monitoração do sistema através

de painéis, visto que para se para enviar as ações de controle é necessário utilizar o teclado do computador.

Assim, por intermédio dos painéis e dessa interface de entrada é possível o usuário monitorar e controlar o sistema sem a necessidade de utilizar diretamente o computador, flexibilizando assim o sistema supervisorio desenvolvido.

Para esse projeto foi utilizado um controle remoto de uma TV Sanyo, o qual utiliza o integrado LC7461M para transmitir o valor da tecla pressionada pelo usuário.

Como na maioria dos controles remotos infravermelhos, o código de cada tecla pressionada é enviado modulando-se o sinal digital com uma portadora na faixa de 30 a 60kHz.

A portadora utilizada pelo LC7461M é de 38kHz, o que corresponde a um período de 26,3 μ s.

O código de transmissão utilizado por esse integrado é parecido com o código de transmissão da NEC; entretanto, diferentemente deste último, o qual transmite 8 bits para identificação do dispositivo e 8 bits para identificação da tecla, o código da Sanyo envia 13 bits para identificação do dispositivo e 8 bits para a identificação da tecla.

Portanto, quando o usuário pressionar qualquer tecla desse controle, serão enviados: 13 bits de identificação do dispositivo, 8 bits de identificação da tecla e mais 21 bits que correspondem aos códigos invertidos desses dois números de identificação.

Na Figura 3.36 é mostrada a organização dos *bits* transmitidos pelo controle remoto infravermelho da Sanyo:

Figura 3.36. Organização dos *bits* transmitidos pelo controle infravermelho da Sanyo.

A transmissão dos códigos invertidos dos *bits* facilita a detecção de erros na recepção dos dados, o que corresponde a uma vantagem em relação a alguns códigos de transmissão, tais como o RC5 da Philips. Entretanto, este último tem a vantagem de se enviar apenas 14 *bits*, enquanto que o código da Sanyo envia um total de 42 *bits*.

Os bits de C₀ a C₁₂ (Figura 3.36) identificam o dispositivo que utiliza o controle remoto, tais como rádio, TV, DVD, etc. Dessa forma, o controle de uma TV Sanyo não interfere no DVD desse mesmo fabricante, pois o código de identificação para cada dispositivo é diferente.

Os bits de D₀ a D₇ identificam a tecla pressionada, assim, é possível se ter até 256 teclas diferentes utilizando-se o LC7461M.

A transmissão dos dados é iniciada enviando-se primeiramente um trem de pulsos (*burst*) de 9ms de duração.

Após isso, os *bits* são enviados segundo a organização mostrada na Figura 3.36. Na Figura 3.37 é mostrado o diagrama de tempo para os dados enviados pelo LC7461M.

Figura 3.37. Diagrama de tempo para os dados enviados pelo LC7461M.

Como se pode verificar por essa figura, a transmissão dos níveis lógicos “0” e “1” é feita através do envio de trens de pulso sucessivos, sendo o nível lógico “0” caracterizado por uma diferença de 1,125ms entre dois trens de pulsos sucessivos, e o nível lógico “1” caracterizado pelo dobro desse intervalo tempo, ou seja, 2,25ms.

A frequência do trem de pulsos corresponde à frequência da portadora, ou seja, 38kHz, conforme mostrado na Figura 3.38.

Figura 3.38. Forma de onda da portadora.

Para a recepção dos dados enviados foi utilizado o módulo receptor de infravermelho TSOP1238. Esse módulo pode ser utilizado não apenas para a recepção dos dados utilizando-se o código da Sanyo, mas também é compatível com os códigos da NEC, Toshiba, Sony, RC5, RC6, R-2000, dentre outros.

Na Figura 3.39 é ilustrado o módulo TSOP1238 utilizado nesse projeto.

Figura 3.39. Módulo receptor de infravermelho TSOP1238.

Para o circuito receptor foi utilizado o TSOP1238 conectado a um microcontrolador (AT89S52), sendo este último responsável por decodificar o sinal proveniente do sensor.

A saída desse módulo (Vout) pode ser conectada diretamente no pino de interrupção $\overline{\text{INT0}}$ do microcontrolador, conforme ilustrado na Figura 3.40.

Figura 3.40. Circuito receptor de infravermelho.

Por meio de um programa implementado no microcontrolador, o código recebido pelo sensor é decodificado, e dessa forma é determinada qual tecla foi pressionada.

Após essa identificação, é enviado via porta serial um caractere ASCII ao computador cliente, representando a tecla pressionada.

Assim, toda vez que uma tecla for pressionada, o programa do cliente receberá esse caractere e o exibirá através do painel eletrônico de *status*. Dessa forma, o usuário poderá ajustar diretamente no painel de *status* a velocidade de rotação do eixo do motor ou sua tensão de armadura, pois conforme os números forem sendo pressionados no controle remoto, esses serão exibidos no painel eletrônico.

Os caracteres especiais do controle remoto tais como “>”, “<”, “+”, “-“, etc, são utilizados para funções especiais, tais como: entrar nos modos de ajuste de tensão ou velocidade de rotação do eixo do motor; enviar os valores programados ao servidor; e correção dos valores entrados no painel.

Quando se pressiona a tecla para entrar no modo de ajuste de tensão do motor, é exibida uma frase no painel de *status* indicando a entrada nesse modo. Procedimento análogo ocorre quando se entra no modo de ajuste da velocidade de rotação do eixo do motor.

Para se enviar a ação de controle utilizando-se do controle remoto, basta pressionar a tecla para envio após ajustar o valor desejado no painel.

Após essa tecla ser pressionada, é mostrado no painel de *status* que a ação de controle foi enviada ao servidor e é mostrado o valor ajustado pelo cliente.

Para o caso do ajuste da velocidade de rotação do eixo do motor, após o computador servidor receber o valor da velocidade a ser ajustada, o programa

alterará a velocidade de rotação de referência na equação de recorrência do PID discreto (Equação (3.27)), e assim será feito o controle da velocidade de rotação do eixo do motor com base nessa nova velocidade de referência, conforme ajustada pelo usuário.

4 Resultados da Aplicação do Sistema Supervisório Desenvolvido

4.1 Introdução

Neste capítulo são apresentados os resultados obtidos na implementação do sistema supervisório. Tal capítulo foi dividido em 7 seções listadas a seguir:

- Módulos de Interface Visual do Sistema Supervisório.
- Resultados do Sistema de Aquisição de Dados.
- Resultados do Sistema de Controle.
- Resultados do Sistema de Transmissão e Recepção de Dados via TCP/IP.
- Resultados do Sistema de Supervisão de Parâmetros.
- Resultados do Sistema de Visualização de Parâmetros via Painel Eletrônico de Mensagens.
- Resultados da Interface de Entrada via Controle Remoto Infravermelho.

As metodologias empregadas em cada uma das seções foram aquelas descritas no capítulo anterior.

Em relação ao custo do sistema, levando-se em consideração os custos dos painéis eletrônicos, sistema de aquisição de dados, sistema de ajuste de tensão e interface de entrada via controle remoto infravermelho, obteve-se um valor inferior a 20% do custo de um sistema supervisório comercial utilizando um CLP e um *software* supervisório do tipo SCADA, mostrando assim, um baixo valor agregado para o sistema supervisório desenvolvido.

4.2 Módulos de Interface Visual do Sistema Supervisório

Para o sistema supervisório desenvolvido, existem 4 principais módulos de interface visual que são os seguintes:

- Visualização do sistema supervisionado através de uma câmera digital.
- Visualização dos parâmetros supervisionados através de *labels*.
- Visualização dos parâmetros supervisionados através de gráficos.
- Visualização dos parâmetros através dos painéis eletrônicos.

As descrições e resultados de cada interface são mostrados nas seções que se seguem.

4.2.1 Visualização do Sistema Supervisionado através de uma Câmera Digital

Com a utilização de uma câmera digital conectada ao computador servidor, a imagem do sistema é obtida e enviada aos clientes conectados, permitindo assim, que estes possam realizar uma inspeção visual do sistema, de forma a detectar possíveis falhas de operação.

A câmera digital utilizada no sistema foi a PC-300 da Creative®, que consiste de uma *webcam* convencional de resolução de 600 Megapixel. A comunicação entre ela e o computador servidor é feita através de um cabo USB.

Embora tenha sido utilizada a câmera PC-300, é possível conectar qualquer tipo de *webcam* no sistema sem nenhuma modificação no programa. Mesmo se a resolução for diferente, o programa identifica automaticamente as configurações da câmera, sendo possível também alterar as configurações da imagem, tais como brilho, contraste, dentre outros parâmetros utilizando-se do programa do servidor.

Na Figura 4.1 é ilustrada a imagem do sistema, vista tanto do lado do servidor como do lado do cliente.

(a)

(b)

Figura 4.1. Visualização do sistema através de uma câmera digital: (a) Imagem do sistema vista pelo programa do servidor; (b) Imagem do sistema vista pelo programa do cliente.

A única diferença entre as imagens é que no lado do servidor a imagem do sistema é atualizada praticamente em tempo real; enquanto que no lado do cliente, esta atualização depende da taxa de envio das imagens, a qual é ajustada através do programa do servidor.

O ajuste dessa taxa de envio é feito através da caixa de seleção (Intervalo de Envio) mostrada na Figura 4.1(a). Abaixo dessa, é mostrada uma caixa de verificação para permitir o ajuste automático pelo próprio programa.

Como mencionado no capítulo anterior, esse ajuste é feito por meio da determinação do tempo médio que o cliente leva para receber todos os *bytes* de uma imagem. Assim, o programa do servidor encontra a melhor taxa de envio de acordo com as disponíveis pelo programa.

4.2.2 Visualização do Sistema Supervisionado através de Labels

Os parâmetros do sistema podem ser visualizados através de *labels* tanto pelo programa do servidor como pelo programa do cliente. Os *labels* nada mais são que rótulos, que contém os valores atuais dos parâmetros coletados. Estes últimos são atualizados a cada determinado intervalo de tempo.

Para o sistema projetado, essa atualização é feita a cada 200 ms. Tal tempo mínimo de atualização é necessário, pois se os valores dos dados fossem exibidos nos *labels* conforme sua aquisição, o usuário se sentiria incomodado ao ler um valor na tela que varia muito rapidamente, mesmo que esse variasse apenas os *bits* menos significativos.

Na Figura 4.2 é mostrada a interface visual através de *labels* para os programas do servidor e do cliente.

(a)

(b)

Figura 4.2. Visualização dos parâmetros através de *labels*: (a) Programa do servidor; (b) Programa do cliente.

4.2.3 Visualização dos Parâmetros Supervisionados através de Gráficos

A visualização dos parâmetros supervisionados através de gráficos foi também implementada tanto no lado do servidor como no lado do cliente.

Para o programa do servidor, conforme os dados vão sendo coletados, estes são plotados em tempo real em suas respectivas janelas gráficas.

Já para o programa do cliente, os dados são plotados conforme a recepção dos blocos de dados enviados pelo computador servidor. Assim, por exemplo, se o servidor regular a taxa de envio dos blocos a cada 1 segundo, com uma taxa de amostragem de 40 pontos/segundo, então a cada 1 segundo o cliente receberá os 40 pontos de uma só vez, plotando-os assim em seus respectivos gráficos.

Na Figura 4.3 são mostradas as interfaces visuais através de gráficos implementadas nos dois programas.

(a)

(b)

Figura 4.3. Visualização dos parâmetros supervisionados através de gráficos: (a) Programa do servidor; (b) Programa do cliente.

4.2.4 Visualização dos Parâmetros Supervisionados através de Painéis Eletrônicos

A visualização dos parâmetros supervisionados através de painéis eletrônicos é a principal inovação proposta nessa pesquisa.

Com a utilização de um painel eletrônico monocromático para a exibição de informações de *status* e um tricolor para exibição dos dados monitorados, a supervisão do sistema se torna muito mais simples e intuitiva, permitindo que o sistema seja supervisionado não apenas através do computador, mas também através de painéis eletrônicos, possibilitando assim uma maior flexibilidade em relação aos programas supervisórios atuais.

Na Figura 4.4 são mostrados os dois painéis eletrônicos projetados e implementados para a exibição de dados e parâmetros do sistema.

Figura 4.4. Painéis eletrônicos projetados para exibição de dados e informações do sistema supervisionado.

O painel eletrônico monocromático, por ser composto de apenas um tipo de *led*, apresenta uma intensidade luminosa muito maior do que a do painel eletrônico tricolor, já que cada ponto luminoso é constituído de apenas um tipo de *led*.

No painel tricolor, cada ponto luminoso é composto de dois tipos de *led* (verde e vermelho) e, portanto, ao se acender o *led* da cor vermelha, por exemplo, é equivalente a acender metade da intensidade máxima do *led* vermelho da matriz monocromática. Por essa razão, o *driver* da matriz de *led* tricolor tem que fornecer uma corrente muito maior em comparação com a matriz de *led* monocromática.

Através da utilização de cores no painel tricolor para a indicação do *status* do parâmetro monitorado, tem-se uma supervisão muito mais intuitiva, permitindo que o usuário não tenha que memorizar a faixa de supervisão pré-programada para fazer seu monitoramento, pois basta visualizar no painel a cor na qual o determinado parâmetro é exibido para conhecer seu *status*.

Se o valor do parâmetro estiver dentro da faixa pré-programada, ele será mostrado na cor verde. Se este, estiver próximo do limite de operação pré-estabelecido, ele será mostrado na cor amarela. E no último caso, se ele ultrapassar

sua faixa de supervisão, ele será mostrado na cor vermelha e uma mensagem no painel de *status* será exibida, enfatizando que o determinado parâmetro saiu fora de sua faixa normal de operação.

4.3 Resultados do Sistema de Aquisição de Dados

Como descrito na metodologia apresentada na Seção 3.3, foram testados inicialmente diferentes amplificadores operacionais para a aquisição da tensão e corrente do motor, sendo escolhido o amplificador de instrumentação TL071 por ter apresentado melhor desempenho em comparação aos demais amplificadores.

Embora o TL062 tenha a vantagem de possuir dois amplificadores de instrumentação em um único encapsulamento, a medida da tensão do motor utilizando o TL071 ficou muito mais estável do que com o TL062.

Além disso, o TL071 apresenta um produto (largura de banda) x (ganho unitário) quase duas vezes maior do que o TL062.

Assim, escolhido o amplificador operacional a ser utilizado no projeto, foram feitos testes de comunicação entre o programa do servidor e o sistema de aquisição de dados, e se executou os ajustes necessários tanto no programa do servidor como no do programa do microcontrolador, de modo a minimizar os erros de transmissão de dados entre esses.

Em seguida, foi modificado o programa do servidor de modo que os dados recebidos do sistema de aquisição fossem plotados em tempo real nos seus respectivos gráficos.

A Figura 4.5 mostra os resultados da aquisição dos parâmetros coletados durante determinado intervalo de tempo.

Figura 4.5. Aquisição dos parâmetros supervisionados pelo programa do servidor.

As metodologias empregadas na aquisição de cada parâmetro sob supervisão funcionaram de acordo com o projetado.

Para verificação do funcionamento do oscilador externo, utilizaram-se um osciloscópio e um multímetro para medir seu período de oscilação.

Verificou-se que seu período de oscilação é de exatamente 1 Hz, como havia sido projetado. A precisão desse oscilador é de grande importância, pois é por meio dele que se determina a frequência da rede elétrica.

Medindo-se a frequência da rede utilizando a metodologia apresentada na Seção 3.3.4, conseguiu-se chegar exatamente aos 60 Hz esperados, mostrando assim a validade do método.

Na Figura 4.6, é mostrada a exibição dos parâmetros coletados, durante determinado intervalo de tempo, por meio de *labels* pelo programa do servidor.

Figura 4.6. Exibição dos dados coletados através de *labels*.

4.4 Resultados do Sistema de Controle da Velocidade de Rotação do Eixo do Motor

Conforme descrito na metodologia dessa seção, utilizou-se a ferramenta “ident” do programa MATLAB para fazer a identificação da função de transferência do sistema a ser controlado.

Na Figura 4.7 é mostrada a resposta ao degrau medida na prática e a obtida através da ferramenta “ident” do programa MATLAB.

Figura 4.7. Resposta de ensaio ao degrau.

Conforme se pode observar nessa figura, a função de transferência estimada pelo programa MATLAB está bem próxima do comportamento real do sistema. Sua função de transferência é dada pela Equação (3.27).

$$F(s) = \frac{551356}{s^2 + 1018s + 18344} \quad (3.27)$$

Conhecendo-se a função de transferência do sistema, simulou-se por intermédio da ferramenta “simulink” do programa MATLAB o método da última sensibilidade, de forma a se determinar as constantes K_{CRIT} e T_{CRIT} do sistema. De posse desses dois valores, foram então determinados os coeficientes do PID utilizando-se da Tabela 3.3, e obtida a equação de recorrência dada pela Equação (3.28).

$$u(k) = u(k-1) + 0,0114 \left[\varpi_{ref}(k) - \varpi(k) \right] - 0,0153 \left[\varpi_{ref}(k-1) - \varpi(k-1) \right] + 0,0092 \left[\varpi_{ref}(k-2) - \varpi(k-2) \right] \quad (3.28)$$

Após a obtenção dessa equação, a mesma foi implementada no programa do servidor e foram feitos pequenos ajustes nos coeficientes do PID de forma a se obter uma taxa de decaimento próxima de $\frac{1}{4}$, conforme a proposta de Ziegler-Nichols.

Assim, a equação de recorrência final para o controlador PID projetado é mostrada na Equação (3.29).

$$u(k) = u(k-1) + 0,01175 \left[\varpi_{ref}(k) - \varpi(k) \right] - 0,0174 \left[\varpi_{ref}(k-1) - \varpi(k-1) \right] + 0,0094 \left[\varpi_{ref}(k-2) - \varpi(k-2) \right] \quad (3.29)$$

A resposta ao degrau para o controlador projetado é mostrada na Figura 4.8.

Figura 4.8. Resposta ao degrau utilizando o PID projetado.

Pela resposta ao degrau obtida, observa-se que o PID projetado possui uma taxa de decaimento bem próxima de $\frac{1}{4}$ e sobre-sinal de 20%, atendendo assim as especificações de projeto.

Modificando-se os valores dos coeficientes do PID, pelo programa do servidor, é possível modificar a dinâmica de controle do sistema. Mas um dos problemas ao se fazer o *tuning* manual é não saber como ajustar corretamente cada um desses três coeficientes.

Assim, é importante o usuário utilizar as ferramentas do MATLAB (“simulink” e “rltool”) para determinar os coeficientes do PID, e a partir de pequenos ajustes manuais feitos pelo programa do servidor, fazer o *tuning* do controlador até se atingir o desempenho desejado.

Como um exemplo dessa metodologia, foram modificados os coeficientes do PID de modo a obter uma resposta de controle mais rápida, conforme mostra a Figura 4.9.

Figura 4.9. Resposta ao degrau modificando os coeficientes do PID projetado.

A equação de recorrência para esse PID é dada pela Equação (3.30).

$$u(k) = u(k-1) + 0,036 \left[\varpi_{ref}(k) - \varpi(k) \right] - 0,0621 \left[\varpi_{ref}(k-1) - \varpi(k-1) \right] + 0,0262 \left[\varpi_{ref}(k-2) - \varpi(k-2) \right] \quad (3.30)$$

Como se pode visualizar pela Figura 4.9, diminuir o tempo de resposta do sistema acarreta em um aumento do sobre-sinal, além de um aumento no tempo de acomodação do sistema.

Portanto, no projeto do controlador deve-se ter sempre o compromisso entre tempo de resposta, sobre-sinal e tempo de acomodação.

Como no projeto foi empregada a metodologia de Ziegler-Nichols, obteve-se como resultado um bom comprometimento entre essas características.

4.5 Resultados do Sistema de Transmissão e Recepção de Dados via TCP/IP

Como discutido na metodologia dessa seção, os dados coletados pelo sistema de aquisição são enviados em blocos a partir do computador servidor aos clientes conectados.

O ajuste do tempo de envio dos blocos e a taxa de aquisição dos dados são feitos através de caixas de seleção, conforme mostradas na Figura 4.6. O intervalo de envio da imagem da câmera pode ser ajustado através de sua correspondente caixa de seleção, ou pode-se selecionar a opção de ajuste automático, de modo que o computador servidor determine a melhor taxa de envio para cada cliente.

O ajuste automático é feito de acordo com o tempo que o cliente leva para receber todos os *bytes* de uma imagem. Assim, como mencionado anteriormente,

não há necessidade de se enviar imagens a cada 250 ms a um cliente que leva 500 ms para receber todos os *bytes* de uma imagem.

O programa do cliente possui caixas de *logs* que registram o horário de recebimento dos blocos de dados, imagens e a latência da rede.

Na Figura 4.10 são mostrados os *logs* gerados durante a recepção de dados e imagens do servidor pelo programa do cliente.

Figura 4.10. Logs gerados no programa do cliente ao receber dados e imagens do servidor.

Como se pode observar nessa figura, os *logs* permitem saber o horário exato que um determinado bloco de dados foi recebido pelo cliente.

Na Figura 4.2(b) mostrada anteriormente, é exibido no programa do cliente o *delay* dos dados recebidos em um *label*. Esse *delay* corresponde ao intervalo de tempo que um cliente leva para receber todos os *bytes* de um bloco.

Em outros *labels* nessa mesma figura é mostrado o tempo de conexão do cliente ao servidor, a latência da rede e o número de blocos recebidos pelo cliente. Assim, esses parâmetros correspondem aos dados que serão enviados ao painel eletrônico de *status*, sendo, portanto, exibidos.

Ao receber um bloco de dados, o programa do cliente os plotará de uma só vez nos seus respectivos gráficos.

Voltando-se à Figura 4.10, observam-se caixas de verificação no programa do cliente. Essas caixas têm a função de fazer a solicitação dos tipos de serviços disponíveis no servidor.

Como mencionado, nem todo cliente necessita de todos os tipos de serviços, e isso contribui para se otimizar o desempenho do servidor, pois nesse caso, não é necessário enviar todos os tipos de serviços para todos os clientes conectados, somente os tipos de serviços que o determinado cliente solicitar.

Portanto, através das caixas de verificação, o cliente pode solicitar ao servidor os tipos de serviços que ele desejar. Os tipos de serviço disponíveis no servidor são: serviços de dados; serviço de imagem; e serviço de controle do sistema.

Na Figura 4.11 é mostrado como o cliente configura o tipo de controle e seu valor.

Figura 4.11. Configuração da ação de controle pelo programa do cliente.

Através da caixa de seleção “Tipo de Controle”, o usuário seleciona se deseja ajustar a tensão de alimentação do motor, ou se deseja controlar a velocidade de rotação do eixo do motor.

Feito isso, basta o usuário digitar na caixa de entrada correspondente o valor desejado e pressionar o botão “Ok”.

Como comentado anteriormente, se o tipo de controle escolhido for o de ajuste de tensão do motor, então quando o cliente enviar ao servidor um valor de tensão a ser ajustado, este último acionará o sistema de ajuste de tensão, regulando dessa forma a tensão desejada pelo cliente.

Se o tipo de controle escolhido for o da velocidade de rotação do eixo do motor, então quando o servidor receber um valor de velocidade a ser ajustado, ele irá alterar na Equação (3.27) a velocidade de referência do eixo do motor, controlando dessa forma a velocidade do eixo do motor.

Na Figura 4.12 é mostrada a configuração do sistema de controle no lado do servidor.

A partir da caixa de seleção “Controle” é possível selecionar se o controle será feito pelos clientes ou pelo servidor. No caso de ser selecionada a opção “Controle Servidor”, o controle do sistema será feito exclusivamente pelo servidor.

Nesse caso, os clientes não poderão controlar o sistema até que seja escolhida a opção “Controle Cliente” no programa do servidor.

Figura 4.12. Configuração da ação de controle pelo programa do servidor.

Dentro da caixa de grupo “Controle do Sistema” é mostrado o nome do controlador atual, o tempo que este está controlando o sistema, o tipo de controle e o último valor de controle ajustado.

Toda vez que um controlador travar ou destravar o sistema, será registrado no *Log* mostrado nessa mesma figura o nome do controlador e seu correspondente

número IP, consistindo portanto, de um histórico dos controladores que atuaram no sistema.

Na Figura 4.13 (a) é mostrada a implementação do *chat* no programa do servidor, enquanto que nas Figuras 4.13 (b) e (c) é mostrada essa implementação no programa do cliente.

Para que um determinado cliente saiba quais são os demais usuários *on-line*, é necessário que ele possua uma lista de clientes conectados. Assim, quando o cliente se conectar ao servidor, ele receberá deste último a lista de clientes conectados no momento.

Antes do cliente se conectar, é necessário que ele defina seu nome ou apelido, escrevendo-o na caixa de texto “Apelido”, conforme mostrada na Figura 4.13 (b). Ao tentar se conectar ao servidor, este o adicionará com tal nome em sua lista de clientes conectados, e ao mesmo tempo enviará uma mensagem a todos os demais clientes, informando o nome do cliente que entrou. Dessa forma, o nome do cliente será também adicionado na lista de usuários *on-line* dos demais clientes que já estavam conectados no servidor.

Quando um cliente se desconecta, este é removido da lista de clientes conectados do servidor, bem como é enviada por esse último, uma mensagem aos demais clientes informando o nome do cliente que se desconectou, de forma a ser removido de suas correspondentes listas de usuários *on-line*.

Procedimento análogo acontece quando um cliente altera seu nome ou apelido. Nesse caso, o nome do usuário é alterado na lista de clientes conectados do servidor, e é enviada uma mensagem aos demais clientes informando o nome do cliente que trocou de apelido.

Através do *chat*, é possível manter conversas reservadas entre os clientes, utilizando-se da caixa de verificação “Reservadamente” mostrada na Figura 4.13(b).

Dessa forma, um cliente pode conversar com outro, sem que os demais clientes visualizem a conversa.

O servidor foi implementado de forma que seja o único que consegue visualizar todas as conversas, mesmo as reservadas.

(a)

(b)

(c)

Figura 4.13. Implementação do *chat* no programa supervisorio: (a) *Chat* no programa do servidor; (b) *Chat* no programa do cliente 1; (c) *Chat* no programa do cliente 2.

Para um cliente conversar com outro, basta selecioná-lo através da caixa de seleção de usuários mostrada na Figura 4.13 (b), escrever a mensagem na caixa de texto ao lado, e pressionar o botão “Enviar”.

O computador servidor é o responsável por fazer o gerenciamento das conexões, bem como dos tipos de serviços solicitados por cada cliente. Assim, na Figura 4.13 (a) é mostrado um *Memo* que armazena as informações dos clientes que se conectaram ou desconectaram do servidor.

As informações armazenadas dos clientes nesse *Memo* são as seguintes: nome do *host*, número IP, apelido do cliente, e *status* para indicar se ele se conectou ou desconectou do sistema.

Esse *Memo* tem a função de armazenar o histórico de todos os clientes que se conectaram e desconectaram do servidor, e não apenas se ele está conectado no momento.

Para mostrar a lista de clientes conectados no instante atual, e o tipo de serviço solicitado por cada cliente, foram implementadas 3 tabelas no programa do servidor que aumentam de tamanho dinamicamente de forma a mostrar as propriedades de cada cliente.

Essas tabelas são mostradas nas Figuras de 4.14 a 4.16.

Figura 4.14. Tabela de clientes conectados.

Figura 4.15. Tabela do serviço câmera.

Figura 4.16. Tabela do serviço dados.

A tabela de clientes conectados, mostrada na Figura 4.14, ilustra a relação de todos os clientes que estão conectados naquele instante no servidor. Por meio dela, o usuário pode visualizar para cada cliente o correspondente nome do *host*, o número IP, o apelido do cliente no *chat*, e o número do cliente que corresponde à ordem na qual o cliente se conectou ao sistema, sendo os clientes do topo os que estão há mais tempo conectados no servidor.

A tabela do serviço câmera, ilustrada na Figura 4.15, mostra os clientes que solicitaram o serviço de envio de imagem do sistema. Além das mesmas propriedades apresentadas na tabela de clientes conectados, esta mostra também a taxa de envio de imagem ajustada para cada cliente, bem como o *delay* médio que o dado cliente demora a receber todos *bytes* de uma imagem. Esse tempo médio é calculado como sendo a média dos tempos das 5 últimas imagens enviadas.

A última tabela apresentada na Figura 4.16 mostra os clientes que solicitaram o serviço de envio de dados do sistema. Para essa tabela, além das propriedades da tabela de clientes conectados, é também mostrada a latência média dos clientes que solicitaram o serviço de dados, bem como o tempo que o dado cliente demora a receber um bloco de dados.

Na Figura 4.13, é mostrado um exemplo de 2 usuários conectados ao servidor, com nomes Fernando e Fábio, sendo que Fernando solicitou o serviço de dados, enquanto Fabio solicitou o serviço de câmera. Como se pode verificar na tabela de clientes conectados (Figura 4.14), os nomes dos dois usuários são mostrados nessa tabela, como era de se esperar.

Como somente Fernando solicitou o serviço de dados, somente ele é mostrado na tabela do serviço de dados (Figura 4.16). No caso da tabela do serviço de câmera (Figura 4.15), apenas é mostrado o nome de Fábio, pois somente ele solicitou esse tipo de serviço.

4.6 Resultados do Sistema de Supervisão de Parâmetros

O programa de supervisão de parâmetros tem o objetivo de verificar se os valores dos parâmetros monitorados estão dentro de suas faixas de supervisão pré-programadas. Nesse caso, o usuário pode programar os valores limites de operação para cada parâmetro sob supervisão.

Na Figura 4.17 é mostrada a tela de edição das faixas de supervisão para cada parâmetro do sistema através do programa do servidor. No lado do cliente, a edição das faixas de supervisão é idêntica à mostrada para o programa do servidor.

Figura 4.17. Edição das faixas de supervisão dos parâmetros supervisionados pelo programa do servidor.

Pressionando-se o botão de edição de faixas, uma tela de seleção do tipo de faixa será exibida conforme mostra a Figura 4.18.

Figura 4.18. Seleção do tipo de faixa de supervisão.

Após a seleção do tipo de faixa, deve-se pressionar o botão “Configurar Faixa”, onde é possível estabelecer os valores dos patamares da faixa de supervisão conforme mostrado na Figura 4.19.

Figura 4.19. Edição dos patamares da faixa de supervisão selecionada.

Assim, através dessa interface, consegue-se facilmente configurar o tipo de faixa de supervisão, bem como os valores dos patamares para cada parâmetro supervisionado.

Se durante a operação do sistema o valor de determinado parâmetro ultrapassar sua faixa de supervisão pré-programada, tanto no lado do cliente como no lado do servidor, será enviada ao painel eletrônico de *status* a mensagem de que o parâmetro ultrapassou sua faixa de supervisão sendo, portanto, exibida no painel.

Nesse caso, será também exibido no painel eletrônico de dados o valor corrente desse parâmetro, interrompendo momentaneamente a seqüência de exibição dos valores dos dados coletados.

4.7 Resultados do Sistema de Visualização de Parâmetros via Painel Eletrônico de Mensagens

Como mencionado na metodologia descrita na Seção 3.7, para a exibição de mensagens através dos painéis eletrônicos, foi primeiramente implementado um programa para apenas exibir mensagens digitadas em uma caixa de texto, sendo estas exibidas no painel eletrônico conectado à porta serial do computador. As opções implementadas no programa projetado são mostradas na Figura 4.20.

Figura 4.20. Programa para exibição de mensagens através dos painéis eletrônicos.

Como se pode visualizar por essa figura, esse programa possui uma caixa de texto que permite a edição da frase, bem como caixas de seleção para a escolha do efeito e velocidade da mensagem. Após a edição da mensagem, basta pressionar o botão “Enviar” para que a mensagem seja encaminhada ao microcontrolador do painel, sendo, portanto, exibida.

Para o caso do painel de dados que consiste de um painel tricolor, é possível configurar também a cor de cada linha da mensagem. Nesse caso, pode-se, por

exemplo, fazer com que as duas primeiras linhas sejam verdes, as duas seguintes amarelas e as restantes vermelhas, dando assim um maior apelo visual à mensagem. Esse exemplo é mostrado na Figura 4.20.

Após a implementação e testes desse programa, este foi adaptado aos programas do cliente e do servidor, conforme mostrado na Figura 4.21.

(a)

(b)

Figura 4.21. Programa do painel eletrônico implementado no servidor e no cliente:
(a) Programa do servidor; (b) Programa do cliente.

Como se pode observar pela Figura 4.21 (a) e (b), a funcionalidade é similar ao do programa implementado anteriormente, com a diferença que esse permite a configuração individual dos parâmetros de exibição para cada painel.

Outra diferença é que para o programa do cliente, foi necessário adaptá-lo para que o mesmo faça a exibição automática dos dados e informações de *status* do sistema.

Na Figura 4.21 (a) e (b) são mostradas as configurações das mensagens manuais digitadas pelo usuário para exibição nos painéis. No caso das mensagens automáticas de dados e *status*, foi implementada uma tela de configuração para cada parâmetro do sistema a ser exibido, conforme mostrado na Figura 4.22.

(a)

(b)

Figura 4.22. Tela de configuração de mensagens automáticas: (a) Mensagens automáticas do painel de *status*; (b) Mensagens automáticas do painel de *dados*.

Através de caixas de seleção, o usuário pode selecionar o parâmetro do sistema sob supervisão e escolher seu correspondente efeito e velocidade. Para aplicar as configurações ao parâmetro, basta o usuário pressionar o botão “Aplicar Efeito” após a seleção das suas propriedades de exibição.

O programa permite também que uma determinada configuração de exibição seja aplicada a todos os parâmetros de uma só vez. Para isso, basta o usuário selecionar na caixa de seleção “Aplicar em:” a opção “Aplicar em Todos os Parâmetros”, e em seguida pressionar o botão “Aplicar Efeito”.

Tanto as configurações dos painéis como as das faixas de supervisão podem ser gravadas em arquivo, de modo a permitir que o usuário as carregue toda vez que desejar.

Após a implementação do programa do painel, tanto no lado do cliente quanto no lado do servidor, foram feitas modificações nestes de modo a permitir a exibição de mensagens de alerta, no caso de algum parâmetro ultrapassar sua faixa de supervisão pré-programada.

Se o valor de determinado parâmetro ultrapassar sua faixa de supervisão no lado do servidor, este enviará aos clientes conectados, via TCP/IP, uma mensagem informando que o determinado parâmetro saiu de sua faixa de operação, sendo, portanto, exibida no painel de *status* de cada cliente conectado.

Outra funcionalidade adicionada foi permitir ao cliente escrever mensagens de informação a serem exibidas em qualquer um dos painéis eletrônicos, permitindo assim que mensagens complementares de informações possam ser adicionadas pelos usuários do programa.

Esse procedimento é feito também no lado do servidor, permitindo que este também tenha acesso aos painéis eletrônicos dos clientes. Nesse caso, as

informações do servidor seriam difundidas como uma espécie de *broadcast* a todos os painéis dos clientes conectados.

A última modificação feita no programa do painel foi fazer com que a cor do parâmetro sob supervisão fosse exibida de acordo com seu valor de *status*, considerando a sua faixa de supervisão pré-programada.

Nesse caso, se seu valor corrente estiver dentro de sua faixa de supervisão, este seria mostrado na cor verde no painel de dados. No caso do valor estar próximo da extremidade da faixa, este seria então mostrado na cor amarela, indicando atenção ao usuário.

No último caso, se esse valor ultrapassasse sua faixa de supervisão, ele seria mostrado na cor vermelha, indicando que uma possível ação de controle deva ser efetuada pelo cliente.

Portanto, a cor de cada parâmetro sob supervisão é mostrada de acordo com seu valor de *status*.

A utilização de cores para a indicação do *status* dos parâmetros sob supervisão constitui uma nova forma de utilização dos painéis eletrônicos coloridos, que na grande maioria dos casos são utilizados apenas para dar um maior apelo visual à mensagem, e não como uma função de *status* conforme a proposta dessa pesquisa.

4.8 Resultados da Interface de Entrada via Controle Remoto Infravermelho

Conforme descrito na metodologia dessa seção, foi projetado um módulo receptor de infravermelho capaz de decodificar o sinal proveniente de um controle

remoto, e enviar o caractere ASCII equivalente à tecla identificada, via porta serial, ao computador cliente conectado.

Na Figura 4.23 são mostrados o receptor de infravermelho projetado e o controle remoto utilizado.

Figura 4.23. Receptor de infravermelho projetado e controle remoto utilizado.

Na Figura 4.24 é mostrada a edição da tensão de armadura do motor através do painel eletrônico de *status*, utilizando o controle remoto infravermelho da Sanyo.

Figura 4.24. Edição da tensão de armadura do motor através do painel eletrônico de *status*.

Na Figura 4.25 é mostrada a edição da velocidade de rotação do eixo do motor através do painel eletrônico de *status*.

Figura 4.25. Edição da velocidade de rotação do eixo do motor através do painel eletrônico de *status*.

Pelas Figuras 4.24 e 4.25 nota-se que a edição dos valores a serem enviados ao computador servidor é bem simples, pois conforme o usuário for pressionando o número desejado no controle remoto, o programa do cliente sabe a posição exata em que esse caractere deve aparecer no painel eletrônico, concatenando assim os caracteres recebidos.

Se um número for pressionado erroneamente, basta apertar a tecla “<” do controle remoto, para que o último caractere digitado se torne o caractere “0” e a próxima tecla pressionada entre nessa posição.

Para enviar a ação de controle ao computador servidor, basta o usuário pressionar a tecla “>” do controle remoto após a edição do valor desejado no painel.

No instante em que a ação de controle for enviada, uma mensagem surgirá no painel status, indicando que a ação de controle foi enviada, e exibindo também o valor desejado pelo cliente.

O controle remoto pode ser utilizado tanto para a edição de valores de controle a serem visualizados no painel, como também para a seleção de funções específicas no programa do cliente.

Dessa forma, algumas teclas que não foram utilizadas para ajustar a tensão ou a velocidade de rotação do eixo do motor foram aplicadas em funções específicas, tais como: conectar/desconectar o cliente ao servidor; solicitar/parar o serviço de envio de dados; e solicitar/parar o envio da imagem do sistema monitorado.

Portanto, através do controle remoto é possível realizar algumas funções sem a necessidade de utilizar o teclado do computador, flexibilizando assim a utilização do programa supervisorio desenvolvido.

5 Conclusões

Nesse trabalho foi apresentado um sistema supervisório via TCP/IP, que se apresenta como uma alternativa viável na supervisão remota de sistemas, possuindo um custo inferior a 20% do valor de um sistema supervisório comercial, possibilitando dessa forma que sistemas de pequena escala possam também ser supervisionados remotamente, sem a necessidade de grandes investimentos tanto em *hardware* como em *software*.

Embora a aplicação do sistema supervisório desenvolvido seja na área de máquinas elétricas, esse sistema pode ser generalizado para as mais diversas áreas e aplicações onde é necessário realizar o monitoramento remoto.

A principal inovação proposta nesse trabalho, em comparação com os sistemas supervisórios do tipo SCADA, consiste da utilização de painéis eletrônicos para a exibição dos dados e informações de *status* do sistema, fornecendo assim, uma nova interface visual ao usuário, facilitando então a sua supervisão.

A utilização de uma câmera digital no projeto possibilita que cada cliente conectado ao servidor possa realizar uma inspeção visual do sistema, de modo a detectar eventuais falhas de operação, consistindo assim, de uma ferramenta adicional de auxílio na monitoração do sistema.

O *chat* implementado dentro do próprio programa supervisório torna muito mais prática a comunicação entre clientes e servidor, dispensando então a utilização de quaisquer programas adicionais para essa finalidade, tais como MSN, ICQ, IRC e SKYPE.

O sistema supervisório desenvolvido foi projetado de acordo com a topologia Servidor/Multi-Clientes, possibilitando que múltiplos clientes possam se conectar ao

mesmo tempo ao servidor e realizar seu monitoramento, tornando o mesmo distribuído e não centralizado em um único computador.

O projeto dos circuitos de acionamento dos painéis eletrônicos nessa pesquisa contribui para disseminar o domínio tecnológico presente nas indústrias de painel eletrônico, permitindo que novos trabalhos nessa área de pesquisa possam ser realizados, tanto para o painel eletrônico monocromático quanto para o painel eletrônico tricolor.

O emprego de um módulo receptor de infravermelho no sistema permite que o usuário possa ajustar a ação de controle a partir do painel de status, utilizando-se de um controle remoto, flexibilizando assim o sistema supervisório desenvolvido, pois este pode ser monitorado e controlado sem a necessidade de utilizar diretamente o computador.

Como trabalhos futuros a serem explorados, pode-se buscar novas aplicações para o sistema supervisório desenvolvido, bem como melhorar o sistema de aquisição de dados objetivando aumentar a taxa de aquisição e transmissão dos dados coletados, permitindo assim, que o sistema supervisório possa ser utilizado para monitorar grandezas que possuam componentes de alta frequência em tempo real.

Trabalhos na área de painéis eletrônicos podem também ser explorados de forma a se otimizar o hardware de acionamento dos painéis eletrônicos, bem como podem ser adicionadas mais linhas ou colunas em relação aos painéis desenvolvidos nessa pesquisa, possibilitando assim, que painéis maiores possam ser construídos e um maior número de informações possa ser exibida em um mesmo painel.

Bibliografia

- [1] MACIEL, C.D.; RITTER, C.M. *TCP/IP networking in process control plants*, Elsevier Science, Vol. 35, Nº 3-4, pp. 611-614, 1998.
- [2] CHRYSOSTOMOU, C.; PITSILLIDES, A.; ROSSIDES, L.; SEKERCIOGLU, A. *Fuzzy logic controlled RED: congestion control in TCP/IP differentiated services networks*, Springer-Verlag, 2003.
- [3] BRADY, K.; TARN, T. "Internet-based remote teleoperation", *Proceedings of the 1998 IEEE International Conference on Robotic and Automation*, Vol. 1, pp. 65-70, 1998.
- [4] SILVA, A. P. G.; SALVADOR M., "O que são sistemas supervisórios ?", <http://www.elipse.com.br/download/download/artigos/rt025.04.pdf>, Acessado em 05/01/2007.
- [5] BALLOCK, I. R. *Protótipo de um Sistema para Supervisão de Equipamentos Industriais*, Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau, FURB, 2003.
- [6] GIOZZA, W. F.; ARAÚJO, J.F.M. ; MOURA, J.A.B.; SAUVÉ, J.P. *Redes Locais de Computadores*, McGraw-Hill, 1986.
- [7] FONTES, I.R.; CAGNON J.A. "Supervisão e controle para aplicação em viveiros de mudas", *AGRENER*, 2002.

- [8] CANSADO, J.C.A.; SARAIVA, A.M. *AGRILOC – Sistema para Experimentação de Controle Climático de Casas de Vegetação*, Dissertação de Mestrado, Escola Politécnica da Universidade de São Paulo, São Paulo, 2003.
- [9] ARAUJO, J.J.; BECKER, L.B.; MEDEIROS, G.C.R. *Interface de Comunicação entre Ambiente de Modelagem e Sistema Supervisório*. Monografia de Bacharelado em Ciência da Computação, Universidade Federal de Pelotas, 2000.
- [10] PUPO, M. S.; GONZAGA, A.G. *Interface Homem-Máquina para Supervisão de Um CLP em Controle de Processos através da WWW*, Dissertação de Mestrado, Escola de Engenharia de São Carlos, São Carlos, 2002.
- [11] BOARETTO, N. “Coleta de dados e monitoramento de chão de fábrica na manufatura discreta – integração com as ferramentas de gestão”, *XI SIMPEP*, Bauru, 2004.
- [12] NAZARENO, C.; BORGES, F.F. “Telemetria do sistema de esgoto sanitário da CAESB – Monitoramento remoto de pequenas unidades a baixo custo”, *XXVII Congresso Interamericano de Engenharia Sanitária e Ambiental*, Porto Alegre, 2000.
- [13] ÁLVARES, J.A.; FERREIRA, J.C.E. “Metodologia para implantação de laboratórios remotos via *internet* na área de automação da manufatura”, *Congresso Brasileiro de Engenharia de Fabricação*, Uberlândia, 2003.
- [14] FERWORN, A.; ROQUE, R.; VECCHIA, L. “MAX: Wireless Teleoperation via World Wide Web”, *Proceedings of the 1999 IEEE Conference on Electrical and Computer Engineering*, Vol. 3, pp. 1380-1384, 1999.

- [15] GOLDBERG, K.; CHEN, B.; SOLOMON, R.; BUI, S.; FARZIN, B.; HEITLER, J.; POON, D.; SMITH, G. "Collaborative Teleoperation via the Internet", *Proceedings of the 2000 IEEE Conference on Robotics & Automation*, Vol. 2, pp. 2019-2024, 2000.
- [16] HU, H.; YU, L.; TSUI, P.W.; ZHOU, Q. "Internet-based Robotic Systems for Teleoperation", *International Journal of Assembly Automation*, Vol. 21, Nº 2, pp. 143-151, 2001.
- [17] KRESS, R.L.; HAMEL, W. R.; MURRAY, P.; BILLS, K. "Control Strategies for Teleoperated Internet Assembly", *IEEE/ASME Transaction on Mechatronics*, Vol. 6, Nº 4, pp. 410-416, 2001.
- [18] WEAVER, A. C.; LUO, J.; ZHANG, X. "Monitoring and control using the internet and Java", *The 25th Annual Conference of the IEEE – Industrial Electronics Society*, Vol. 3, pp. 1152-1158, 1999.
- [19] ÁLVARES, A.J.; FERREIRA, J. C. E.; SHIMABUKURO, L.; FERNANDES, C.A. "Um sistema de telemanufatura baseado na WEB orientado ao processo de oxicorte", *XII Encontro Nacional de Engenharia de Produção*, ENEGEP, 2001.
- [20] LUO, R.C.; SU, K.L.; SHEN, S.H.; TSAI, K.H. "Networked Intelligent Robots Through the Internet: Issues and Opportunities", *Proceedings of IEEE*, Vol. 91, pp. 371-382, 2003.
- [21] SOARES, L.F.G.; LEMOS, G.; COLCHER, S. *Redes de Computadores: Das Lans, Mans e Wans às Redes ATM*, Editora Campus, 2ª Edição, 1995.

- [22] COMER, D.E. *Internetworking with TCP/IP: Principles, Protocols and Architecture*, Prentice-Hall, 3ª Edição, 1999.
- [23] CHAPMAN, S.J. *Electric Machinery Fundamentals*, McGrawHill, 4ª Edição, 2005.
- [24] OLIVEIRA, V. A.; AGUIAR, M.L.; VARGAS, J.B. *Sistemas de Controle: Aulas de Lab oratório*, Editora USP, 1ª Edição, 2001.
- [25] VISOLLI, A. "Tuning of PID controllers with fuzzy logic", *Proceedings of the IEE - Control Theory and Applications*, Vol. 148, pp. 1-8, 2001.
- [26] KUNG, Y.S.; OUYANG, M.; LIAW, C.M. "Adaptative speed control for induction motor using neuralnetwork", *Proceedings of IECON '93*, Vol. 1, pp. 188-193, 1993.
- [27] LIAW, C.M.; CHENG, S.Y. "Fuzzy two-degrees-of-freedom speed controller for motor drives", *IEEE Transactions on Industrial Electronics*, Vol. 42, pp. 209-216, 1995.
- [28] TIPSUWAN, Y.; MO-YUEN CHOW "Fuzzy logic microcontroller implementation for DC motor speedcontrol", *Proceedings of IECON '99*, Vol. 3, pp.1271-1276, 1999.
- [29] GONSALVES, P.G.; CAGLAYAN, A.K. "Fuzzy logic PID controller for missile terminal guidance", *Proceedings of the 1995 IEEE International Symposium on Intelligent Control*, Vol. 1, pp. 377-382, 1995.

- [30] MALKI, H.A.; MISIR, D.; FEIGENSPAN, D.; CHEN, G. "Fuzzy PID control of a flexible-joint robot arm with uncertainties from time-varying loads", *IEEE Transactions on Control Systems Technology*, Vol. 5, pp. 371-378, 1997.
- [31] ZIEGLER, J. G.; NICHOLS, N. B. "Optimum settings for automatic controllers", *ASME Transactions*, Vol. 64, pp. 759-768, 1942.
- [32] JEONG, K.C.; KWON, S.H.; LEE, D.H.; LEE, M.W.; CHOI, J.Y. "A fuzzy logic-based gain tuner for PID controllers", *Proceedings of the 1998 IEEE International Conference on Fuzzy Systems*, Vol. 1, pp. 551-554, 1998.
- [33] VEGTE, J. V. *Feedback Control Systems*, Prentice-Hall, 3ª Edição, 1994.
- [34] FRANKLIN, G.F.; POWELL, J.D.; NAEINI, A.E. *Feedback Control of Dynamic Systems*, Prentice-Hall, 4ª Edição, 2006.
- [35] SOARES, L.F.G.; LEMOS, G.; COLCHER, S. *Rede de Computadores*, Editora Campos, 2ª Edição, 1995.
- [36] PANG, P.; KWAN, T.; CHAN, C.; LIU, H. "Led Traffic Light as a Communications Device", *Proceedings of the IEEE International Conference on Intelligent Transport Systems*, pp. 788-793, 1999.
- [37] SURAKAMPONTORN, W.; JUTAVIRIYA, S.; RIEWRUJA, V. "Segmented thai numeric display", *IEEE Transactions on Consumer Electronics*, Vol. 35, Nº 4, pp. 840-844, 1989.
- [38] LAU, K.T.; JAWALEKAR, S.R. "A display system for hindi numerics", *IEEE Transactions on Consumer Electronics*, Vol. 34, Nº 2, pp. 352-356, 1988.

- [39] LAKEOU, S.; CHAU, M.; GIDAY, A. "Display controller IC for amharic numerals", *IEEE Transactions on Consumer Electronics*, Vol. 39, N° 2, pp. 77-81, 1993.
- [40] BALASUBRAMANIAN, K.; RAJAN, P.K. "Multi-numeric display systems", *IEEE Transactions on Consumer Electronics*, Vol. 35, N° 1, pp. 24-31, 1989.
- [41] NIZAMANI, A.S. "On dot matrix displays for Arabic text", *IEEE Transactions on Consumer Electronics*, Vol. 36, N° 2, pp. 76-83, 1990.
- [42] BALASUBRAMANIAN, K. "Multi-numeric decoder for dot matrix display", *IEEE Transactions on Consumer Electronics*, Vol. 38, N° 1, pp. 37-41, 1992.