

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO**

**MÍDIAS E TECNOLOGIAS INSTRUCIONAIS PARA O ENSINO/APRENDIZADO
DE SISTEMAS DE CONTROLE**

Tese submetida ao PPGEP da Universidade Federal de Santa Catarina para a obtenção do
Grau de Doutor em Engenharia de Produção.

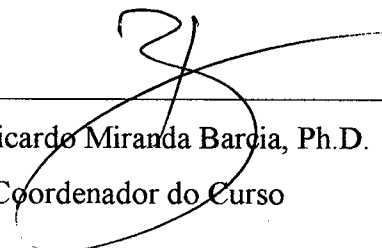
ORLANDO FONSECA SILVA

Florianópolis, Dezembro/2001

ORLANDO FONSECA SILVA

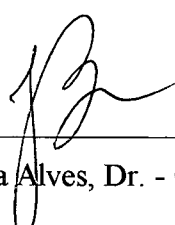
**MÍDIAS E TECNOLOGIAS INSTRUCIONAIS PARA O ENSINO/APRENDIZADO
DE SISTEMAS DE CONTROLE**

Esta Tese foi julgada adequada para obtenção do título de “Doutor”, especialidade em Engenharia de Produção e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Produção.



Prof. Ricardo Miranda Bardia, Ph.D.
Coordenador do Curso


Banca examinadora:




Prof. João Bosco da Mota Alves, Dr. - Orientador



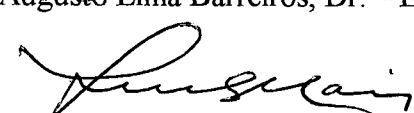
Prof. Raimundo Nazareno Cunha Alves, Dr. - Moderador



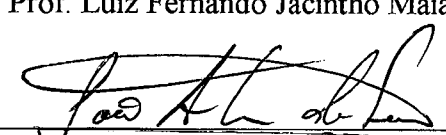
Prof. Ubiratan Holanda Bezerra, Dr. – Examinador externo



Prof. José Augusto Lima Barreiros, Dr. – Examinador externo



Prof. Luiz Fernando Jacintho Maia, Dr.



Prof. João Artur de Souza, Dr. – Examinador externo

DEDICATÓRIA

À Maria Fonseca, Eliana Elena, Juliana e Maria Elena.

AGRADECIMENTOS

Ao Professor João Bosco, pela orientação desburocratizada, consistente e objetiva.

Ao Maia, Bira, Barreiros, João e Maca, pelas contribuições para o texto final.

Ao Max pela estratégica “hospedagem” com direito a *rock*, revisões e *cocota*.

Ao Tobias, que agilizou a possibilidade de sermos vistos mundo afora.

Ao Barra e Tavares, pelas conversas sobre conteúdos e apresentação.

Ao Oscar, pelas poucos, mas profundos papos no Tadeu.

À Bernadete, que pacientemente revisou as referências.

Ao Johnny, pelas aulas práticas de redes e segurança.

À Andreyra e ao Reinaldo, pelas dicas sobre a placa.

A CAPES pela bolsa de estudos.

Ao Mario, Mauro, Ronaldo e Wilson, pelos profundos ensinamentos sobre a Ilha.

Ao Ferreira e todo pessoal do Iega, que sempre nos receberam com profissionalismo e consideração, permitindo um clima propício para a troca de idéias.

SUMÁRIO

LISTA DE FIGURAS	8
LISTA DE TABELAS.....	10
RESUMO.....	11
ABSTRACT	12
CAPÍTULO 1 – INTRODUÇÃO.....	13
CAPÍTULO 2 – LABORATÓRIO DE EXPERIMENTAÇÃO REMOTA	16
2.1 – CONCEITO	16
2.2 – ESTADO DA ARTE	17
2.2.1 – <i>Laboratório de sistemas dinâmicos e controle da Universidade do Texas</i>	<i>17</i>
2.2.2 – <i>Laboratório de experimentação remota da Universidade de Carnegie Mellon ..</i>	<i>17</i>
2.2.3 – <i>Laboratório de experimentação remota da Universidade de Western Austrália</i>	<i>17</i>
2.2.4 – <i>Laboratório de experimentação remota da Universidade de Brasília.....</i>	<i>18</i>
2.2.5 – <i>RExLab – O Laboratório de Experimentação Remota da Universidade Federal de Santa Catarina</i>	<i>18</i>
2.2.6 – <i>Laboratório da Universidade do Tennessee- Chattanooga.....</i>	<i>18</i>
2.3 – ANÁLISE DOS LABORATÓRIOS DE EXPERIMENTAÇÃO REMOTA.....	18
2.4 – CONCLUSÕES	20
CAPÍTULO 3 – LABORATÓRIO PRESENCIAL	21
3.1 – OBJETIVOS DO LACOS/DEEC/UFPA.....	21
3.1.1 – DINÂMICA DO LABORATÓRIO DE SISTEMAS DE CONTROLE	22
3.2 – EQUIPAMENTOS.....	23
3.2.1 – <i>Treinamento com amplificadores operacionais.....</i>	<i>23</i>
3.2.2 – <i>Simulador para controle de processos</i>	<i>23</i>
3.2.3 – <i>Treinamento com servomecanismo.....</i>	<i>24</i>
3.2.4 – <i>Sistema térmico</i>	<i>25</i>
3.2.5 – <i>Fun plate</i>	<i>25</i>
3.2.6 – <i>Ball beam</i>	<i>26</i>

3.2.7 – Esteira Transportadora.....	26
3.2.8 – Pêndulo invertido	27
3.2.9 – Sistema massa-mola-amortecedor.....	27
3.2.10 – Placa conversora AD/DA.....	28
3.2.11 – Ambientes de software	28
3.3 – CLASSIFICAÇÃO DOS EQUIPAMENTOS EM RELAÇÃO À EXPERIMENTAÇÃO REMOTA.....	28
3.3.1 – Equipamentos selecionados	30
3.4 – BASE DE CONCEITOS	32
3.4.1 – Seleção de elementos para se trabalhar conceitos.....	34
3.4.2 – O acesso aos meios.....	37
3.5 – CONCLUSÕES	39
CAPÍTULO 4 – ESPECIFICAÇÃO DE UM MODELO PARA LABORATÓRIO	
REMOTO DE CONTROLE.....	41
4.1 – ARQUITETURA.....	41
4.1.2 – Formulação do modelo.....	42
4.2 – RESUMO DAS ESPECIFICAÇÕES DO MODELO.....	47
4.3 – ESPECIFICAÇÃO DA ARQUITETURA.....	48
4.4 – CONCLUSÕES	50
CAPÍTULO 5 – MATERIALIZAÇÃO DO MODELO.....	52
5.1 – IMPLEMENTAÇÃO	52
5.2 – MÓDULOS	53
5.2.1 – Cliente.....	53
5.2.2 – Servidor central.....	54
5.2.3 – Servidor auxiliar – Gerenciador de experiência.....	56
5.2.4 – Programa Aplicativo.....	56
5.2.5 – Aplicativo Administrador.....	58
5.3 – INSTALAÇÃO FÍSICA	59
5.4 – USO DO APLICATIVO CLIENTE E RESULTADOS	60
5.5 – CONCLUSÕES	70
CAPÍTULO 6 – CONCLUSÕES.....	71
6.1 – CONCLUSÕES	71
6.2 – SUGESTÕES PARA TRABALHOS FUTUROS.....	74

REFERÊNCIAS.....	75
ANEXO A – SISTEMAS DE CONTROLE	81
ANEXO B – SINAIS DE TESTE TÍPICOS.....	85
ANEXO C – MODELO MATEMÁTICO DE SISTEMAS FÍSICOS.....	86
ANEXO D – REPRESENTAÇÕES DE MODELOS MATEMÁTICOS DE SISTEMAS FÍSICOS	90
ANEXO E – IDENTIFICAÇÃO DE SISTEMAS.....	93
ANEXO F – MODELO MATEMÁTICO DE UM SERVOMOTOR.....	95
ANEXO G – EXPERIÊNCIA 01: IDENTIFICAÇÃO DE UM MOTOR DC.....	99
ANEXO H – EXPERIÊNCIA 02: CONTROLE DE POSIÇÃO DE UM MOTOR D.C.....	102
ANEXO I – EXPERIÊNCIA 03: CONTROLE DE POSIÇÃO DE UM MOTOR D.C.	105
ANEXO J – PROGRAMA ADMINISTRADOR.....	108

LISTA DE FIGURAS

Figura 1 – Laboratório de experimentação remota genérico.....	16
Figura 2 – Treinamento com amplificadores operacionais.	23
Figura 3 – Simulador analógico.....	24
Figura 4 – Servomecanismo: unidade eletromecânica e acessórios.	24
Figura 5 – Sistema térmico.....	25
Figura 6 – <i>Fun Plate</i>	25
Figura 7 – <i>Ball Beam</i>	26
Figura 8 – Esteira transportadora.....	26
Figura 9 – Pêndulo invertido.	27
Figura 10 – Sistema massa-mola-amortecedor.....	27
Figura 11 – Base de conceitos.	33
Figura 12 – Arquitetura para realização dos experimentos.....	41
Figura 13 – Arquitetura geral.	49
Figura 14 – Módulos componentes do sistema implementado.....	53
Figura 15 – Detalhes do servidor central.	54
Figura 16 – Sistema de referência para o deslocamento angular do eixo do motor real.	58
Figura 17 – <i>Login</i> no sistema.	60
Figura 18 – Tela principal do programa cliente.....	61
Figura 19 – Janela do modo laboratório remoto.....	62
Figura 20 – Selecionando uma planta.	63
Figura 21 – Selecionando uma experiência.....	64
Figura 22 – Estado inicial e parâmetros para um experimento.	65
Figura 23 – Definindo uma <i>function</i> : relé com zona morta.	66
Figura 24 – Identificação da experiência.	66
Figura 25 – Resultados de uma experiência em forma de tabela.	67
Figura 26 – Resultados de uma experiência na forma de gráficos.	68
Figura 27 – Animação do experimento realizado.....	68
Figura 28 – <i>Function</i> com erro.....	69
Figura 29 – Mensagem de erro do compilador.....	69
Figura 30 – Sistema de controle em malha fechada.....	81
Figura 31 – Controlador PID.....	82

Figura 32 – Diagrama de blocos para o problema de regulação.....	83
Figura 33 – Diagrama de blocos para o problema de rastreamento.....	84
Figura 34 – Sinal degrau.	85
Figura 35 – Sinal onda quadrada.	85
Figura 36 – Sinal senoidal.	85
Figura 37 – Representação de sistemas físicos.....	86
Figura 38 – Sistema translacional mecânico.	87
Figura 39 – Sistema rotacional mecânico.....	87
Figura 40 – Circuito L-R.	88
Figura 41 – Diagrama de blocos do sistema translacional mecânico.	91
Figura 42 – Servomotor.....	93
Figura 43 – Constante de tempo.	94
Figura 44 – Esquema de um motor D.C. com excitação independente.	95
Figura 45 – Diagrama de blocos do servomotor da <i>Feedback</i>	97
Figura 46 – Sinais lidos dos sensores e sinal enviado para o motor.	98
Figura 47 – Motor D.C. em malha aberta.....	99
Figura 48 – Modelo para a experiência 1.	100
Figura 49 – Modelo para a experiência 2.	102
Figura 50 – Controle de posição com ação de controle PID.	103
Figura 51 – Modelo para a experiência 3.	105
Figura 52 – Controle de posição com ação de controle qualquer.	106
Figura 53 – Relé com zona morta.	106
Figura 54 – Janela principal do administrador.	108
Figura 55 – Janela para inclusão/exclusão de usuários.	108
Figura 56 – Tarefas realizadas pelo usuário.	109
Figura 57 – Janela para inclusão/exclusão de experimentos mostrando os três experimentos que se pode realizar com o motor via simulação.....	109
Figura 58 – Janela para inclusão/exclusão de experimentos mostrando os três experimentos que se pode realizar com o motor real	110
Figura 59 – Definição dos nomes/tipos de variáveis/parâmetros de um experimento.....	110
Figura 60 – Continuação da Figura 59.....	111
Figura 61 – Protótipo da <i>function</i> disponibilizada ao usuário.	111
Figura 62 – Janela do gerenciador de experiências para o experimento do PC5 do arquivo de inicialização “LabRem.ini”	114

LISTA DE TABELAS

Tabela 1 – Classificação em termos dos tipos de transferências	19
Tabela 2 – Classificação em termos dos tipos de acessos	19
Tabela 3 – Resumo dos laboratórios remotos analisados.....	20
Tabela 4 – O modelo proposto e os laboratórios remotos analisados.....	51
Tabela 5 – Determinação de parâmetros do motor e sensores	101
Tabela 6 – Listagem para implementação de um relé.....	106
Tabela 7 – Arquivo de inicialização atual do servidor.....	112
Tabela 8 – Um arquivo de inicialização.....	114
Tabela 9 – Listagem de um aplicativo.	115

RESUMO

Este trabalho propõe um modelo instrucional como um instrumento auxiliar no ensino e aprendizagem do conjunto de disciplinas que compõem a área de sistemas de Controle. Tal modelo incorpora mídias e tecnologias instrucionais, incluindo a *Web*, ao repertório pedagógico do educador que atua nessa área.

O objetivo é explorar o compartilhamento de recursos de um laboratório de Sistemas de Controle, tanto de *software* quanto de *hardware*, ampliando tanto a audiência atendida quanto seu tempo de uso, uma vez que os experimentos, no modelo, são realizados de forma remota (não presencial), possibilitando um uso mais racional desses recursos, com a conseqüente redução de custos.

Tratando-se de um sistema aberto, sob o ponto de vista da Teoria Geral de Sistemas, o modelo permite a inclusão de novos experimentos, na medida em que os mesmos puderem ser disponibilizados ao projetista, bem como uma atualização dos já utilizados. Assim, novas mídias e tecnologias podem ser incorporadas, não somente para ampliar os recursos dos experimentos já implementados, mas também para permitir a inclusão de novos experimentos.

Para ilustrar a viabilidade do modelo, um protótipo é apresentado, o qual contém experimentos práticos sobre identificação e teste de algoritmos de controle num equipamento real, no caso, um servo mecanismo.

ABSTRACT

This work proposes an instructional model as an auxiliary instrument for teaching and learning disciplines that compose control system area. Such model incorporates medias and instructional technologies, including Web, to the pedagogical repertory of the educator who works on this area.

The objective is to share the resources from a control system laboratory, both in software and hardware, amplifying the number of users and the time of utilization, since the experiments are realized in a remote way, allowing a more rational use of these resources and cost reduction.

Like an open system, under general system theory's point of view, the model admits the inclusion of new experiments, in proportion they can be disposed to the projectionist, as well as an up dating from those ones already used. So, new medias and technologies can be incorporated not only to amplify the capability of experiments already implemented, but also to permit the inclusion of new ones.

In order to show the model viability, a prototype is presented, which has experiments on system identification and control algorithm testing using a real equipment or simulation.

CAPÍTULO 1 – INTRODUÇÃO

O trabalho aqui apresentado encontra-se na confluência das seguintes áreas de pesquisa: Redes de computadores, Educação a distância, Experimentação remota, e Sistemas de Controle. Dentro destas áreas, procura-se inicialmente delimitar suas reais fronteiras.

As redes de computadores, em particular a *Internet*, têm-se mostrado uma ferramenta poderosa, permitindo dentre outras coisas, um compartilhamento mais racional de recursos e que vêm adquirindo um maior número de usuários a cada dia. A interconexão de computadores em grande escala só foi possível após a inclusão de um conjunto de padrões de rede, especificando os detalhes da comunicação entre computadores e do roteamento das mensagens, denominado de pilha de protocolos de interligação em redes TCP/IP (*transmission control protocol/internet protocol*), sobre os quais são definidos vários outros serviços como *e-mail*, transferência de arquivos, *web* e outros, com protocolos próprios, que utilizam o TCP/IP como protocolo de transferência de mensagens [1]. Hoje, diversos aplicativos estão disponíveis em vários setores da sociedade, por exemplo: comercial (comércio eletrônico), econômico (bancos) e educacional (educação a distância).

Em relação à educação a distância, novas possibilidades, desafios e incertezas surgiram com a *Internet* no processo ensino – aprendizagem. Se por um lado não se pode esperar das redes a solução mágica para modificar profundamente a relação pedagógica, por outro lado, elas facilitam, como nunca antes, a pesquisa individual e em grupo e o intercâmbio entre professores, alunos e professores e alunos [2]. Ensinar a distância é muito diferente de ensinar presencialmente, são diferentes habilidades de apresentação da informação e de planejamento [3]. Um grande desafio da escola e demais ambientes educacionais é saber utilizar as diferentes mídias e tecnologias no ensino, assim como conduzir e orientar os alunos na avalanche de informações disponíveis em todos os contextos de ambientes de ensino – aprendizagem. O objetivo é, portanto, que a educação entre em “sintonia” com o mundo tecnológico atual para satisfazer e atender a sociedade [4].

Uma situação específica da educação a distância com a *Internet*, diz respeito a práticas de laboratório, comuns e necessárias em diversas áreas, particularmente nas engenharias. Um trabalho pioneiro neste sentido foi o laboratório virtual da Universidade de *Carnegie Mellon*. Neste caso, disponibilizou-se alguns instrumentos reais de laboratório para a realização de

experimentos de forma remota, ou seja, os experimentos podiam ser feitos via *Internet*. Este novo paradigma foi denominado então de experimentação remota [5].

Atualmente a experimentação remota tem sido aplicada noutras áreas do conhecimento como em medicina [6] e robótica [7], [8]. Por outro lado, novas aplicações deste paradigma na área de educação a distância surgiram e continuam a surgir, como o treinamento em programação de microcontroladores [9] ou controle de plantas industriais [10], cada um dentro de um contexto específico, visando a atender as necessidades existentes nos mesmos.

Dentre os inúmeros contextos nos quais a experimentação remota pode ser aplicada, o enfoque de sua aplicação, neste trabalho, se refere a sistemas de controle e automação [11], [12], [13], mais especificamente, sobre práticas de laboratório voltadas ao ensino – aprendizagem nessa área do conhecimento. Esta escolha foi feita a partir de necessidades observadas num laboratório presencial, específico de sistemas de controle e que dizem respeito, fundamentalmente, à questão do acesso aos meios e equipamentos para a realização das práticas de laboratório.

Assim, delimita-se como fronteiras para o trabalho, o uso da experimentação remota para a realização de práticas de laboratório, visando o ensino – aprendizagem na área de sistemas de controle, e ainda, enquanto uma ferramenta alternativa para atender as necessidades de acesso aos meios e equipamentos de um laboratório de sistemas de controle específico.

Visando atender tais necessidades, este trabalho têm como objetivos gerais: apresentar um levantamento do estado da arte dos laboratórios de experimentação remota, no que se refere as aplicações mais expressivas do ponto de vista de ensino – aprendizagem em sistemas de controle e, desse levantamento, identificar as principais características desses laboratórios e estabelecer uma classificação para os mesmos. Como consequência desta classificação, mostrar que existem algumas lacunas relevantes a serem exploradas na área de controle, e que justificam o desenvolvimento de um novo laboratório.

Para o desenvolvimento de um novo laboratório remoto, este trabalho tem os seguintes objetivos específicos: fazer um estudo de caso de um laboratório presencial no sentido de avaliar os equipamentos de que dispõe e estabelecer um critério de classificação desses equipamentos em termos de potencial de uso, ou não, em aplicações de forma remota. Caracterizar melhor a questão de dificuldades de acesso aos meios e equipamentos e discutir alternativas de solução do problema, mostrando a viabilidade do uso da experimentação remota, como solução mais racional. Descrever uma base de conceitos representativa de

sistemas de controle, associada ao referido laboratório presencial, e dela extrair os principais conceitos envolvidos. Relacionar tais conceitos com um conjunto mínimo de equipamentos, com potencial para experimentação remota, que permitam que estes conceitos possam ser trabalhados no processo ensino – aprendizagem. Estabelecer um modelo com as seguintes características: permitir explorar ao máximo os diversos aspectos relacionados à base de conceitos, sem deixar de levar em consideração as questões relativas à implementação e voltado ao ensino – aprendizagem. E finalmente, descrever os principais componentes de um protótipo que implementa o modelo estabelecido e seus resultados.

O que se espera como primeiro resultado, da estratégia adotada para o estabelecimento do modelo e sua implementação, é um laboratório remoto voltado para o ensino – aprendizagem de sistemas de controle que, além de preencher as lacunas mencionadas nos objetivos gerais, avance mais um pouco, propiciando mais alternativas de experimentos aos seus usuários finais, como por exemplo:

- permitir tanto experimentos com equipamentos reais como simulados;
- experimentos em malha aberta ou malha fechada;
- tanto experimentos com controladores contínuos como discretos;
- experimentos específicos, como regulação ou rastreamento, incorporando um certo grau de autonomia para garantir o pré estabelecimento de condições iniciais;
- uso de estruturas de controle pré-definidas ou definidas pelo usuário, e neste último caso, possibilitando explorar rudimentos de programação com uma determinada linguagem.

Além disso, propiciando mais informações ao usuário como: mensagens provenientes de um compilador ou geradas pelo próprio laboratório; resultados dos experimentos na forma de tabelas, gráficos ou animações; textos auxiliares sobre sistemas de controle, experimentos e uso do próprio laboratório; permitir com certa facilidade a inclusão de novos experimentos e, principalmente, que atenuem suas dificuldades de acesso aos meios.

O trabalho encontra-se portanto estruturado nos seguintes capítulos, além deste: no capítulo 2 trata-se da experimentação remota, sua definição, características estado da arte e classificações, o capítulo 3 refere-se ao estudo de casos de um laboratório presencial, no capítulo 4 o estabelecimento do modelo é descrito, no capítulo 5 apresenta-se os módulos do protótipo desenvolvido e no capítulo 6 são apresentados os comentários, conclusões e sugestões

CAPÍTULO 2 – LABORATÓRIO DE EXPERIMENTAÇÃO REMOTA

Neste capítulo conceitua-se laboratório de experimentação remota, mostra-se sua estrutura genérica e aponta-se algumas de suas vantagens. Em seguida, apresenta-se um levantamento do estado da arte, no que se refere a sistemas de controle, e a partir deste, são identificadas suas principais características. Baseado nestas características, procura-se estabelecer uma classificação para os laboratórios de experimentação remota. Detecta-se nesta classificação duas lacunas no uso do conceito de laboratório remoto. Estas lacunas propiciam uma primeira contribuição deste trabalho, pois percebe-se nela o uso ainda restrito de todo potencial do conceito de experimentação remota, e ainda, do ponto de vista de sistemas de controle, entende-se que o preenchimento dessas lacunas seja imprescindível.

2.1 – Conceito

Laboratório de experimentação remota consiste num agrupamento de instrumentos de propósito geral, conectados a um conjunto de sistemas de computadores pessoais, conectados à *Internet* [14]. Na Figura 1 ilustra-se o conceito.

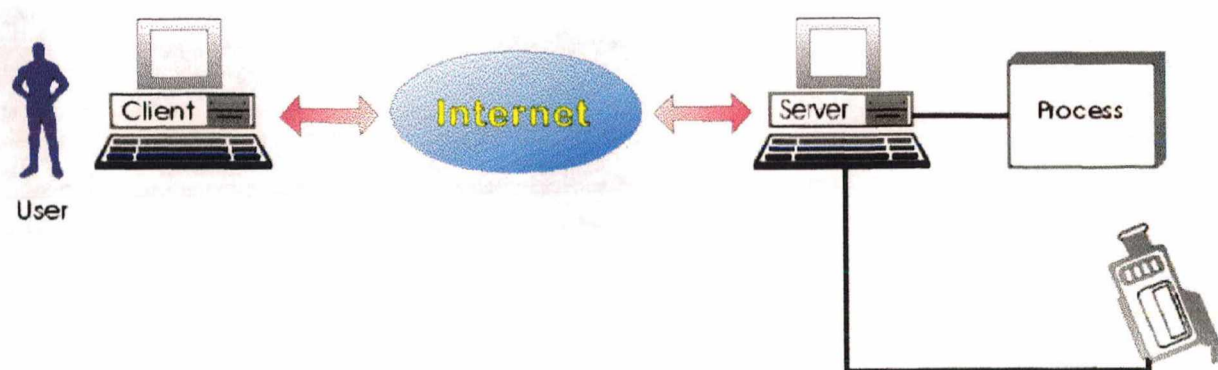


Figura 1 – Laboratório de experimentação remota genérico.

Este paradigma apresenta algumas características importantes:

- Permitir interações com o mundo físico, o que garante que os resultados obtidos são os mesmos que se obteria localmente;
- Não exclui a possibilidade de interações com simuladores;

- Permite que os usuários tenham acesso a recursos que não possuem localmente e o número de usuários pode ser ilimitado;
- Custo reduzido para a realização de experimentos reais;
- Experimentos podem ser realizados a qualquer hora e de qualquer lugar, desde que se tenha um acesso à *Internet*;
- Possibilita uma utilização mais ampla das redes disponíveis nas universidades e escolas técnicas brasileiras;
- O usuário pode ganhar sentimento prático, mesmo não estando no laboratório para a realização de experimentos.

2.2 – Estado da Arte

Diversos laboratórios de experimentação remota vêm sendo desenvolvidos nos últimos anos, alguns dos quais são relacionados a seguir, com uma breve descrição dos mesmos.

2.2.1 – Laboratório de sistemas dinâmicos e controle da Universidade do Texas

Permite o treinamento em simulação e tecnologias de simulação. Apresenta sistemas dinâmicos, como um pêndulo submerso ou dois tanques interligados e códigos de programa para serem executados no ambiente do Matlab [15].

2.2.2 – Laboratório de experimentação remota da Universidade de *Carnegie Mellon*

Permite experimentos com circuitos eletrônicos reais isolados, ou para controle de um motor. Uma câmera de vídeo coloca a disposição imagens dos equipamentos em uso. Alguns circuitos são restritos a alunos matriculados em determinadas disciplinas [5].

2.2.3 – Laboratório de experimentação remota da Universidade de *Western Austrália*

É constituído de um robô industrial e permite a manipulação de objetos com o uso de uma garra. O usuário pode especificar as coordenadas desejadas para o movimento e a operação de abrir ou fechar a garra. Imagens de vídeo do robô são disponíveis [16].

2.2.4 – Laboratório de experimentação remota da Universidade de Brasília

É composto por um manipulador de dois graus de liberdade, ao qual está acoplada uma câmera de vídeo. Permite que se visualize áreas específicas de um ambiente pela movimentação do manipulador. As coordenadas do movimento são selecionadas pelo usuário [17].

2.2.5 – RExLab – O Laboratório de Experimentação Remota da Universidade Federal de Santa Catarina

É constituído de uma placa que contém um microcontrolador 8051 e permite que programas em *assembly* sejam executados no 8051. Extensões foram feitas para o microcontrolador PIC17C43 e os resultados gerados podem ser visualizados por meio de uma câmera [18].

2.2.6 – Laboratório da Universidade do *Tennessee- Chattanooga*

É constituído de várias plantas reais, e permite que se realize experimentos de identificação ou controle. Pode-se especificar o tipo de sinal de comando e os parâmetros de um controlador PID (ANEXO A). Apresenta seus resultados na forma de gráficos ou dos dados gerados no experimento. Dispõe ainda de um vasto conjunto de textos relativos a Teoria de Controle e dos experimentos em si. Utiliza o *LabView* para aquisição de dados e *software* de controle. Permite tanto uso local quanto remoto [19].

2.3 – Análise dos Laboratórios de Experimentação Remota

A análise a ser apresentada em seguida diz respeito a realização de experimentos remotos usando os laboratórios descritos no item 2.2. Para tanto a seguinte classificação foi estabelecida:

- I) Natureza da informação que trafega na rede (*Internet*)
 - I.1) transferência de parâmetros.
 - I.2) transferência de códigos de programas.

II) Natureza do experimento realizado

II.1) com sistemas reais.

II.2) com sistemas simulados.

A transferência de parâmetros significa, que o usuário pode enviar números, seja por exemplo os parâmetros K_p , T_i e T_d de um controlador PID (ANEXO A), ou ainda, os parâmetros amplitude e período de um comando do tipo onda quadrada (ANEXO B).

Já a transferência de códigos de programas significa que o usuário pode enviar trechos de programas (*functions*) ou programas completos (*main* e suas *functions*) escritos em alguma linguagem de programação específica.

Cabe ao servidor tratar estas informações de forma adequada, ou seja, se a ocorrência foi envio de parâmetros estes simplesmente serão atribuídos à variáveis previamente definidas para recebê-los. Se a ocorrência foi o envio de código de programa então um compilador deve ser ativado para tratar este código. De forma análoga, se o experimento é com um equipamento real o servidor deve ativá-lo, se for com um sistema simulado deve ativar o simulador do respectivo sistema.

Da classificação anterior resulta o seguinte quadro, ilustrado nas Tabelas 1 e 2, para os laboratórios do item 2.2.

Tabela 1 – Classificação em termos dos tipos de transferências.

Laboratório	Texas	Mellon	Western	UnB	UFSC	Tennessee
Transferência de parâmetros		✓	✓	✓		✓
Transferência de programas					✓	

Tabela 2 – Classificação em termos dos tipos de acessos.

Laboratório	Texas	Mellon	Western	UnB	UFSC	Tennessee
Acesso a plantas reais		✓	✓	✓	✓	✓
Acesso a plantas simuladas	✓					

Para uma descrição mais completa dos laboratórios remotos, são consideradas também as diferentes mídias implementadas em cada um, conforme ilustrado na Tabela 3.

Tabela 3 – Resumo dos laboratórios remotos analisados.

Laboratório	Texas	Mellon	Western	UnB	UFSC	Tennessee
Transferência de parâmetros		✓	✓	✓		✓
Transferência de códigos					✓	
Sistemas reais		✓	✓	✓	✓	✓
Sistemas simulados	✓					
Textos (html)	✓	✓	✓	✓	✓	✓
Resultados: gráficos						✓
Resultados: numéricos						✓
Imagens de Vídeo		✓	✓	✓	✓	
Animações			✓			

2.4 – Conclusões

Das Tabelas 1 e 2, nota-se que existem duas lacunas nos laboratórios de experimentação remota apresentados. Em relação ao tipo de transferência (Tabela 1), nota-se que nenhum permite a transferência tanto de parâmetros como de programas e quanto ao tipo de acesso (Tabela 2), nenhum permite a possibilidade de acesso tanto a plantas reais como a plantas simuladas. Além do que, da Tabela 3 observa-se que as mídias não são totalmente exploradas, particularmente no que se refere a imagens e animações.

A possibilidade de transferência tanto de parâmetros quanto de programas, ou de acesso a plantas reais ou simuladas, num mesmo laboratório remoto, muito mais do que simplesmente sugerir um laboratório mais completo, tem um caráter prático muito importante do ponto de vista de sistemas de controle em termos de ensino – aprendizagem, conforme será discutido no capítulo 3. Já em relação às mídias, uma melhor exploração destes recursos também está relacionada diretamente as questões de aprendizagem.

CAPÍTULO 3 – LABORATÓRIO PRESENCIAL

Neste capítulo é feito um estudo de caso, objetivando caracterizar em que pontos a experimentação remota pode contribuir no domínio de conhecimentos Sistemas de Controle. Considera-se como objeto deste estudo o Laboratório de Controle e Sistemas do Departamento de Engenharia Elétrica e de Computação da Universidade Federal do Pará (LACOS/DEEC/UFGA) [20].

Para o LACOS os seguintes aspectos são evidenciados: seus objetivos, usuários e a dinâmica existente no laboratório para atender seus objetivos. Faz-se o levantamento de seus equipamentos e meios específicos existentes para realização de experimentos nesta área do conhecimento.

Propõe-se uma classificação para avaliar quais equipamentos do LACOS podem ser incorporados ao conceito de experimentação remota e apresenta-se uma base de conceitos associada a Sistemas de Controle, com a qual seus usuários lidam no desenvolvimento de suas atividades.

Identifica-se então, um conjunto mínimo de elementos na base de conceitos, a partir dos quais se pode trabalhar de forma consistente os conceitos básicos da teoria de controle e por fim, compara-se o uso das abordagens presencial e remota, quanto ao uso desses elementos mínimos, sob o ponto de vista de custos, segurança, ensino – aprendizagem e compartilhamento de equipamentos visando ao atendimento dos objetivos do laboratório

3.1 – Objetivos do LACOS/DEEC/UFGA

Este laboratório presencial tem como principais objetivos dar suporte aos cursos de graduação e pós-graduação em engenharia elétrica da UFGA, na área de sistemas de controle e automação. Seus principais usuários são:

- grupo I: alunos do curso de graduação matriculados nas seguintes disciplinas:
 - grupo Ia: laboratório de sistemas de controle;
 - grupo Ib: análise de sistemas lineares;
 - grupo Ic: sistemas de controle I;
 - grupo Id: sistemas de controle II;

- grupo II: alunos do curso de pós-graduação (mestrado e doutorado);
- grupo III: bolsistas de iniciação científica;
- grupo IV: alunos desenvolvendo trabalho de conclusão de curso;
- grupo V: professores da área de sistemas de controle.

3.1.1 – Dinâmica do laboratório de sistemas de controle

A descrição a seguir esclarece de que maneira ocorre o compartilhamento de equipamentos do laboratório em relação a seus principais usuários e as dificuldades que este compartilhamento acarreta.

O laboratório tem a seguinte dinâmica: durante quatro dias da semana, seus equipamentos são disponibilizados para qualquer um dos usuários mencionados no item 3.1, e durante um dia da semana, restringe-se o uso de um conjunto de equipamentos aos alunos da disciplina laboratório de sistemas de controle (grupo Ia). Nesse dia, a cada duas horas, grupos distintos de vinte alunos, realizam quatro experimentos pré definidos dividindo-se portanto, em grupos de cinco alunos por experimento. Como, usualmente um experimento não é realizado de forma completa em duas horas, resulta que são necessários os dois primeiros meses de um período letivo, para que cada grupo de cinco alunos realize o total de experimentos. Nos dois últimos meses, cada grupo de cinco alunos tem a oportunidade de rever os experimentos realizados (e os equipamentos) mais duas vezes antes do período de avaliação.

Como resultado da dinâmica anterior, cada um dos grupos de cinco alunos tem contato assegurado com os equipamentos de um experimento, no máximo quatro vezes num período letivo.

Como fora do horário que lhes é reservado, os equipamentos podem estar (e normalmente estão) sendo utilizados pelos outros usuários do laboratório, os alunos além de trabalhar em número de cinco por equipamento, o que do ponto de vista de aprendizagem pode ser comprometedor, só têm contato com o equipamento em quatro oportunidades.

Mais prejudicados que estes alunos, são ainda os pertencentes aos grupos: Ib, Ic e Id do item 3.1, pois a eles não é estipulado nenhum horário.

3.2 – Equipamentos

Visando atingir seus objetivos, o laboratório de sistemas de controle da UFPA dispõe hoje de um conjunto de equipamentos listados abaixo, com uma breve descrição de suas principais características:

3.2.1 – Treinamento com amplificadores operacionais

Permite a construção de blocos básicos de controle como: somador, inversor, e sistemas de primeira ou segunda ordem, pela conexão adequada de resistores e/ou capacitores a amplificadores operacionais, e destes últimos entre si [21]. Na Figura 2 mostra-se a parte superior (pontos de conexão) desse.

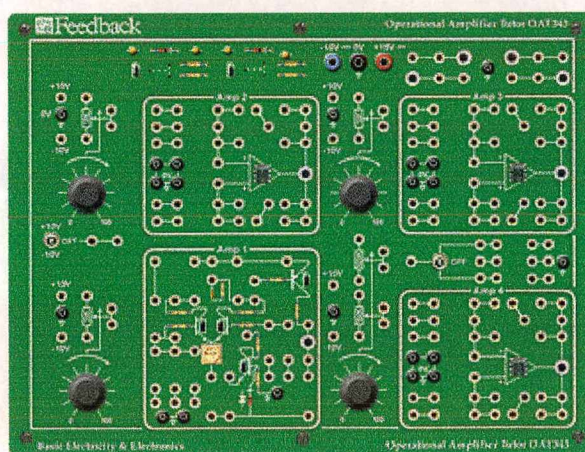


Figura 2 – Treinamento com amplificadores operacionais.

3.2.2 – Simulador para controle de processos

Possui blocos básicos de controle pré-definidos, por exemplo: somador, inversor, sistema de primeira ordem, integrador, controlador PID, não linearidades (saturação, *backlash*) e atraso, que podem ser conectados entre si [22]. Desta forma, permite a montagem de sistemas em malha fechada (ANEXO A) tendo: controlador do tipo PID, uma planta de até terceira ordem, com atraso puro e não-linearidades. O equipamento é mostrado na Figura 3.

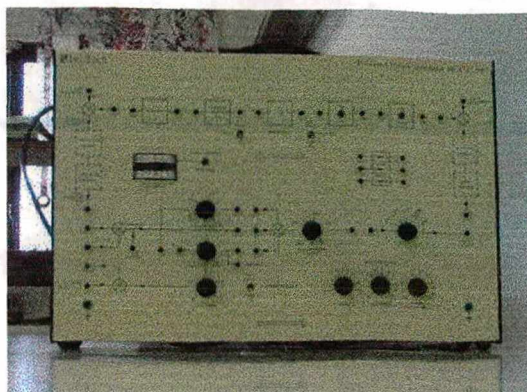


Figura 3 – Simulador analógico.

3.2.3 – Treinamento com servomecanismo

É constituído dos seguintes elementos: uma unidade eletromecânica ilustrada na Figura 4, que contém basicamente: um motor D.C. conectado a um sistema de correias com redução de velocidade de 1/32; um sensor de posição angular e um sensor de velocidade angular (tacômetro), conectados, respectivamente, num eixo após a redução de velocidade e diretamente no eixo do motor; um sensor de velocidade angular (foto-sensor) ligado após a redução de velocidade; além de um freio eletromagnético operado manualmente, que pode funcionar como perturbação externa. Há também uma unidade analógica que contém um amplificador de potência para acionamento do motor D.C., sinais de teste do tipo degrau, onda quadrada e senoidal (ANEXO B), podendo-se ajustar a frequência dos dois últimos, e um módulo de controlador do tipo PID com ganhos ajustáveis por potenciômetros, que permite operar o sistema em malha fechada [23].

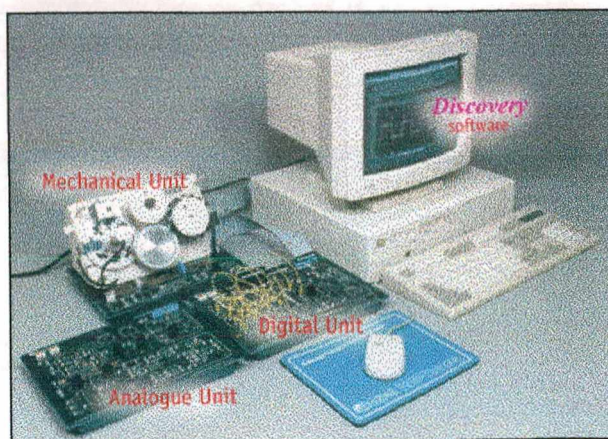


Figura 4 – Servomecanismo: unidade eletromecânica e acessórios.

3.2.4 – Sistema térmico

Permite o controle da temperatura no interior de um tubo que tem conectado, numa de suas extremidades, um ventilador e uma tela resistiva pela qual pode-se aplicar uma tensão variável para gerar um fluxo de ar quente, sendo a outra extremidade aberta. Pode-se conectar um sensor de temperatura em três posições na superfície do tubo. Efeitos de perturbações externas podem ser obtidos por uma obstrução manual momentânea da extremidade aberta [24].

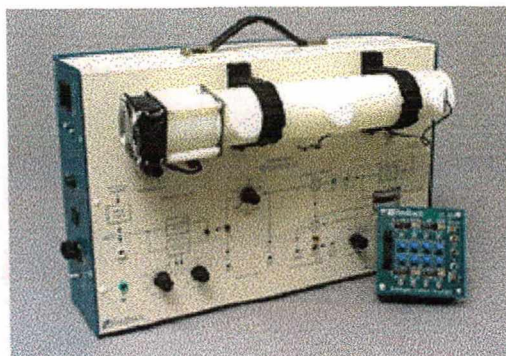


Figura 5 – Sistema térmico.

3.2.5 – Fun plate

É constituído de uma placa de madeira, que pode ser movimentada pela ação do vento proveniente de um ventilador e canalizado ao longo de um túnel. O acionamento do ventilador se faz por intermédio de um motor D.C. acionado por um modulador PWM (*Pulse Width Modulation*). Próximo à placa móvel existe um sensor para sua posição angular. Efeitos de perturbações externas podem ser obtidos pela abertura de janelas laterais situadas ao longo do túnel ou mesmo por toques manuais efetuados na placa móvel de madeira [25].

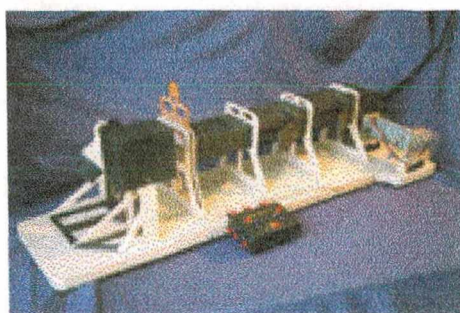


Figura 6 – Fun Plate.

3.2.6 – *Ball beam*

Este equipamento é constituído de uma esfera de aço que pode ser movimentada ao longo de uma barra. O acionamento da barra se faz por meio de um motor D.C. Três sensores estão disponíveis, um para a posição linear da bola, um para a posição angular e outro para a velocidade angular da barra [26].

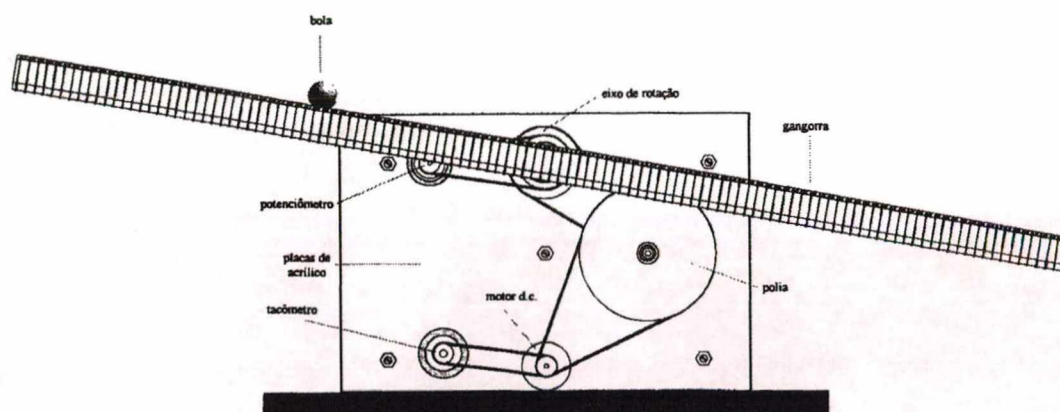


Figura 7 – *Ball beam*.

3.2.7 – Esteira Transportadora

Consiste numa esteira sobre a qual são posicionados recipientes vazios, que podem então ser movimentados por um motor D.C. conectado a esteira por um sistema de engrenagens. Um sensor ótico detecta a passagem dos recipientes e então, o movimento da esteira é interrompido para que nestes seja injetado água, bombeada de um reservatório, durante um período de tempo pré-definido [27]. O controle de cada evento é realizado por um CLP [28] e os recipientes, uma vez cheios, devem ser retirados manualmente no final da esteira.

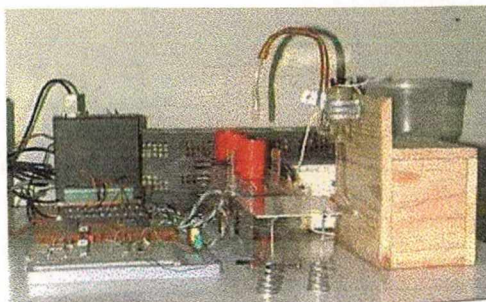


Figura 8 – Esteira transportadora.

3.2.8 – Pêndulo invertido

É constituído de duas hastes, uma vertical que pode ser rotacionada e outra horizontal que pode sofrer um movimento de translação. Ambas são acionadas por meio de motores de passo. Dispõe de sensores para medição da posição e velocidade angular das duas hastes. Acompanha o equipamento um *software* proprietário e uma placa de aquisição de dados permitindo que experimentos de controle ou identificação possam ser realizados a partir de um PC. Um dos principais experimentos é o controle da posição angular da haste vertical [29].

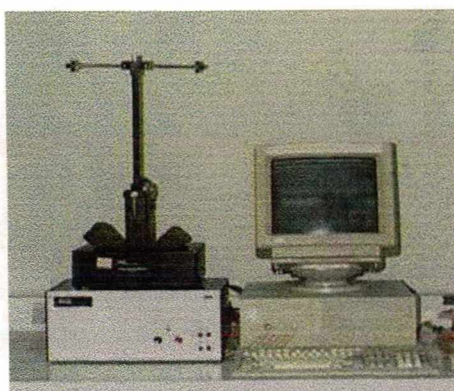


Figura 9 – Pêndulo invertido.

3.2.9 – Sistema massa-mola-amortecedor

Possui um conjunto de massas e molas que podem ser conectados entre si em série. Numa das extremidades da conexão, um motor de passo é conectado a uma mola e, na outra, um êmbolo pode ser conectado ao último bloco de massas. Sensores para medição da posição dos três blocos de massa estão disponíveis. Um *software* e uma placa, semelhantes aos disponíveis no módulo do pêndulo invertido, permitem a realização de experimentos a partir de um microcomputador. As massas e molas podem ser manualmente substituídas [30].

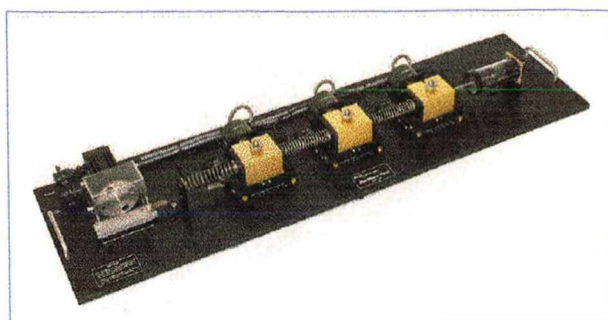


Figura 10 – Sistema massa-mola-amortecedor.

3.2.10 – Placa conversora AD/DA

Permite a aquisição e transmissão de sinais pelo uso de conversores analógico/ digital e vice-versa. Tem dois canais de conversão digital/analógico cada um com resolução de 12 bits e 16 canais de conversão analógico/digital. Trabalha acoplada a um computador pessoal e possui um *software* utilitário compatível com o Windows 98 ou superior. Dispõe de um conjunto de bibliotecas para as linguagens C, Pascal, BorlandC Builder e outras, a partir das quais pode-se fazer sua programação [31].

O laboratório dispõe ainda de equipamentos de instrumentação como osciloscópios e multímetros, além de microcomputadores e ambientes de *software*.

3.2.11 – Ambientes de *software*

Os ambientes disponíveis são: MATLAB/SIMULINK [32], SIMULACOS [33] e CLPLab [34]. O MATLAB/ SIMULINK, provê, dentre outras coisas, uma vasta biblioteca matemática, muito útil por exemplo para a realização de projeto de controladores, além de permitir a realização de simulações de sistemas dinâmicos, seja pelo desenvolvimento de um código de programa, ou pelo uso de diversos blocos pré-definidos que podem ser interligados entre si. O SIMULACOS também permite a simulação de sistemas dinâmicos usando o mesmo recurso de diversos blocos pré-definidos que podem ser interligados entre si. Finalmente, o CLPLab é um ambiente que permite, basicamente, o treinamento da linguagem por lista de instruções usada na programação de controladores lógico programáveis - CLPs.

3.3 – Classificação dos Equipamentos em relação à Experimentação Remota

Neste tópico apresenta-se uma classificação dos equipamentos existentes no LACOS, sob o ponto de vista de seus potenciais de uso em experimentos remotos e baseada em dois critérios:

a - Necessidade de intervenção do usuário

A necessidade de intervenção, diz respeito ao grau de manipulação ou intervenção manual do usuário no equipamento, para a realização de experimentos.

b - Flexibilidade tecnológica

Em relação ao *hardware*, a flexibilidade tecnológica diz respeito à facilidade que se tem para acoplar o equipamento a outros, ou ainda, se ele permite fácil acesso aos seus sinais de entrada e de saída. Em relação ao *software*, diz respeito ao acesso a códigos fonte, ou seja, se estes são disponíveis.

Para experimentos a serem realizados de forma remota, serão considerados equipamentos viáveis aqueles que apresentarem o menor grau de intervenção do usuário e maior flexibilidade tecnológica. Estes dois critérios foram adotados, considerando-se o fato de que experimentos remotos com equipamentos reais devam acontecer de forma autônoma [35], ou seja, independentemente da presença de um operador, logo, surgindo a necessidade de se incorporar tecnologias, seja de *hardware* ou de *software*, a fim de substituir a intervenção do usuário, deixando a este apenas a realização de aspectos específicos do experimento.

A seguir são caracterizados os equipamentos já descritos em relação a estes critérios:

- Treinamento com amplificadores operacionais e simulador para controle de processos.

Estes equipamentos podem ser caracterizados como tendo um elevado grau de intervenção posto que, neles o usuário deve fazer todas as conexões necessárias para a realização de um dado experimento. De fato são voltados exclusivamente para cursos presenciais, e nestes são de extrema importância, pois vão servir como base na formação de alunos iniciantes.

Portanto não são candidatos à experimentação remota, apesar de não apresentarem nenhuma dificuldade em relação à flexibilidade tecnológica

- Treinamento com servo mecanismo, sistema térmico e *fun plate*.

Os três equipamentos quando trabalhando conectados a uma placa conversora AD/DA, apresentam um grau de intervenção muito pequeno, neste caso, o usuário necessita ligar a fonte de alimentação, acessar um código de programa no PC que contém a placa e especificar neste código, por exemplo, o sinal de comando ou o algoritmo de controle que deseja implementar; noutras palavras, estabelece comandos ou um código de programa. Em relação a flexibilidade tecnológica não apresentam dificuldades. São portanto candidatos a experimentação remota.

- *Ball beam*.

O *ball beam*, mesmo nas mesmas condições descritas para os três equipamentos no item anterior, ainda assim apresenta um grau de intervenção elevado, devido ao fato de que a bola deve ser primeiro posicionada manualmente sobre a barra pelo usuário. Portanto, não é candidato, mesmo não apresentando dificuldades em relação a flexibilidade tecnológica.

- Esteira transportadora e pêndulo invertido.

Os dois equipamentos, têm um grau de intervenção elevado. A esteira transportadora devido a colocação e remoção manual dos recipientes sobre a mesma. O pêndulo, visto que este deve ser inicialmente posicionado manualmente na vertical pelo usuário.

Em relação à flexibilidade tecnológica, a esteira transportadora não apresenta nenhuma dificuldade, por outro lado, o pêndulo não permite fácil acesso por exemplo, aos sinais de seus sensores ou atuadores, e ainda, por ter disponível apenas a versão executável de seus programas. Assim os dois equipamentos não são candidatos à experimentação remota.

- Sistema massa-mola-amortecedor.

Este equipamento só apresenta grau elevado quando se considerada a substituição das massas ou das molas, pois é feita manualmente. Neste sentido é candidato, guardando-se a restrição de limitação de variações de experimentos que poderia realizar, pois apenas um conjunto único de massas e molas seria utilizado. Por outro lado, têm as mesmas características do pêndulo em relação à flexibilidade, logo não é candidato.

- Placa conversora AD/DA.

Uma vez conectada a um PC e a algum equipamento real, o grau de intervenção do usuário se reduz a especificar (inserir) códigos de programa para operá-la. Quanto à flexibilidade, por natureza é flexível, pois visa exatamente ser interligada a dispositivos diversos, como sensores, atuadores, etc., então é candidata.

3.3.1 – Equipamentos selecionados

Do exposto no item 3.3, os seguintes equipamentos podem ser selecionados para utilização de forma remota: o servomecanismo, o sistema térmico e o *fun plate*, desde que, conectados a uma placa de aquisição de dados.

Ressalta-se ainda, que o grau de intervenção nestes equipamentos se eleva quando consideram-se as condições iniciais para a experiência ou o efeito de perturbações externas, como se observa a seguir.

Para o servomecanismo, o usuário, antes de iniciar seu experimento propriamente dito, deve ajustar manualmente a posição angular inicial desejada do eixo de saída do motor. Caso deseje impor alguma perturbação durante o experimento esta também é feita manualmente.

Para o sistema térmico a temperatura inicial sempre será a temperatura interna do equipamento no ponto onde está situado o sensor, e este, por sua vez, é posicionado manualmente. Caso o usuário queira aplicar uma perturbação, esta também é feita manualmente, por exemplo, passando rapidamente sua mão em frente a extremidade aberta do equipamento.

O *fun plate* não apresenta dificuldades em relação às condições iniciais, pois a placa sempre estará na vertical, já efeitos de perturbação só podem ser obtidos manualmente com ligeiros toques na placa.

Sob o ponto de vista de experimentação remota estas observações relativas às condições iniciais e aos efeitos de perturbações externas, incorporam um maior grau de complexidade à solução do problema, ou seja, à realização de experimentos de controle com equipamentos reais, caso se deseje reproduzi-las de forma remota.

Por fim verifica-se que plantas “desafiadoras” do ponto de vista de controle como o *ball beam* e o pêndulo invertido foram excluídas, pela classificação adotada, ficando seu uso restrito a um laboratório presencial. Na realidade, este fato pode ser extrapolado quando se considera que, em muitos casos é simplesmente impossível se dispor de um sistema real para realização de experimentos. Como exemplo, tem-se os casos de sistemas elétricos de potência [36] e de foguetes [37], que nos permitem no máximo, mesmo em centros de pesquisa que desenvolvem estes tipos de sistemas, experimentos híbridos, ou seja, envolvendo alguns equipamentos reais integrados com simuladores, mas nunca experimentos em laboratório, envolvendo o sistema como um todo. Por outro lado, a investigação do desempenho de algoritmos de controle em sistemas dessa natureza, via simulação, tem importância não apenas acadêmica bem como de pesquisa, e muitas vezes com reflexos econômicos significativos, como é o caso do controle da geração e transmissão de energia em sistemas elétricos de potência [38].

Desta forma, entende-se que, no que se refere a sistemas de controle, seja em ensino ou pesquisa, a experimentação remota necessariamente deve ser explorada de modo que se assegure, simultaneamente, o acesso a sistemas reais e simulados, visto que, nem sempre será

possível realizá-la com qualquer equipamento real, seja pelas limitações de flexibilidade tecnológica, do grau de intervenção do usuário ou mesmo pelo fato de que alguns destes equipamentos reais devam ser utilizados de forma absolutamente presencial (como os dos itens 3.2.1 e 3.2.2).

3.4 – Base de Conceitos

Neste tópico descreve-se, numa visão macroscópica, com que conteúdos um usuário genérico do laboratório de sistemas de controle pode lidar, ou por outra, qual a base de conceitos que se relaciona com a teoria de controle, além dela própria. A Figura 11 ilustra, senão todos os conteúdos, os que se considera mais relevantes. Nela, as fontes de problemas cujas soluções envolvem o uso de teoria de controle (ANEXO A) são diversas, passando deste à construção de protótipos didáticos [39], até o controle de sistemas aeroespaciais. Uma vez caracterizado o problema, os recursos humanos responsáveis por sua solução necessitam, usualmente, de um modelo matemático (ANEXO C) e uma representação adequada do modelo (ANEXO D) e os “produtos” (soluções do problema) que devem ser gerados podem ser: um modelo de simulação (ANEXO F), um algoritmo de identificação (ANEXO E) ou um controlador (ou algoritmo de controle). Para obtê-los, necessita-se trabalhar com recursos de *software* e *hardware*, além da teoria de controle. Este trabalho ocorre de modo cíclico, ou seja, o que se gera, seja em *software* ou *hardware*, antes de ser considerado um produto final acabado, sofre realimentações em relação aos recursos humanos responsáveis pela solução do problema, e muitas vezes com a própria fonte do problema afim de que se garanta um produto final bem acabado.

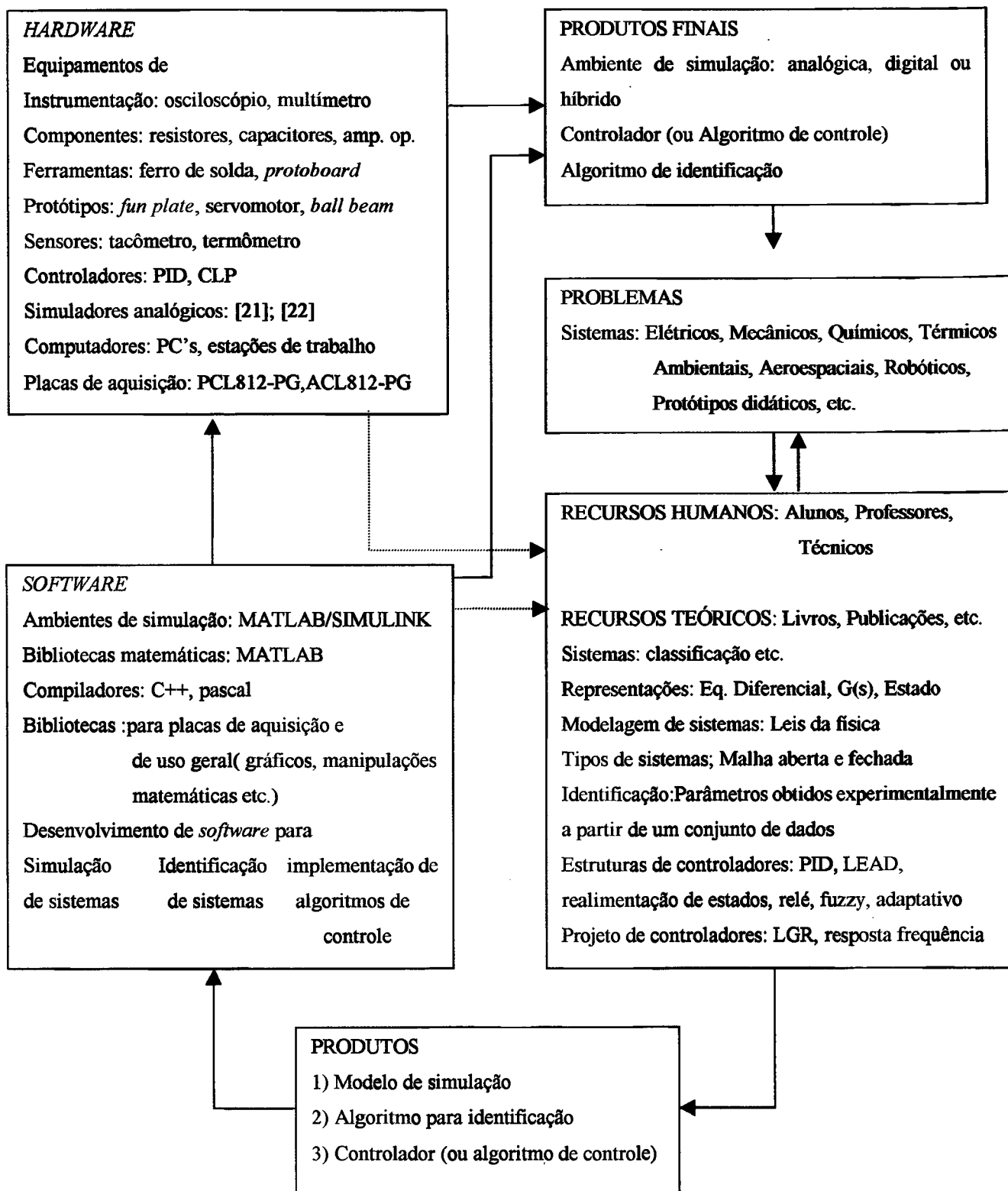


Figura 11 – Base de conceitos.

3.4.1 – Seleção de elementos para se trabalhar conceitos

A condição ideal, seria que todos os usuários do laboratório, pudessem participar em todas as etapas do ciclo da base de conceitos, contudo na prática isto não ocorre, particularmente com os usuários dos grupos Ia, Ib, Ic e Id do item 3.1 (alunos do curso de graduação matriculados nas disciplinas: laboratório de sistemas de controle, análise de sistemas lineares, sistemas de controle I e sistemas de controle II respectivamente), que além disso (não participam de todas as etapas) têm um tempo de participação bastante limitado. Portanto, considera-se fundamental identificar na base de conceitos um conjunto mínimo e necessário de elementos a partir dos quais se possa trabalhar de forma consistente os conceitos básicos da teoria de controle.

Para a seleção destes elementos, é importante ainda levar em consideração as tendências tecnológicas atuais. Neste caso, hoje existe uma grande utilização de tecnologia digital, na verdade uma substituição abrangente da tecnologia analógica pela digital. Portanto, este foi o ponto de partida para a seleção dos seguintes elementos:

- *hardware*: Computador, Placa de aquisição, Protótipo;
- *software*: Compilador, Bibliotecas ;
- recursos: Humanos e teóricos.

Supondo que este seja o conjunto mínimo de elementos necessários para se trabalhar os conceitos básicos da teoria de controle, seria ele suficiente ?

A resposta a esta pergunta não parece trivial, tendo em vista o terceiro item – recursos humanos e teóricos – mais precisamente o aspecto relativo aos recursos humanos, pois nele estão presentes: professor, aluno e o tempo, ou seja, os elementos essenciais que compõem o processo ensino – aprendizagem .

Não pretendendo esgotar, muito menos resumir a questão ensino – aprendizagem, mas antes disso, procurando estabelecer um referencial que subsidie uma possível resposta à pergunta formulada, concorda-se com Becker [40]:

“É necessário incluir as contribuições da Epistemologia Genética, na compreensão do ato educativo. Essas contribuições levantam o fato de que o professor trabalha com o conhecimento: o conhecimento constitui a matéria prima do seu fazer.”

“...A função da escola é, pois, garantir a produção e a transmissão do conhecimento. Isto é, cabe à escola fazer com que o conhecimento produzido pela humanidade chegue até o aluno. Cabe a ele apropriar-se desse patrimônio.

Isso impõe dois problemas:

1) Como selecionar os conhecimentos relevantes – quais são eles? – perante a gigantesca massa de conhecimentos existentes?

2) Um conhecimento nunca pode ser diretamente transmitido, por mais que a crença no processo de transmissão habite o imaginário da escola convencional – e do professor em particular.”

“...o conhecimento procede a partir, não do sujeito, nem do objeto, mas da interação entre os dois.”

“...a consolidação das conquistas que explicitam as condições fundamentais do ensino, comprometido com a concepção de aprendizagem humana como processo de construção. São três essas condições – condições necessárias! – sem as quais se comprometem esses avanços: a) todo o ensino deve partir do quadro atual do aluno b) deve incluir a fala do aluno, e outras formas equivalentes de expressão, em sua metodologia; e, finalmente, c) deve considerar o erro como componente do processo de construção do conhecimento e da aprendizagem.”

O autor ainda desenvolve um quarto ponto:

“d) pôr o aluno em interação com as ciências, as artes, a ética e a filosofia.”

E ainda concorda-se com Ramos [41]:

“... apesar das diferentes visões sobre o trabalho cooperativo, este traz muitos benefícios para os aprendizes. Dentre eles podem ser destacados: melhores níveis de desempenho e aprendizado; maior qualidade das estratégias de raciocínio; processos de solução novos e criativos; melhor transferência de aprendizado; oportunidade do compartilhamento de idéias e informações; maiores habilidades de comunicação, um contexto capaz de dar ao estudante mais controle sobre o seu próprio aprendizado; validação de idéias individuais e maneiras de pensar através da conversação; capacidade de perceber através de múltiplas perspectivas

(reestruturação cognitiva); e capacidade de argumentação (com resolução de conflitos conceituais).”

Partindo destes destas observações, identificamos inicialmente um protótipo para o item *hardware* dos elementos mínimos, que permita trabalhar com os conhecimentos mais relevantes da teoria de controle. Do conjunto de equipamentos selecionados, o candidato que apresenta este perfil é o servomecanismo. Vale dizer que os outros dois, *fun plate* e sistema térmico, não foram eleitos neste momento pelas seguintes razões: o *fun plate* é fortemente não linear na relação força do vento sobre a placa e o deslocamento angular da placa e ainda, sua dinâmica, no movimento ascendente e descendente da placa, se altera [42]. Já o sistema térmico apresenta um atraso de tempo significativo. Tais características, dos dois equipamentos, implicam no uso de técnicas de controle mais elaboradas, senão muito específicas. Por outro lado, no servomecanismo as não-linearidades existem, mas não tão acentuadas e, não têm atraso de transporte significativo, o que permite se utilizar de técnicas de controle simples, permitindo também o uso de técnicas de controle sofisticadas. [43].

Para os demais elementos dos itens *hardware* e *software* dos elementos mínimos, a placa de aquisição a ser selecionada pode ser a PCL812-PG [31] e suas bibliotecas, o compilador pode ser, por exemplo, o C++ 4.0 [44] e um PC com sistema operacional *Windows*.

Com estes equipamentos é possível explorar vários pontos da base de conceitos:

- a - aprendizagem de linguagem de programação;
- b – aprendizagem (uso)/desenvolvimento de pequenos simuladores de sistemas;
- c – aprendizagem (uso)/desenvolvimento de programas para placa de aquisição;
- d – aprendizagem (uso)/desenvolvimento de programas para interação com o servomecanismo em malha aberta (aplicar um sinal elétrico no motor e registrar sua resposta);
- e - determinação de parâmetros do motor (identificação) usando resultados do item “d”;
- f - dos resultados do item (e) projetar um controlador, por exemplo proporcional analógico depois um proporcional digital;
- g - evoluir o item (d) para programas de interação com o motor em malha fechada, ou seja, que incorporem o controlador projetado e testa-los;
- h - projetar outros tipos de controladores, por exemplo relé, relé com zona morta, PID, realimentação de estados e testa-los no item (g);
- i - avaliar o comportamento dos controladores diante de problemas específicos como

- regulação (manter a posição angular do eixo do motor em zero) e rastreamento (fazer com que a posição angular do eixo do motor acompanhe um perfil pré-estabelecido);
- j - projetar e avaliar o comportamento de controladores mais sofisticados usando por exemplo: controle *fuzzy*, adaptativo, etc.;
- k - aprendizagem/desenvolvimento de simuladores de sistemas mais complexos com os quais se deseje testar algum controlador específico;

Desta forma acredita-se ter estabelecido um conjunto mínimo de elementos que permitem uma ampla exploração da base de conceitos. Contudo, ainda não se tem uma solução bem definida sobre o tempo de acesso a estes elementos, ou seja, para os alunos matriculados na disciplina laboratório de sistemas de controle (grupo Ia do item 3.1).

3.4.2 – O acesso aos meios

Uma primeira abordagem para a questão do tempo de acesso a um conjunto mínimo de elementos, seria disponibilizar um conjunto mínimo exclusivamente para cada grupo de cinco alunos. Nesta abordagem quatro aspectos são relevantes:

- custos
- segurança
- ensino – aprendizagem
- uso mais racional de equipamentos (compartilhamento)

Dentre os elementos mínimos especificados, o que pode ser considerado mais caro é o servomecanismo (Figura 4). Contudo uma possível atenuação deste fator é a fabricação deste protótipo (quatro unidades) pelo próprio laboratório.

Quanto à segurança e ensino – aprendizagem, tem-se um quadro mais complexo posto que, neste caso uma das primeiras metas do professor é a conscientização dele próprio e dos alunos quanto ao manuseio dos equipamentos, afim de que se mantivesse a integridade física dos mesmos.

Outra questão importante é o agendamento do professor e dos alunos, em função de suas outras atividades. Uma estratégia possível é garantir, no dia e hora de aula da disciplina de laboratório, a participação do professor. Nos demais dias os alunos trabalhariam em grupos ou não, com a presença do professor ou não. Notando-se ainda que esta estratégia não exclui a

possibilidade do uso do conjunto mínimo de elementos pelos demais alunos do grupo I ou seja Ib, Ic e Id. O que também não excluiria as questões de segurança e agendamento já mencionadas.

Por fim, na questão do uso mais racional de equipamentos, ou ainda, de compartilhamento de equipamentos, certamente que esta abordagem, se por um lado pode resolver o problema específico do grupo I, por outro, não parece ser a mais racional. De fato, ela assegura o uso dos equipamentos a estes alunos durante um período de tempo maior, tal qual já é feito para os demais usuários do laboratório, então, se esta é uma solução, outra seria assegurar o uso dos equipamentos já existentes no laboratório para os alunos do grupo I, por um período maior, diminuindo o acesso dos demais usuários. Neste ponto, acredita-se ser necessário olhar o problema sob um ponto de vista mais geral, ou seja, o que está em questão é o direito de acesso aos meios relacionados a um determinado conhecimento, e garantir que os usuários do LACOS (incluindo os do grupo I) tenham este direito é uma obrigação, mas este direito deveria valer também para quem não é oficialmente um usuário. A rigor, deveria valer para qualquer aluno do curso de Engenharia Elétrica, ou mesmo para alunos não pertencentes à própria instituição de ensino, no caso a UFPA. Assegurar este direito, levando em consideração as questões de segurança, conscientização e agendamento é uma missão impraticável; neste sentido, não vislumbra-se um uso mais racional dos meios por esta abordagem.

Uma segunda abordagem para a questão do tempo de acesso a um conjunto mínimo de elementos, seria o uso da experimentação remota. Em relação aos custos, conforme mostrado na Figura 1 do capítulo 2, necessita-se de um computador a mais (servidor) a ser acoplado aos elementos mínimos já estabelecidos (placa de aquisição, servomecanismo e PC). Por outro lado só é necessário um servomecanismo, uma placa e um PC. Assim, tem-se inicialmente uma diminuição de custos correspondente a três PC's, três placas e três servomecanismos. A economia associada aos três PC's pode ser utilizada para a aquisição do servidor, e supondo que nem todos os alunos do grupo I tenham um computador próprio, a economia associada as três placas e aos custos de construção de três protótipos (supondo que os mesmos fossem construídos pelo próprio laboratório como na abordagem anterior), pode ser aplicada na compra de um PC, e este, por sua vez, ficar disponível a estes alunos de forma equivalente a dos elementos mínimos sugerida na primeira abordagem, observando-se que neste caso, este PC seria utilizado para acessar o protótipo de forma remota. Então, quando muito, os custos

seriam equivalentes (considerando-se que os PC's atuais já incluem placa de rede e que o LACOS já dispõe de uma rede local).

Como os elementos mínimos, serão disponibilizados via rede (*Internet*), então não só os alunos do grupo I teriam acesso, mas qualquer usuário do LACOS, ou mesmo não usuário, incluindo alunos de outras instituições de ensino. Neste sentido, entende-se que a experimentação remota propicia de fato um uso mais racional dos meios, e que se não alcança, se aproxima muito mais da idéia de “fazer com que o conhecimento produzido pela humanidade chegue até o aluno” (Becker [40]).

Em relação à segurança, o quadro também é complexo, como na primeira abordagem. Do ponto de vista de preservação física do equipamento (servomecanismo) pode ser necessário a inclusão de sistemas de proteção físicos (fusíveis, relés etc.), ou lógicos (incorporando-se *software* para supervisão e tomadas de decisão em situações que podem comprometer o equipamento), o que não se constitui um problema. Mas, do ponto de vista de redes (*Internet*) a segurança é um ponto crítico e sem uma solução definitiva até o momento, contudo, este fato não tem impedido o crescimento de seu uso, mesmo em situações extremamente delicadas, como a de transações bancárias.

Quanto a questão ensino – aprendizagem, uma primeira estratégia poderia ser o uso da experimentação remota como uma ferramenta adicional num curso presencial; assim, professor e alunos poderiam utilizar a ferramenta ao longo do curso de forma equivalente a sugerida na primeira abordagem, com a vantagem de que o acesso à ferramenta é muito mais fácil (via *Internet*) e a qualquer momento (incluindo fins de semana). Por outro lado, desde que não se incorporem recursos adicionais, ou que os alunos se reúnam num mesmo local para usar a ferramenta (o que é perfeitamente viável na primeira abordagem), pode-se criar uma condição menos favorável para a aprendizagem no que diz respeito à colaboração, que seria cada aluno utilizar a ferramenta isoladamente.

3.5 – Conclusões

A experimentação remota, se não apresenta vantagens de custos também não possui desvantagens. Em termos de segurança e ensino – aprendizagem, se for utilizada como uma ferramenta adicional no ensino, ainda assim, deve incorporar recursos que garantam uma maior exploração da base de conceitos (não só equipamentos reais mas simuladores) e crie uma situação mais favorável ao aprendizado (colaboração), tendo-se a clareza de que em

situações específicas não deve ser usada (equipamentos para formação básica de alunos), e especificamente sobre segurança, esta passa para um campo técnico, e soluções técnicas devem ser propostas, independentemente de questões de consciência ou conscientização. E ainda, em relação ao acesso aos meios, de fato esta abordagem tem muito a contribuir na problemática discutida ao longo deste capítulo, pois não apenas permite um compartilhamento mais racional de equipamentos dentro de uma instituição de ensino, como também permite o compartilhamento entre instituições, e por isso foi escolhida para este trabalho. No próximo capítulo, procura-se estabelecer um modelo para experimentação remota levando em consideração os diversos aspectos aqui apresentados.

CAPÍTULO 4 – ESPECIFICAÇÃO DE UM MODELO PARA LABORATÓRIO REMOTO DE CONTROLE

Partindo dos elementos a serem explorados na base de conceitos e da arquitetura resultante dos elementos mínimos definidos no capítulo 3, especifica-se no item 4.1 deste capítulo, um modelo que permita lidar de forma consistente e, muitas vezes, completa com os elementos a serem explorados. Dificuldades, limitações e alternativas procuram ser evidenciadas. No item 4.2 é feito um resumo das especificações. No item 4.3 uma arquitetura para implementação do modelo é especificada, levando-se em consideração possíveis expansões. Por fim, no item 4.4, o modelo estabelecido é incluído na Tabela 1 do capítulo 2 para efeitos de comparação e a questão de implementação é abordada.

4.1 – Arquitetura

Os elementos mínimos definidos no capítulo 3, resultam na arquitetura ilustrada na Figura 12, disponível para a realização de experimentos de forma remota. Ainda no capítulo 3, também foram apresentados os vários pontos da base de conceitos a serem explorados com esta arquitetura. Estes pontos são listados novamente no sentido de facilitar a descrição da especificação do modelo.

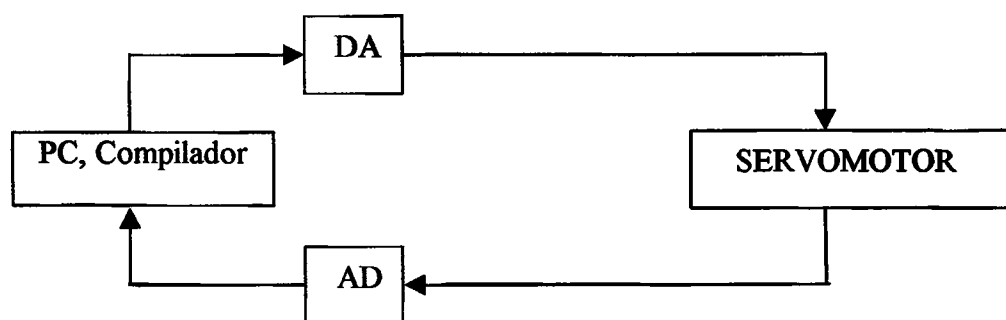


Figura 12 – Arquitetura para realização dos experimentos.

Os conteúdos a serem explorados (aprendizagem) com a arquitetura da Figura 12, estão relacionados nos itens de (a) a (k) a seguir e esta será a nomenclatura para referenciá-los doravante, ou seja entre parênteses:

- a - aprendizagem de linguagem de programação;
- b – aprendizagem (uso)/desenvolvimento de pequenos simuladores de sistemas;

- c – aprendizagem (uso)/desenvolvimento de programação da placa de aquisição;
- d – aprendizagem (uso)/desenvolvimento de programas para interação com o servomecanismo em malha aberta (aplicar um sinal elétrico no motor e registrar sua resposta);
- e - determinação de parâmetros do motor (identificação) usando resultados do item “d”;
- f – uso dos resultados do item (e) para projetar um controlador, por exemplo proporcional analógico depois um proporcional digital;
- g - evoluir o item (d) para programas de interação com o motor em malha fechada, ou seja, que incorporem o controlador projetado e testá-los;
- h - projetar outros tipos de controladores, por exemplo relé, relé com zona morta, PID, realimentação de estados e testá-los no item (g);
- i - avaliar o comportamento dos controladores diante de problemas específicos como regulação (manter a posição angular do eixo do motor em zero) e rastreamento (fazer com que a posição angular do eixo do motor acompanhe um perfil pré-estabelecido);
- j - projetar e avaliar o comportamento de controladores mais sofisticados usando por exemplo: controle *fuzzy*, adaptativo e etc.;
- k - aprendizagem/desenvolvimento de simuladores de sistemas mais complexos com os quais se deseje testar algum controlador específico.

4.1.2 – Formulação do modelo

Quanto à aprendizagem de uma linguagem de programação ou do desenvolvimento de *software*, itens (b), (c), (d), (g) e (j), sob o ponto de vista da experimentação remota, para que se pudesse utilizar todos os recursos do compilador (ambiente de desenvolvimento: editor, *help*, etc.), seu ambiente de desenvolvimento deveria ser colocado à disposição em rede (*Internet*). Este quadro tem dois fatores importantes: a questão de autoria, ou seja, haveria de ser em concordância com o fabricante do compilador e a questão da velocidade de interação com usuário. Neste trabalho, não será tratada a experimentação remota com o uso do compilador desta forma. A abordagem adotada será a de tornar disponível o uso do compilador (não de seu ambiente de desenvolvimento) em rede (*Internet*) para compilação de códigos gerados num editor de texto qualquer, no sentido de avaliar a sintaxe do código e gerar um executável, pois nela, não se viu dificuldades em relação à autoria (posto que a partir do momento em que se compra um compilador ele pode ser usado para compilar programas de qualquer pessoa, não apenas de quem o comprou) ou a velocidade (neste caso, não se estaria desenvolvendo um *software*, mas apenas avaliando um código) e ainda, esta

segunda estratégia não exclui totalmente o aprendizado de uma linguagem de programação em si. Se o aprendiz tiver acesso ao manual da linguagem, pelo menos rudimentos da linguagem podem ser trabalhados.

Em relação ao teste de controladores, item (i), desde que são inúmeras as estratégias existentes na teoria de controle para concepção de um controlador, seria uma tarefa exaustiva tentar disponibilizar ao usuário todas as estratégias (estruturas previamente desenvolvidas para as quais o usuário só especificaria parâmetros), além do que, limitaria a capacidade de criatividade do usuário no sentido de pesquisar novas estratégias. A especificação portanto, será garantir ao usuário a possibilidade de implementação de um controlador qualquer. Este controlador deve ser definido na forma de uma *function*, guardada as devidas restrições no que se refere às variáveis de entrada para esta *function*, e ao usuário deverá ser disponibilizado um protótipo da referida *function* a partir do qual ele possa desenvolver (escrever) a sua própria. Esta alternativa é suportada pela disponibilização do uso do compilador como descrito no parágrafo anterior. Como nesta situação, a possibilidade de erro existe, seja um erro de digitação, ou esquecimento de definição de alguma variável, no código fonte gerado pelo usuário, e o compilador gera mensagens de erros, ou seja faz uma realimentação com o usuário, uma especificação adicional é a disponibilização de mensagens de erro geradas pelo compilador ao usuário. Vale dizer que neste caso usa-se transferência de uma *function* (código de programa) além de admitir-se trabalhar com o erro.

Quanto ao “uso”, mencionado nos itens (b), (c), (d) ou (g) a especificação será a seguinte: desenvolvimento e instalação de programas executáveis no PC, específicos para cada experimento, e a disponibilização dos seus resultados de forma remota, no formato de um arquivo de dados. Por exemplo, um executável que simule a dinâmica do servomecanismo em malha aberta, outro para malha fechada com um controle PID, um que interaja com o servomotor real em malha aberta, outro para malha fechada com um controle PID. Neste momento, então, além de experimentos com o equipamento real, garante-se o uso de simuladores e ainda, a simulação de sistemas mais complexos (item (j)), bastando desenvolver e instalar seus executáveis no PC. Além disso, facilita-se a inclusão de novos experimentos, assim como o aproveitamento de executáveis já desenvolvidos, devendo-se apenas rever a formatação de seu arquivo de dados. Esta especificação não exclui a possibilidade de se ter mais de um PC (conectado ou não a um equipamento real), neste caso, cada PC deverá ter instalado os executáveis específicos de um experimento mais um *software* que permita a

comunicação com o servidor de experimentação remota, que deverá gerenciar a requisição de um PC ou outro. Por fim, observa-se que neste caso a transferência de parâmetros está sendo utilizada.

Para o caso particular de um controlador qualquer, item (i), a estratégia que se especifica é a instalação de um programa fonte no PC que será compilado e *linkado* com o código submetido pelo usuário na forma de uma *function*. Este programa fonte deve conter todos os elementos necessários para a realização do experimento a menos da rotina de controle. Assim, para cada equipamento real (ou simulado) deve-se ter um programa fonte específico.

Quanto às abordagens analógica ou digital mencionadas no item (f), a seguinte estratégia é especificada: a disponibilização do período de amostragem ao usuário na forma de uma variável, permitindo-lhe escolher seu valor dentro das limitações da placa de aquisição, para o servomotor real, ou, de acordo com o passo de integração para simulação nas inicializações do experimento. Neste caso, como a placa selecionada (item 3.4.1) permite que se trabalhe com um período de amostragem mínimo de 5 milissegundos, tem-se que usando este valor mínimo se assegura um controlador contínuo e para valores maiores (múltiplos inteiros) controladores discretos. Um aspecto importante diz respeito ao tempo necessário para realizar os cálculos do controlador, na situação de um controlador definido pelo usuário. Neste caso, se estes cálculos demandarem mais tempo do que o período de amostragem, os resultados gerados não são corroborados pela teoria de controle de sistemas contínuos ou discretos. A especificação é a interrupção da solicitação do usuário e o envio de uma mensagem sobre o ocorrido.

Para a realização dos itens (e) e (f), ou seja a realização de experimentos em si, a especificação se refere a interface com o usuário, e neste caso, a rigor, um primeiro aspecto importante diz respeito a ergonomia [45] desta interface. Inicialmente, a única especificação é que esta interface seja visual, baseada em “botões” e que traga informações sobre o significado destes “botões”. Outro aspecto importante diz respeito aos recursos adicionais a serem disponibilizados ao usuário para a realização de um experimento. O seguinte conjunto mínimo é especificado: textos (html) relacionados a teoria de controle e roteiros de experiências, acesso aos arquivos de dados contendo resultados de um experimento, um modo gráfico para visualização de resultados, imagens (vídeo ou pelo menos animações) do experimento, possibilidade de gravar/imprimir os resultados no computador do usuário, textos

específicos sobre cada etapa da realização de um experimento, um mecanismo que permita, mesmo de forma rudimentar, se explorar a questão de trabalho cooperativo e por fim, que este conjunto mínimo funcione de forma integrada, ou seja, sem a necessidade de se sair do ambiente.

Para permitir o tratamento dos problemas específicos do item (h) a seguinte abordagem é especificada para o servomotor real: incorporar procedimentos de controle aos programas executáveis responsáveis pela interação com o motor. Estes procedimentos, naturalmente, devem ocorrer antes do procedimento que realiza o experimento em si, solicitado pelo usuário. Por exemplo, dado que ao iniciar uma experiência a posição angular do eixo de saída do motor pode ser qualquer, então primeiramente o referido eixo deve ser posicionado na referência zero. Em seguida, visto que um dos parâmetros de entrada a ser especificado pelo usuário é a posição angular inicial desejada para o referido eixo, então o eixo deve ser posicionado no valor especificado. Feito isso, o experimento em si é executado. Esta abordagem garante a autonomia do equipamento real além de dar maior flexibilidade de experimentos possíveis de serem realizados.

Para que se complete as especificações, restam ainda os seguintes itens discutidos no capítulo 3:

- Custos;
- Segurança;
- Ensino – Aprendizagem.

De fato os custos já foram analisados no próprio capítulo 3 e o que pode-se acrescentar é a especificação do servidor e do PC (a placa já foi especificada no referido capítulo), porém, acredita-se ser mais apropriado acrescentá-las no próximo capítulo que trata da implementação.

Quanto ao ensino – aprendizagem já foram especificados neste capítulo, os recursos adicionais a serem disponibilizados ao usuário a fim de propiciar uma melhor interação com o ambiente de experimentação remota. Apenas se reforça a idéia de que num primeiro momento, esse ambiente de experimentação remota pode ser usado como uma ferramenta adicional num curso regular da área de controle.

Sobre a segurança, antes de estabelecer suas especificações, observa-se que diferentemente do capítulo anterior, este conceito agora pode incorporar vários aspectos distintos de segurança, seja em relação a redes de computadores, aos equipamentos a serem utilizados ou a seres vivos que eventualmente possam estar envolvidos no experimento. Deste modo, o uso do termo segurança deve passar a ser entendido como: **um bloco de preservação das condições de operação do objeto do experimento**. Para o caso particular do laboratório remoto de sistemas de controle, as especificações são as seguintes: *firewall* e antivírus no que se refere a segurança de rede (*Internet*). No que se refere ao uso do equipamento real (servomotor), seja nos itens (e), (g), (h) e (j), incorporar procedimentos de supervisão e controle (*software*) que avaliem:

- as reais possibilidades de uso de equipamentos reais. Eventualmente, por exemplo, pode ocorrer algum defeito na fonte de alimentação do motor, então, ele não poderá ser utilizado, neste caso deve-se interromper o experimento gerando uma mensagem ao usuário do ocorrido e disparar algum tipo de alarme próximo ao motor;
- o nível de tensão que será aplicada ao servomotor (malha aberta ou fechada) e caso este ultrapasse o valor máximo admitido (± 10 V), interromper o experimento gerando uma mensagem ao usuário do ocorrido;
- a frequência do sinal que está sendo aplicada ao servomotor (malha aberta ou fechada) e caso esta seja maior que 1 Hz, interromper o experimento gerando uma mensagem ao usuário com um aviso do ocorrido.

As três especificações anteriores visam a preservação física do equipamento, principalmente a terceira, pois caso o servomotor opere em frequências maiores que a especificada há um estresse muito grande do ponto de vista mecânico. Nota-se que a questão de estabilidade não foi incorporada nesta especificação visto que o servomotor é um equipamento didático (e esse tema pode ser exatamente o objetivo de um experimento), por outro lado, para um equipamento industrial esta especificação é imprescindível.

Por fim tem-se ainda no item (i) a possibilidade de teste de um controlador qualquer definido pelo usuário (*function*), neste caso, as especificações a seguir, valem tanto para o sistema real como para o simulado:

- Um pré compilador para avaliar as instruções da *function* antes que esta seja compilada a fim de gerar um programa executável. Caso seja detectado o uso de alguma instrução que possa ser nociva (afete área de memória, afete arquivos, etc.), ou ainda, que a *function* entre em *loop* infinito, interromper o experimento gerando uma mensagem ao usuário avisando a ocorrência.

4.2 – Resumo das Especificações do Modelo

Neste tópico é apresentado um resumo das especificações anteriores.

A – Interface multimídia

- Garantir ao usuário recursos para visualização de gráficos;
- Acesso a resultados numéricos dos experimentos;
- Tornar disponível os roteiros dos experimentos e textos sobre: teoria de controle, descrição de equipamentos e experiências;
- Visualização do experimento por vídeo ou animação gráfica baseada nas variáveis geradas pelo sistema real ou simulado;
- Gravar ou imprimir resultados no computador do usuário;
- Visualização de mensagens: compilador e da própria plataforma;
- *Help* sobre o uso do próprio aplicativo;
- Comunicação entre usuários do laboratório (colaboração, *chat*).

B – Flexibilidade

- Permitir realização de experimentos em malha aberta e fechada;
- Permitir teste de algoritmos de controle pré definidos (envio de parâmetros), ou teste com um controlador qualquer (envio de *function*);
- Permitir experimentos com controladores contínuos ou discretos;
- Permitir a inclusão de novos experimentos (reais ou simulados).
- Permitir experimentos com equipamentos reais ou simulados.

C – Integração

- Dos itens relativos à flexibilidade (B);
- Dos itens relativos à interface multimídia (A).

D - Autonomia e segurança

- Funcionamento durante 24 horas;
- Permitir a realização de experimentos específicos (regulação ou rastreamento) de forma autônoma;
- Testes iniciais automatizados para verificação das reais possibilidades de uso de equipamentos reais;
- Intervenção automática, se necessário, para garantir integridade de equipamentos e da própria plataforma;
- No caso específico da proposição de algoritmos de controle, avaliar se estes não têm instruções nocivas ou que de alguma forma impeçam o funcionamento normal da plataforma;
- Mecanismos para segurança de redes (*firewall*, antivírus).

4.3 – Especificação da Arquitetura

A arquitetura que se especifica para implementação do modelo definido no item 4.2, é baseada na referência [14], ou seja, usando o modelo cliente servidor. Prevendo a possibilidade de incorporação de novos elementos mínimos (PC, equipamento real e placa) ou simplesmente novos PC's para experimentos simulados, acrescentou-se uma rede local para dar suporte a esta possível expansão, como mostrado na Figura 13.

Descreve-se a seguir, o que deve estar presente nos elementos da arquitetura.

PC1:

- 1 - Compilador (linguagem C);
- 2 - Placa de aquisição conectada ao equipamento real (motor);
- 3 - Programa executável para o problema de identificação do motor;
- 4 - Programa executável para o problema de teste de controlador no motor, quando o controlador é previamente definido;
- 5 - Programa fonte para o problema de teste de controlador no motor quando, o controlador é definido pelo usuário;
- 6 - Arquivos de lote para compilação e *linkagem* do programa fonte com a rotina de controle proposta pelo usuário;

7 - Monitor para gerenciamento da comunicação com o servidor e chamadas aos programas adequados para a realização do experimento solicitado.

Observação:

Os programas mencionados em 3, 4 e 5 devem incorporar os aspectos de segurança e autonomia (exceto funcionamento 24 horas),

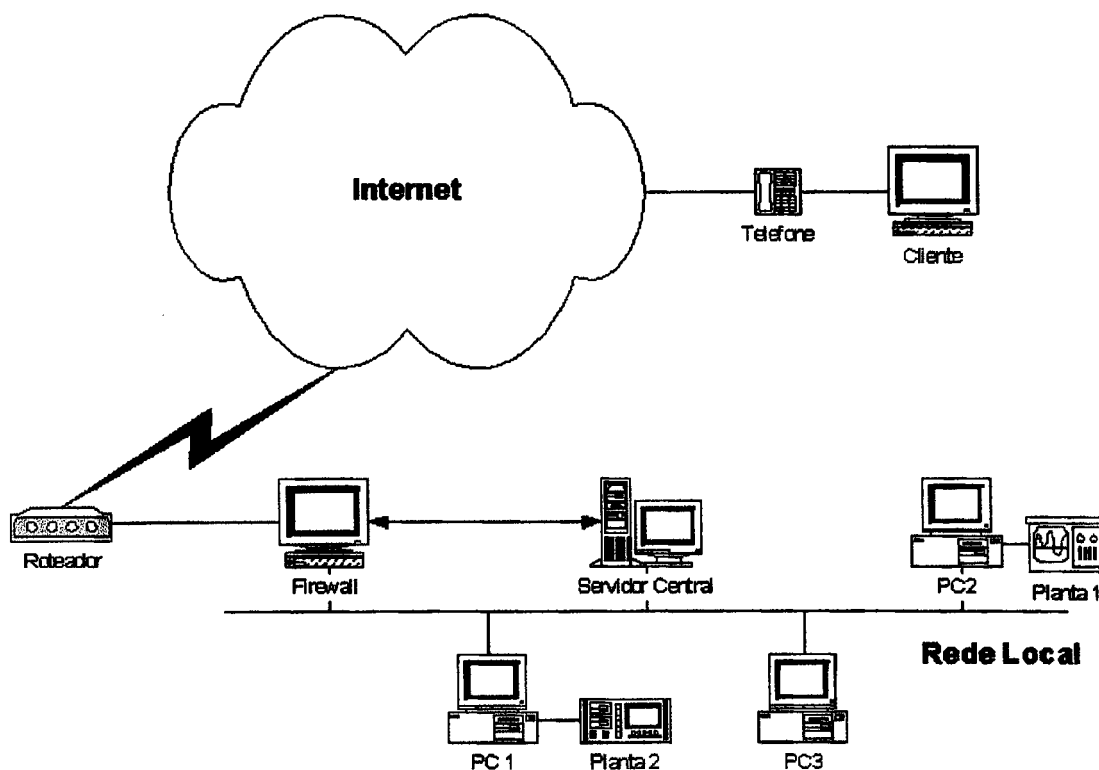


Figura 13 – Arquitetura geral.

PC2:

Os mesmos itens de 1 a 7 do PC1, substituindo-se o equipamento real (motor) por um outro, por exemplo o *fun plate*.

PC3:

- 1 - Compilador (linguagem C);
- 2 - O mesmo do item 3 do PC1 só que usando simulação do motor;

- 3 - O mesmo do item 4 do PC1 só que usando simulação do motor;
- 4 - O mesmo do item 5 do PC1 só que usando simulação do motor;
- 5 - O mesmo do item 6 do PC1;
- 6 - O mesmo do item 7 do PC1.

Observação:

O programa do item 4 deve incorporar o aspecto de segurança.

Servidor:

- 1 - O aplicativo para o gerenciamento de informações com os clientes e de informações com os PC's;
- 2 - Um servidor *Web* para fornecimento de páginas html com os roteiros das experiências e características de equipamentos;
- 3 - Um servidor de *Chat* para troca de mensagens entre usuários;
- 4 - O programa "CLIENTE" para *download*.

Cliente:

- 1 - Todos os elementos da interface multimídia (A);
- 2 - Meio para o estabelecimento de conexão com o servidor;
- 3 - Meio visual para a especificação do experimento que deseja realizar.

4.4 – Conclusões

O modelo estabelecido preenche a última coluna da Tabela 1 do capítulo 2, como mostrado na Tabela 4. De fato, o modelo apresenta contribuições referentes a uma maior exploração dos conteúdos relativos a sistemas de controle. Do ponto de vista de ensino, incorpora mídias consideradas relevantes neste processo. Desta forma, pode-se dizer que, teoricamente, o modelo é aceitável. Por outro lado, é necessário considerar ainda sua materialização, o que será objeto do próximo capítulo.

Tabela 4 – O modelo proposto e os laboratórios remotos analisados.

Labotatório	Texas	C.Mellon	Western	UnB	UFSC	Tennessee	Proposto
Transferência de Parâmetros		✓	✓	✓		✓	✓
Transferência de Códigos					✓		✓
Sistemas reais		✓	✓	✓	✓	✓	✓
Sistemas simulados	✓						✓
Textos (html)	✓	✓	✓	✓	✓	✓	✓
Resultados: gráficos						✓	✓
Resultados: numéricos						✓	✓
Imagens de Vídeo		✓	✓	✓	✓		✓
Animações			✓				✓
Mensagens de erro							✓
Controle contínuo e discreto							✓
<i>Chat</i>							✓
Rudimentos de programação					✓		✓
Experimentos específicos							✓

CAPÍTULO 5 – MATERIALIZAÇÃO DO MODELO

Como parte deste trabalho, as seguintes atividades foram realizadas visando a implementação do modelo estabelecido no capítulo 5: co-orientação de uma dissertação de mestrado [46], implementação de programas aplicativos e elaboração/edição de textos relacionados a teoria de controle e aos experimentos a serem realizados.

A dissertação de mestrado foi submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Pará. e seu objetivo central consistiu no desenvolvimento de quatro módulos: Cliente, Servidor, Gerenciador de experiências e Administrador. A estratégia adotada foi o uso de modelos matemáticos (simulação) como aplicação final para os módulos cliente e servidor, sem deixar de considerar que posteriormente seriam incorporados equipamentos reais.

A implementação de aplicativos foi realizada no Laboratório de Controle e Sistemas do Departamento de Engenharia Elétrica e de Computação da UFPA, tendo como objetivo central o desenvolvimento dos seguintes módulos: aplicativos para simulação do servomecanismo, aplicativo para aquisição de dados com a placa PCL-812PG, aplicativos para realização de experimentos com servomecanismo real (incluindo inicializações e segurança).

Neste capítulo descreve-se as principais características da implementação, seus módulos, os elementos que compõem sua estrutura física atual, o *software* que está instalado em cada elemento, o uso do programa cliente com alguns resultados e ao final, algumas conclusões são apresentadas e os testes já realizados são citados.

5.1 – Implementação

O modelo está implementado na forma de um sistema distribuído onde um servidor central fornece vários serviços aos clientes conectados a ele e conforme as requisições, faz solicitações aos PC's, onde um módulo servidor - o gerenciador de experiência – executa um aplicativo que realiza um experimento. A Figura 14 ilustra os módulos desenvolvidos.

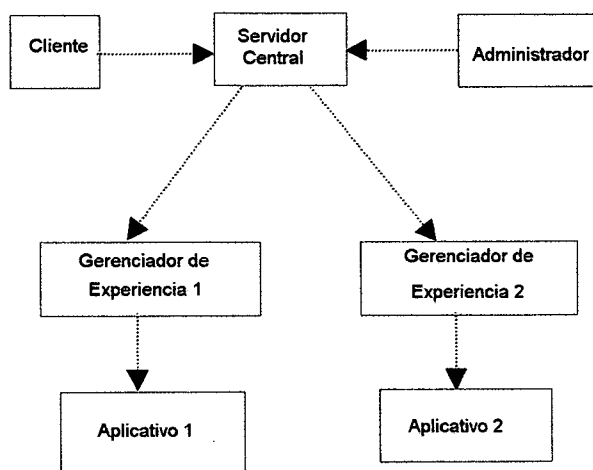


Figura14 – Módulos componentes do sistema implementado.

Para a implementação dos módulos cliente e servidor, a solução adotada foi o desenvolvimento de um protocolo próprio objetivando:

- maior controle e flexibilidade na interação com o usuário;
- processamento das requisições e formatação das respostas;
- modelar a interação entre cliente e servidor de maneira mais precisa;

5.2 – Módulos

Cada um dos módulos mostrados na Figura 14 é descrito a seguir:

5.2.1 – Cliente

Fornece de maneira integrada, acesso aos serviços disponíveis no laboratório remoto, permitindo ao usuário, submeter as experiências ao laboratório remoto, visualizar os resultados (numéricos, gráficos, animação), acesso a material didático sobre o experimento e teoria de controle, e interagir com outros usuários do laboratório remoto. Para obter o aplicativo cliente o usuário deve se conectar ao servidor através de um *browser*, fazer o *download* do mesmo e instalá-lo em seu computador. Para que possa acessar o laboratório remoto, deve estar cadastrado (*login* e senha) o que pode ser feito por *e-mail* ou pessoalmente com os responsáveis para gerência do laboratório remoto. O uso deste aplicativo e alguns resultados obtidos na realização de alguns experimentos, serão ilustrados no item 5.4.

5.2.2 – Servidor central

Atende aos aplicativos clientes fornecendo dados sobre as experiências, material didático e serviços que permitam aos usuários interagirem entre si. Por outro lado ele providencia a coordenação ou agendamento da execução dos experimentos, fazendo solicitações a um ou mais servidores auxiliares (gerenciador de experiência) encarregados de efetuar o experimento.

5.2.2.1 – Serviços e banco de dados implementados no servidor central

A Figura 15 ilustra os serviços e banco de dados implementados no servidor central.

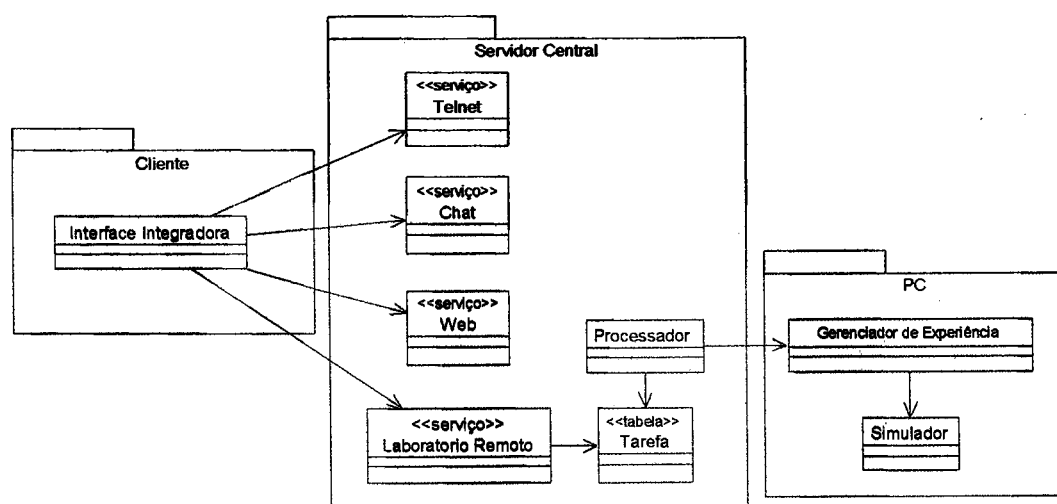


Figura15 – Detalhes do servidor central.

Cada serviço e o banco de dados são descritos a seguir:

- Servidor *WEB*

O servidor *web* implementado fornece páginas *html* para o aplicativo cliente e com isso permite a navegação dentro das informações sobre os experimentos e da teoria de controle. Os anexos de A a I ilustram as páginas *html*.

- Servidor de *chat*

Repassa as mensagens dos usuários para os outros criando pontos de

convergência (salas de bate papo) onde eles podem trocar mensagens entre si compartilhando um ambiente comum.

- Servidor do laboratório remoto

O servidor implementa um protocolo com as seguintes funcionalidades:

- Informa ao cliente a relação de plantas e experiências disponíveis bem como as informações necessárias para se compor uma experiência;
- Fornece o estado das experiências realizadas pelo usuário, inclusive o resultado das mesmas;
- Permite ao cliente compor experiências de identificação e controle em malha fechada, sendo que nesta última, permite que o usuário defina uma rotina de controle na linguagem de programação “C”;
- Relaciona as experiências com as respectivas páginas *web* que as descrevem;

- Banco de dados

O servidor central possui armazenado num banco de dados relacional os seguintes modelos:

- Das experiências que se pode realizar ;
- Dos dados de inicialização de uma experiência;
- Dos usuários;
- Das tarefas.

Estes modelos permitem que o servidor forneça ao usuário os dados sobre quais experiências podem ser realizadas e as variáveis que podem ser inicializadas pelo usuário. O aplicativo cliente disponibiliza estes dados numa seqüência de telas para composição do experimento do usuário e envio da tarefa ao servidor. O servidor a adiciona a uma lista de tarefas com o *status* “agendada”. De acordo das as disponibilidades do sistema, as tarefas agendadas são submetidas aos gerenciadores de experiências, alterando o *status* da tarefa para “em realização”. Ao receber o resultado do gerenciador de experiência o servidor pode passar o *status* da tarefa para “erro” se algum erro ocorreu no processamento ou para “realizada” se não houver erro. A cada alteração de *status* o cliente é notificado sobre a situação do processamento.

O servidor mantém o estado de todas as tarefas agendadas e já executadas dos clientes. O aplicativo cliente mantém uma réplica dos registros que pertencem ao cliente. A submissão de tarefas consiste na adição de um novo registro, a execução consiste na mudança de estados

dos registros. As alterações de estado são efetuadas no servidor que notifica os clientes quando isto acontece. Assim, a dinâmica de interação entre cliente e servidor consiste na adição, deleção e alteração dos registros no banco de dados.

5.2.3 – Servidor auxiliar – Gerenciador de experiência

Gerencia as experiências a serem executadas em um PC. O gerenciador é um servidor que implementa os serviços de comunicação TCP/IP integrando programas aplicativos ao laboratório remoto, ele encapsula todo o protocolo de comunicação e mecanismo de controle, liberando o programa aplicativo. Recebe solicitações do servidor central e executa a experiência solicitada pelo mesmo. Para isso ele executa um programa aplicativo responsável pelo experimento (simulado ou real) e para cada programa aplicativo deve haver um gerenciador de experiência.

Para experimentos de simulação, um PC pode realizar várias experiências em várias plantas, bastando que exista o programa aplicativo responsável pela simulação. Para experimentos com equipamentos reais, um PC, dependendo do *hardware* (placa de aquisição), poderá executar várias experiências em um ou mais equipamentos.

5.2.4 – Programa Aplicativo

No caso de experiência usando simulação, é um programa executável ou fonte, que realiza a simulação da experiência através do modelo matemático da mesma. Nos casos de experimentos em malha aberta ou em malha fechada com controlador pré-definido utiliza-se um programa executável. No caso de um controlador definido pelo usuário usa-se um programa fonte. Para experiência usando um equipamento real, valem as mesmas observações feitas para o caso de simulação em termos de executável ou fonte.

O programa aplicativo é ativado pelo gerenciador de experiência e recebe um arquivo de entrada contendo os dados necessários para a realização da experiência (fornecidos pelo usuário) e como resultado gera um arquivo com os dados do resultado da experiência. A interface com o gerenciador é feita através destes arquivos de entrada e saída que seguem um padrão de formatação. Assim, os programas aplicativos são relativamente independentes do resto do sistema, não necessitando implementar nenhum mecanismo de comunicação interprocessos a não ser ler e escrever em arquivos. Isso facilita a adição de novos experimentos (reais ou simulados) ao laboratório remoto, e o aproveitamento de aplicativos já

existentes, alterando apenas a entrada e saída de dados de forma a adequar a interface do gerenciador de experiência.

5.2.4.1 – Programas aplicativos para experimento com servomotor real

Tratando-se do motor real, foi incorporado no código destes programas aplicativos, antes do *loop* que realiza o experimento solicitado, alguns procedimentos de inicialização que são descritos a seguir:

- O primeiro refere-se apenas ao experimento em malha aberta e consiste em verificar se o nível de tensão do sinal de entrada a ser aplicado no motor (especificado pelo usuário) é superior ao valor máximo estabelecido (+/- 10V), caso afirmativo retorna-se uma mensagem do ocorrido ao usuário e o experimento é abortado;
- O segundo refere-se apenas ao experimento em malha fechada com o controlador definido pelo usuário, neste caso é feita uma avaliação do tempo necessário para executar (numa única iteração) o algoritmo proposto e caso este tempo seja maior que o período de amostragem, a mesma atitude do primeiro procedimento é adotada;
- O terceiro procedimento vale para qualquer experimento e, para sua apresentação, na Figura 16, ilustra-se o sistema de referência para o deslocamento angular (em graus) do eixo do motor, bem como as indicações (em volts) do sensor de posição angular. Nota-se que o sensor, pode indicar 0 volts tanto a 0° como a 180°, contudo a referência 0° é única. Como ao iniciar um experimento não se sabe *a priori* qual será a posição angular do eixo, a primeira etapa do procedimento consiste na leitura da posição angular e se esta for zero aplica-se uma tensão no motor durante um intervalo de tempo. O nível da tensão e o intervalo de tempo foram ajustados de tal modo que o eixo do motor se desloque o suficiente para que o sensor de posição angular não indique mais zero. Caso contrário (leitura da posição angular diferente de zero), então executa-se um algoritmo de controle que posicione o eixo na referência zero. A Segunda etapa do procedimento consiste em verificar qual posição angular inicial para o eixo do motor foi especificada pelo usuário, se for nula nada é feito, senão executa-se um algoritmo de controle que posiciona o eixo do motor no valor especificado pelo usuário.

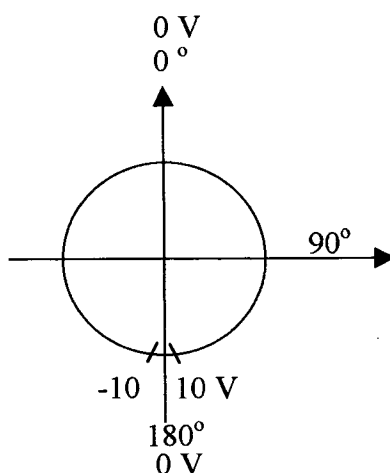


Figura 16 – Sistema de referência para o deslocamento angular do eixo do motor real.

- O quarto procedimento vale para a execução dos experimentos em malha fechada (controlador pré-definido ou definido pelo usuário) e consiste na avaliação do nível de tensão a ser aplicada na armadura do motor, caso seja superior ao valor máximo estabelecido ($\pm 10V$) o experimento é interrompido e uma mensagem é retornada ao usuário.

5.2.5 – Aplicativo Administrador

Permite a inclusão/exclusão de usuários ou experimentos, nele:

- Dados de um usuário são armazenados (nome, endereço, *e-mail* etc.) e principalmente é onde se cria um *login* e senha que permitirão ao usuário acessar o sistema;
- Dados associados a um experimento já existente estão armazenados e dados associados a um novo experimento podem ser incluídos. Estes dados podem ser: tempo total do experimento, valores iniciais de variáveis relativas ao experimento (variáveis de estado ou de saída), período de amostragem, sinais disponíveis (degrau, quadrada, etc.), ou outros de acordo com o experimento. A cada um destes dados é criado um nome e um valor *default* que são apresentados ao usuário durante a execução do experimento.

Os procedimentos para inclusão/exclusão de usuários e experimentos são ilustrados no ANEXO J.

5.3 – Instalação Física

O laboratório remoto encontra-se, no momento da redação deste texto, instalado nas dependências do laboratório do curso de engenharia elétrica da UFPA e conectado a sua rede local, esta por sua vez, pertencente ao conjunto de redes que compõem a rede da referida Universidade.

O aplicativo servidor central, seu banco de dados e arquivo de inicialização, estão instalados num computador com as seguintes características:

- PENTIUM III 750 Mhz;
- 128 MB RAM;
- Placa de rede 10/100 RTL 8139;
- HD 13.06 GB;
- *Windows 2000 Professional – Service Pack 1.*

além do aplicativo servidor e banco de dados, também estão instalados neste computador:

- compilador *Borland C++ Builder 5*;
- Antivírus AVG 6.0;
- aplicativo administrador

e os seguintes aplicativos:

- um para experimento em malha aberta (simulação);
- um para malha fechada (simulação);
- um para experimento com controlador definido pelo usuário (simulação);
- os respectivos aplicativos gerenciadores de experiências para os três experimentos;
- o aplicativo cliente.

Para experimentos com o servomotor real [23], o computador utilizado tem as seguintes características:

- AMD K6II- 450 Mhz;
- 32 MB RAM;
- Placa de rede 10/100 RTL 8029;
- HD 6.02 GB;
- *Windows 98.*

Nele encontram-se também instalados:

- a placa de aquisição PCL – 812PG Advantech [PCL 93];

- *driver* da placa;
- o compilador *Borland C++ Builder 4*;

e os seguintes aplicativos de experiências com o motor real:

- um para experimento em malha aberta;
- um para malha fechada;
- um para experimento com controlador definido pelo usuário;

com seus respectivos aplicativos gerenciadores de experiências e arquivos de inicialização.

5.4 – Uso do aplicativo cliente e resultados

Para se utilizar o aplicativo cliente os seguintes pré-requisitos devem ser satisfeitos:

- sistema Operacional: Windows 95/98/Me/NT/2000;
- espaço em disco: 2Mb;
- conexão com a Internet.

A primeira tarefa do aplicativo cliente é solicitar os dados do usuário para *login* no sistema, como mostrado na Figura 17.

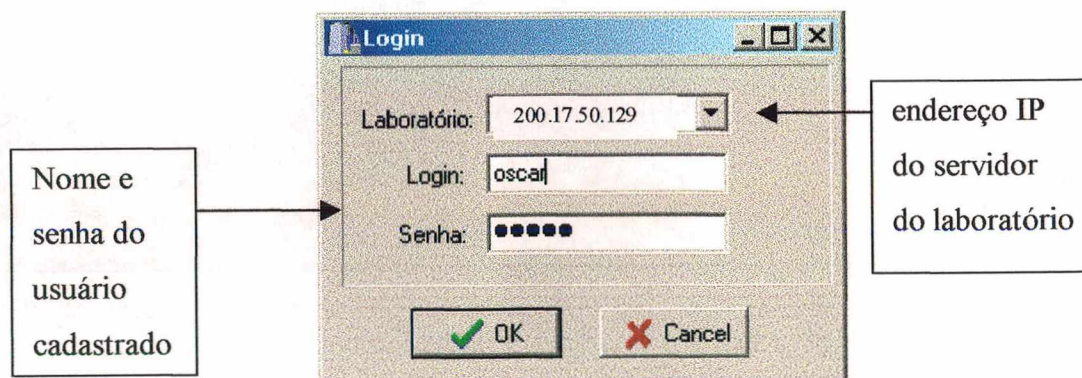


Figura17 – *Login* no sistema.

Se o *login* for bem sucedido o programa cliente irá para a tela principal mostrada na Figura 18, que pode ser dividida nos seguintes módulos:

- um menu que permite acessar as principais funções do programa cliente;
- uma barra de tarefas para seleção do modo de operação de maneira rápida;
- uma área de mensagens que permite acompanhar a execução, ou diagnosticar possíveis problemas, de um experimento;

Os modos de operação são três, com as seguintes características:

- modo Laboratório Remoto: permite realizar experiências e visualizar os resultados;
- modulo *Browser*: permite acessar os conteúdos didáticos do laboratório;
- modo *Chat*: permite interagir com outros usuários conectados ao laboratório.

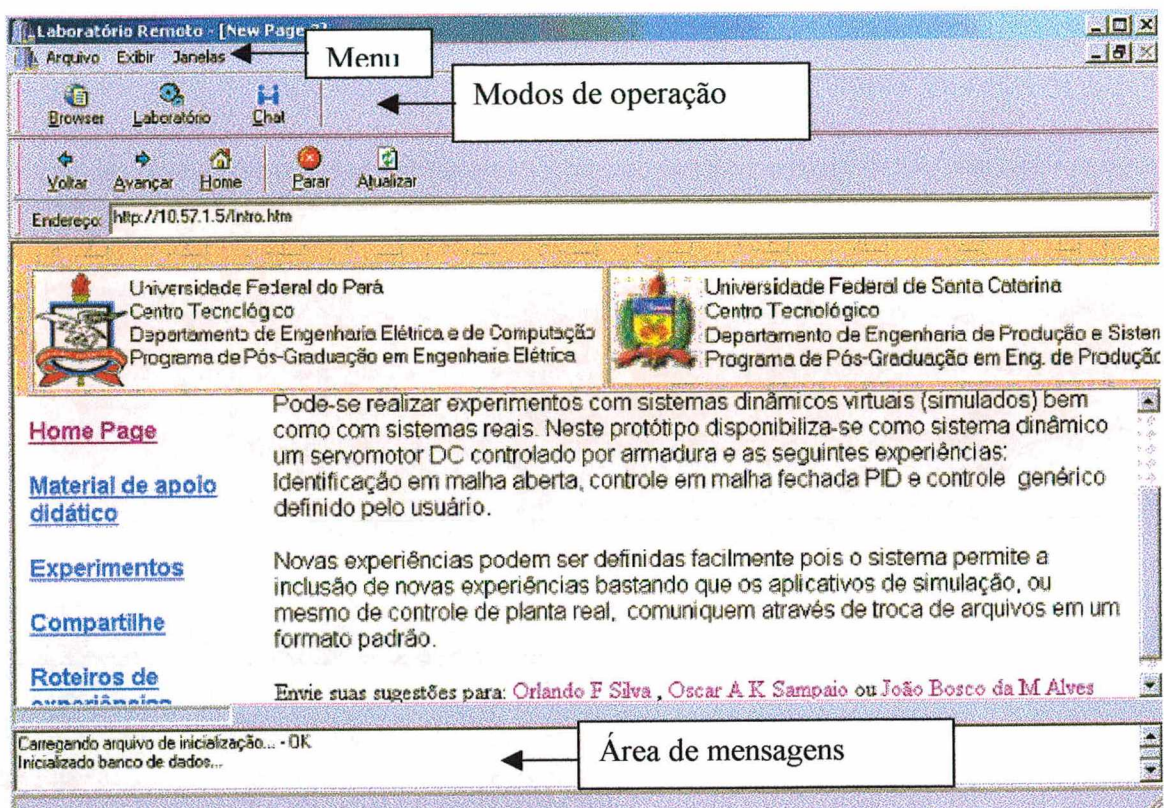


Figura18 – Tela principal do programa cliente.

Selecionado o modo de operação “Laboratório”, a janela da Figura 19 se torna ativa.

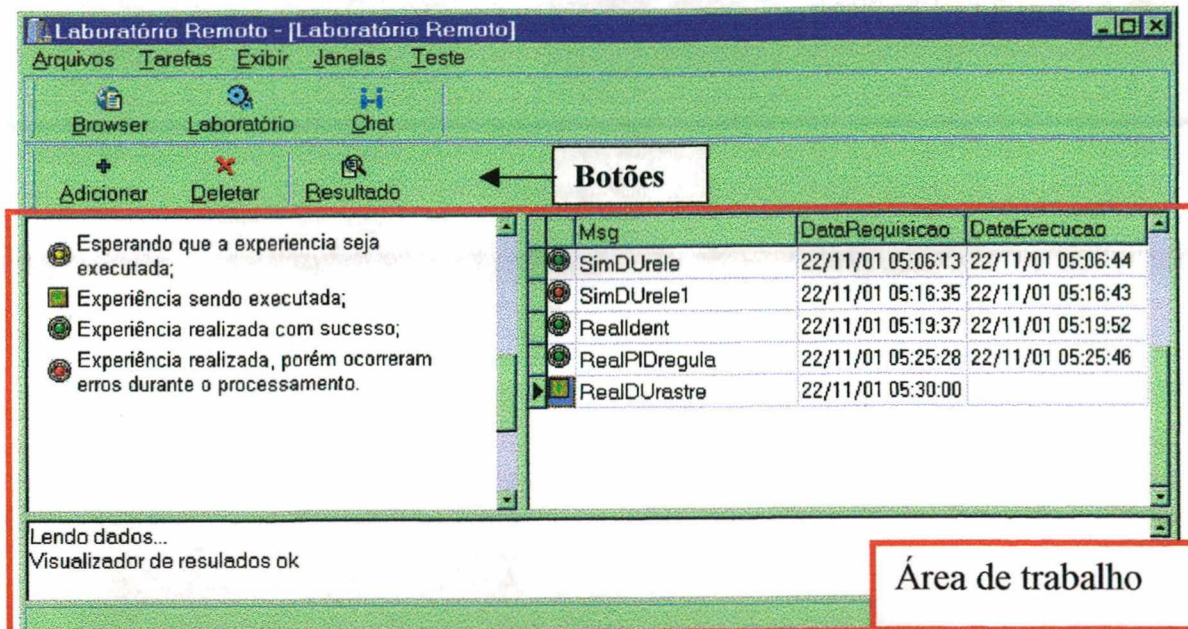


Figura19 – Janela do modo laboratório remoto.

No lado esquerdo da área de trabalho, existe uma legenda com o significado dos ícones adotados, e no direito são listadas as experiências já realizadas com a informação do seu estado.

O usuário pode acessar através de botões as seguintes funções:

- : adicionar nova experiência, esta opção abrirá uma janela com um assistente para a criação de uma experiência;
- : deletar uma experiência selecionada na lista a direita;
- : visualizar o resultado de uma experiência selecionada na listagem a direita.

A realização de uma nova experiência é feita clicando sobre o botão “Adicionar” na Figura 19. Neste caso, um assistente ficará ativo ao lado esquerdo da área de trabalho, como mostrado na Figura 20, e visa ajudar na definição da experiência a ser realizada, guiando o usuário, até completar os dados necessários para a sua realização.

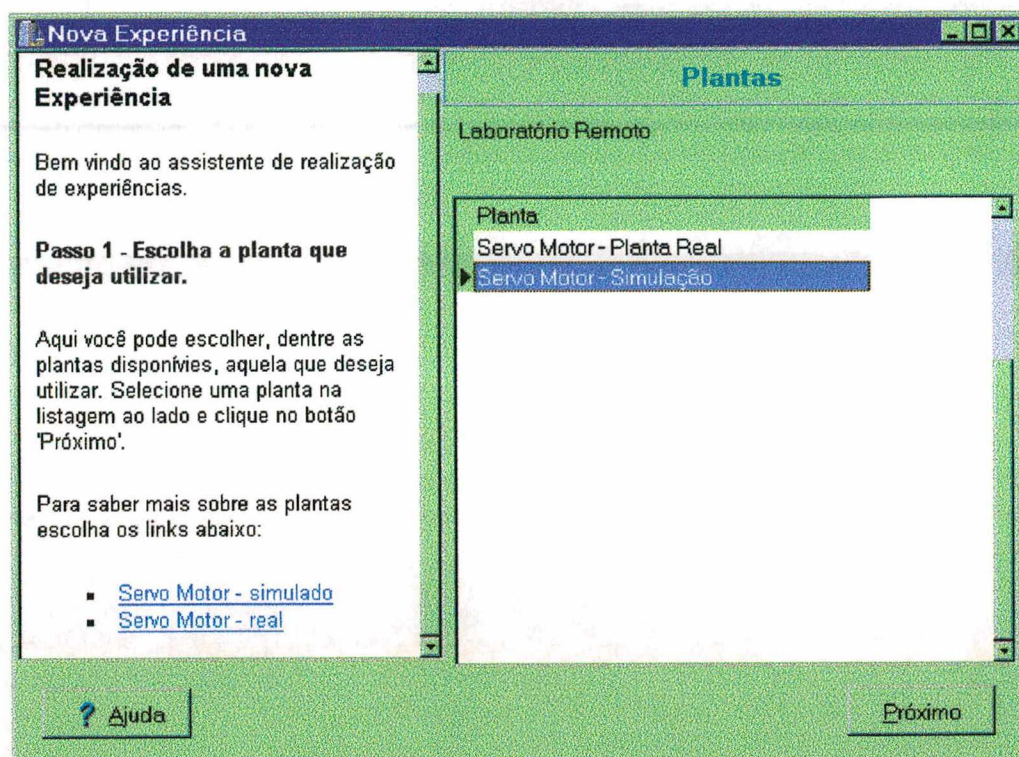


Figura 20 – Selecionando uma planta.

No lado esquerdo da área de trabalho na Figura 20, o usuário pode navegar em textos relativos às plantas disponíveis para realização de experiências, enquanto no lado direito é apresentada uma listagem das plantas disponíveis, dentre as quais deve escolher qual planta deseja utilizar para a realização de experimentos. Tendo selecionado uma, deve clicar no botão “Próximo” para passar para a próxima tela mostrada na Figura 21.

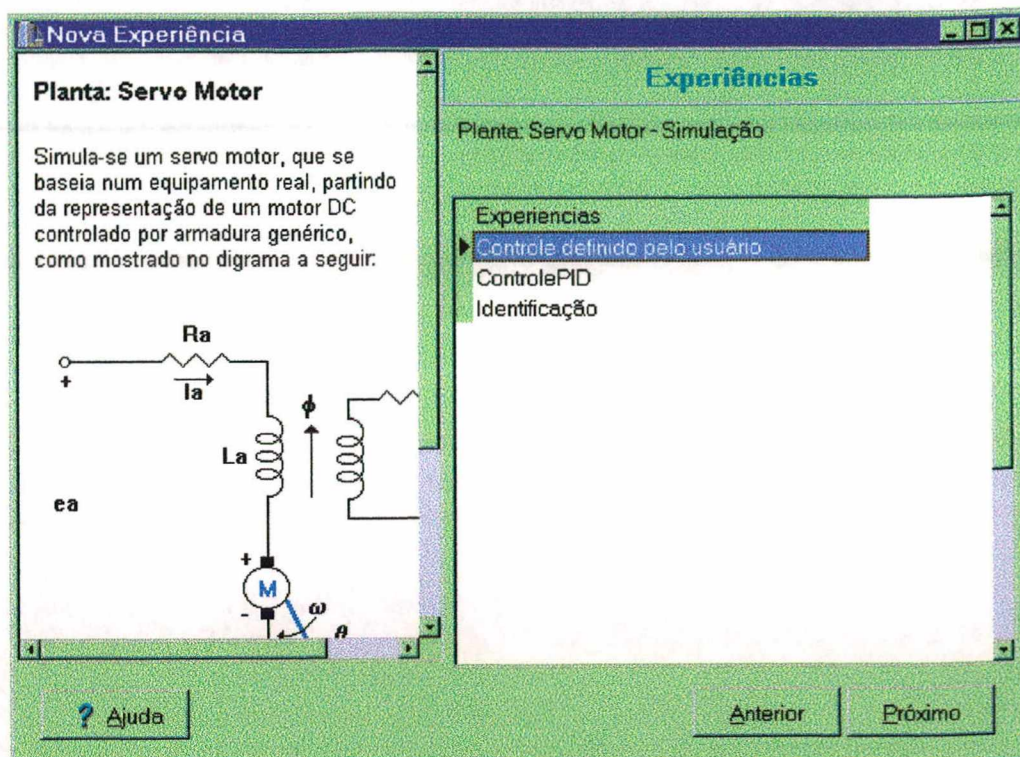


Figura 21 – Selecionando uma experiência.

Na Figura 21, ao lado esquerdo, o assistente mostra uma descrição da planta, e ao lado direito, as experiências disponíveis. Se houver dúvida, o usuário pode navegar nos textos apresentados pelo assistente. O botão “Ajuda” faz o retorno a página inicial. Senão, deve selecionar dentre as experiências disponíveis para aquela planta, uma que deseje realizar e clicar no botão “Próximo”, que levará à tela mostrada na Figura 22.

O próximo passo é fornecer o estado inicial e os parâmetros para a experiência. O assistente solicita que o usuário entre com estes dados no lado direito da janela mostrada na Figura 22. Do lado esquerdo é mostrada uma descrição da experiência, dos modelos utilizados e das variáveis. Uma vez estabelecidos os valores desejados, o botão “Próximo” deve ser ativado para se passar para outra etapa mostrada na Figura 23.

Nova Experiência

Planta: Servo-Motor

Experiência: Controlador Genérico Definido pelo Usuário

Neste caso o usuário pode criar seus próprios algoritmos de controle baseado em alguma técnica de seu interesse, conforme a figura 1. Para tanto, é disponibilizado o protótipo de uma função cujas variáveis de entrada são: o sinal de referência especificado pelo usuário (V_{in}), as provenientes dos sensores $XOV = [XOV[0] \ XOV[1]]^T = [\theta \ \omega]^T$, a variável associada à base de tempo (**tempo**) e o período de amostragem especificado pelo usuário já expresso em segundos (**Amostragem_seg**). A partir destas, o usuário deve gerar o cálculo da ação de controle. A linguagem de programação aceita para criar esta

Dados

Planta: Servo Motor - Simulação
Experiência: Controle definido pelo usuário

teta inicial:

omega inicial:

Sinal de Referência:

Amplitude:

perodo:

instante final:

perodo d amostragem:

Figura 22 – Estado inicial e parâmetros para um experimento.

Como a experiência do exemplo mostrado na Figura 22 é “controle definido pelo usuário”, o usuário deverá especificar qual a rotina de controle deseja utilizar, sendo ela definida na linguagem de programação “C”. O assistente apresenta um protótipo de uma rotina de controle (no caso, um controlador PID). A partir desse protótipo o usuário pode editar aquela que deseja trabalhar. Neste exemplo implementou-se um relé com zona morta, conforme mostrado na Figura 23.

Na última etapa na definição de uma experiência, o usuário deve fornecer um nome para sua futura identificação (Figura 24). Ao pressionar “Ok” a experiência será remetida ao servidor para processamento.

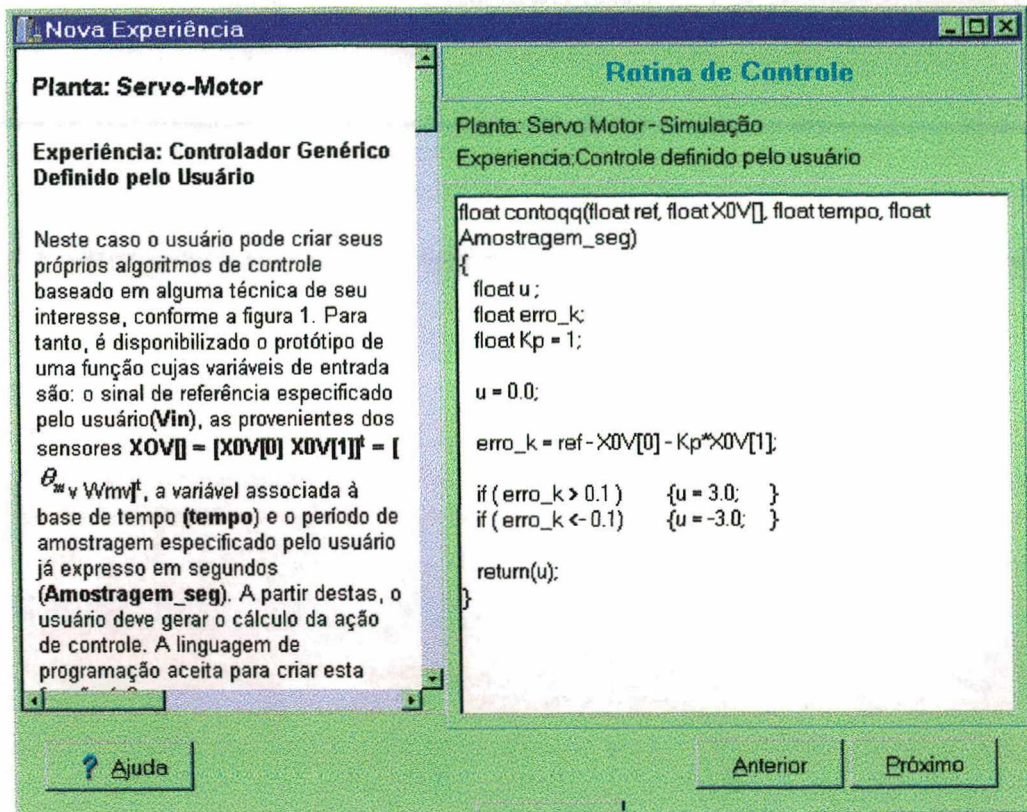


Figura 23 – Definindo uma *function*: relé com zona morta.

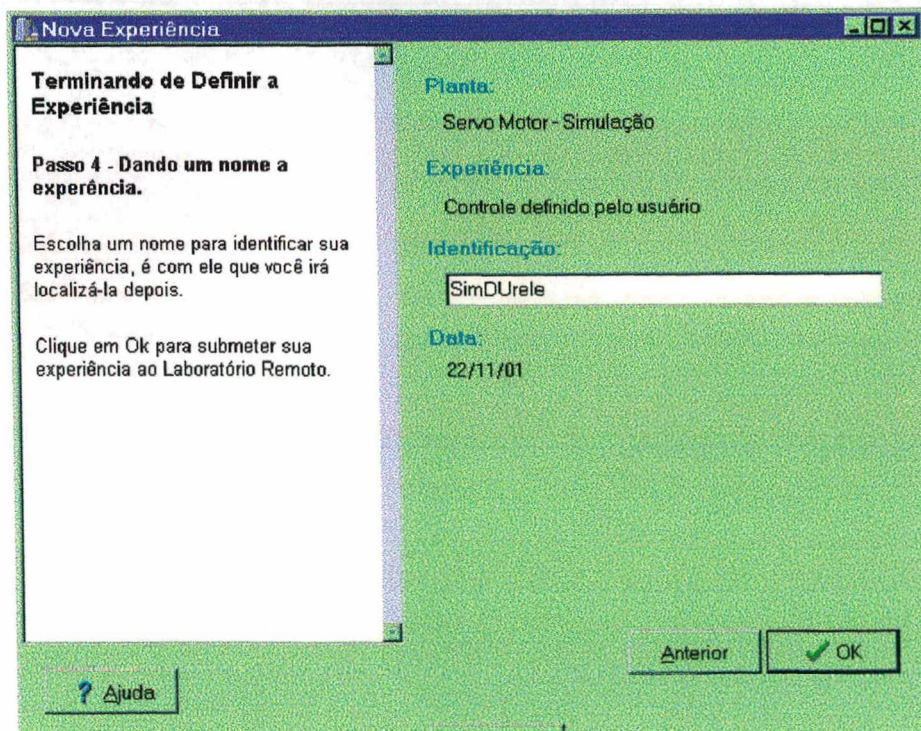
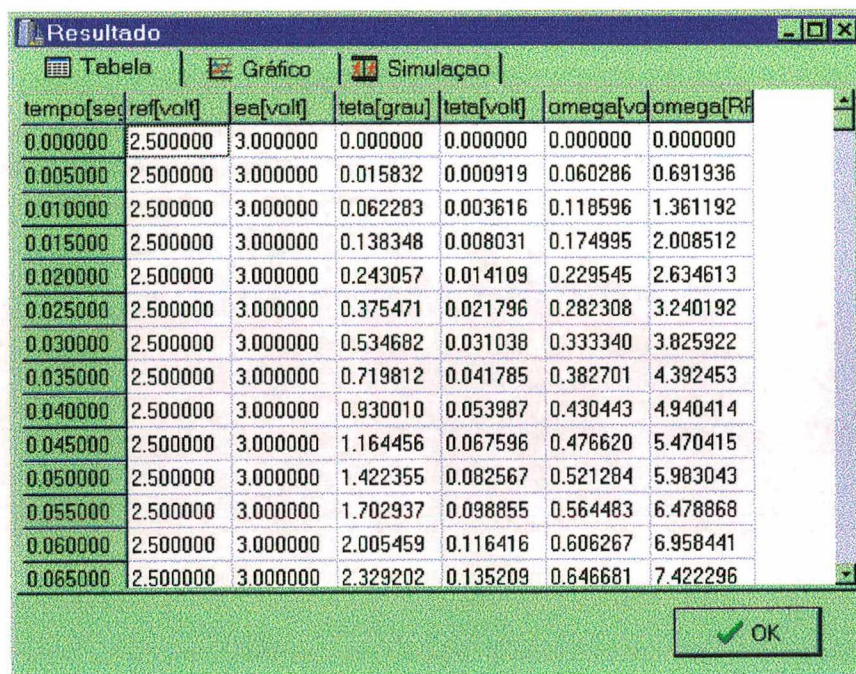


Figura 24 – Identificação da experiência.

O programa cliente irá retornar à tela do Modo Laboratório Remoto (Figura 19) onde a experiência irá aparecer com o *status* de “espera” (🕒). O estado irá mudar conforme o servidor for processando (🟢) e disponibilizando o resultado (🟢 - “Ok” ou 🚫 - Erro).

Uma vez processada com êxito a experiência, ao clicar em “Resultado” (📄 da Figura 19) a janela de resultado será apresentada, como mostrado na Figura 25.



tempo[seg]	ref[volt]	ea[volt]	teta[grau]	teta[volt]	omega[vo]	omega[Rf]
0.000000	2.500000	3.000000	0.000000	0.000000	0.000000	0.000000
0.005000	2.500000	3.000000	0.015832	0.000919	0.060286	0.691936
0.010000	2.500000	3.000000	0.062283	0.003616	0.118596	1.361192
0.015000	2.500000	3.000000	0.138348	0.008031	0.174995	2.008512
0.020000	2.500000	3.000000	0.243057	0.014109	0.229545	2.634613
0.025000	2.500000	3.000000	0.375471	0.021796	0.282308	3.240192
0.030000	2.500000	3.000000	0.534682	0.031038	0.333340	3.825922
0.035000	2.500000	3.000000	0.719812	0.041785	0.382701	4.392453
0.040000	2.500000	3.000000	0.930010	0.053987	0.430443	4.940414
0.045000	2.500000	3.000000	1.164456	0.067596	0.476620	5.470415
0.050000	2.500000	3.000000	1.422355	0.082567	0.521284	5.983043
0.055000	2.500000	3.000000	1.702937	0.098855	0.564483	6.478868
0.060000	2.500000	3.000000	2.005459	0.116416	0.606267	6.958441
0.065000	2.500000	3.000000	2.329202	0.135209	0.646681	7.422296

Figura 25 – Resultados de uma experiência em forma de tabela.

Selecionando a opção “Gráfico” (Figura 25) os resultados são apresentados na forma de gráficos como mostrado na Figura 26, e selecionando a opção “Simulação”, uma animação do experimento realizado é efetuada, como ilustra a Figura 27.

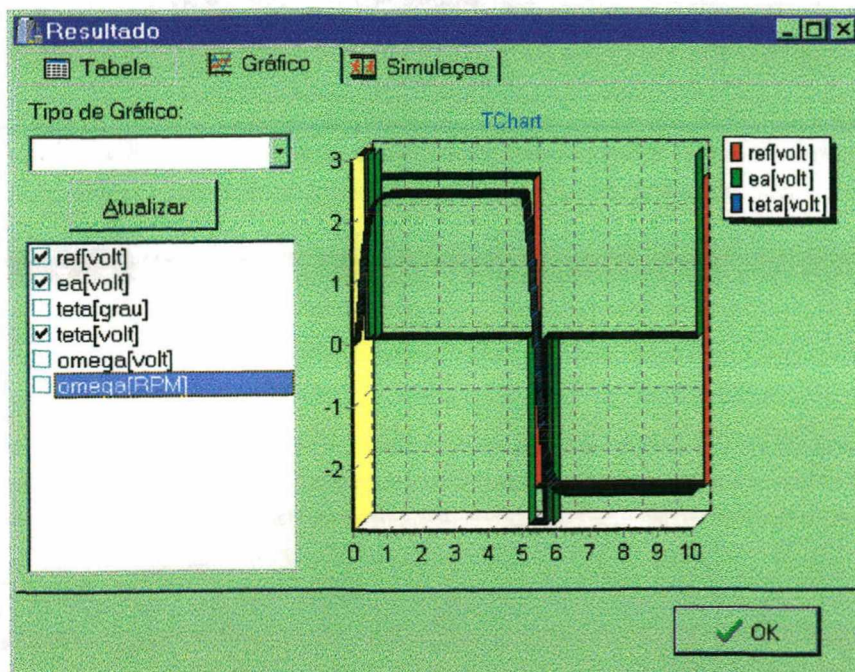


Figura 26 – Resultados de uma experiência na forma de gráficos.

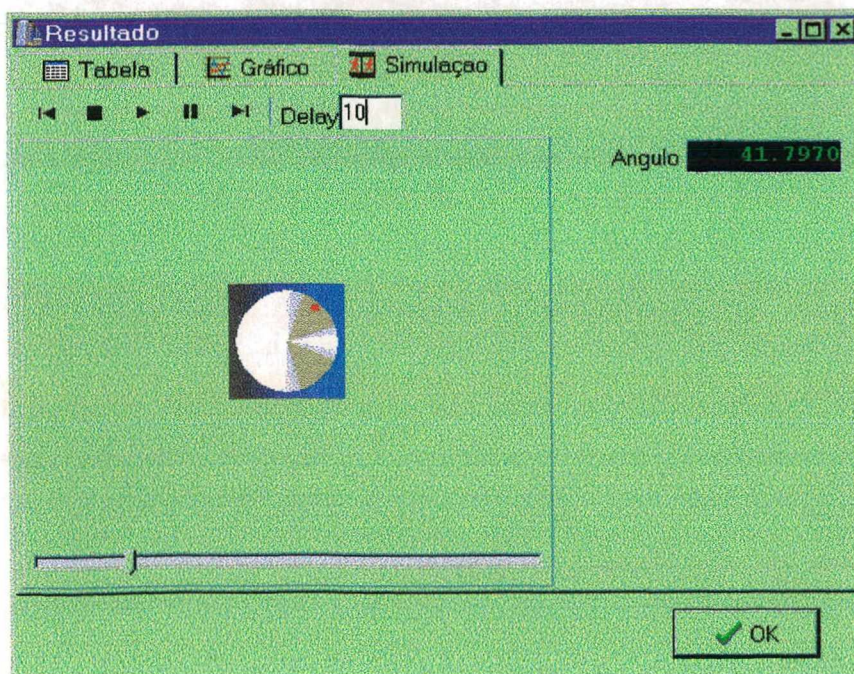


Figura 27 – Animação do experimento realizado.

Pode-se traçar vários tipos de gráficos: Linha, Linha 3D, pontos e barras. A animação pode ser executada pelo usuário utilizando os botões: ◀ ■ ▶ || ▶ (início, parar, rodar, pausa e fim, respectivamente), ou a barra de posição localizada na parte inferior da Figura 27. A velocidade de animação pode ser controlada mudando o parâmetro *delay* (quanto

maior menor será a velocidade). A direita, um *display* exibe a variável ângulo que é atualizada durante a animação.

As Figuras 28 e 29 ilustram o caso de erro. A Figura 28 ilustra a edição de uma *function* de controle, onde omitiu-se um “;” e a mesma foi submetida ao servidor, que por sua vez retornou a mensagem do compilador mostrada na Figura 29.

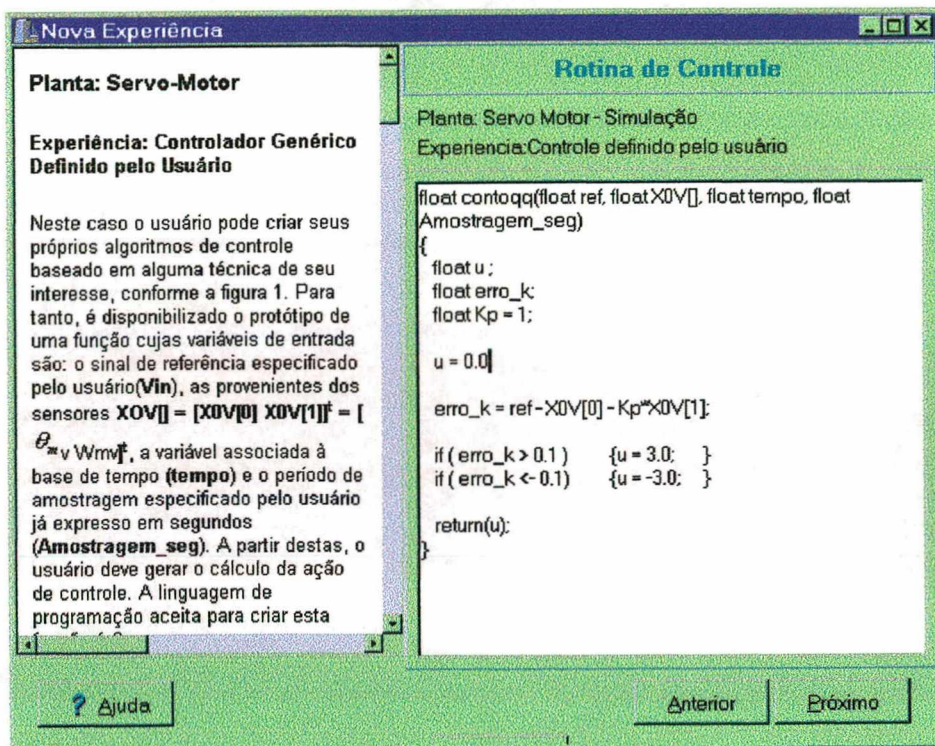


Figura 28 – *Function* com erro.

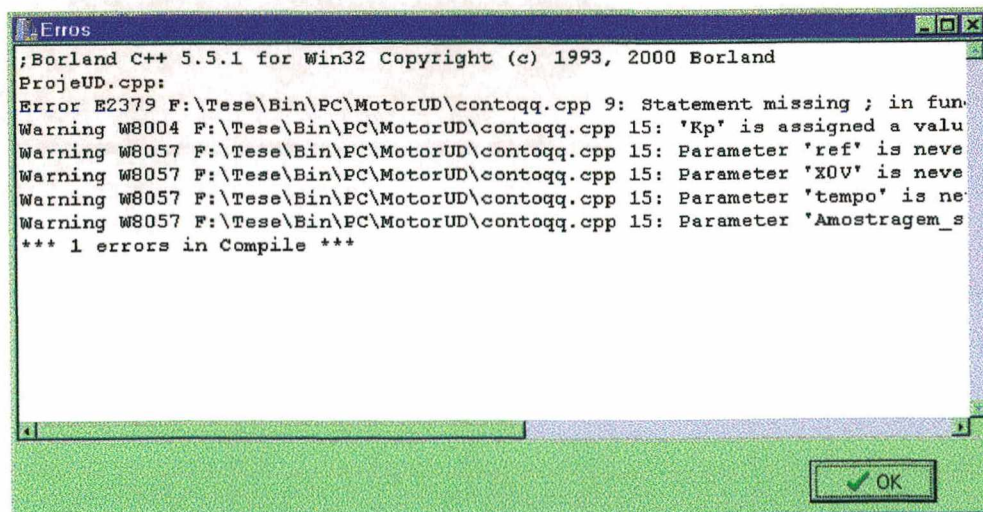


Figura 29 – Mensagem de erro do compilador.

5.5 – Conclusões

A co-orientação da dissertação de mestrado [46] permitiu uma aproximação aos principais conceitos relacionados à programação distribuída e, principalmente, uma melhor caracterização de várias especificações do modelo, a incorporação de novas especificações (já incluídas no capítulo 4) e a materialização do modelo num intervalo de tempo mais curto.

Quanto aos módulos desenvolvidos, testes preliminares foram realizados envolvendo experimentos de simulação e experimentos com o servomotor real. Estes testes ocorreram tanto com programas cliente instalados em computadores da própria rede da UFPA, quanto com programas cliente instalados em computadores de outras redes (CEFET-PA.- Escola Técnica Federal do Pará) e via conexão discada acessando o servidor da UFPA, ou da UFSC – Universidade Federal de Santa Catarina. Em relação ao número de clientes conectados simultaneamente, ainda não foram realizados testes com um número maior que cinco.

Conclusões, sucessos, dificuldades, melhorias e futuros trabalhos são apresentados e discutidos no próximo capítulo.

CAPÍTULO 6 – CONCLUSÕES

Neste trabalho foi proposto um modelo de laboratório de experimentação remota, voltado ao ensino – aprendizagem de sistemas de controle e apresentado os componentes de um protótipo que o implementa. Neste capítulo, as conclusões extraídas durante seu desenvolvimento são apresentadas no item 6.1. O item 6.2 trata das sugestões para trabalhos futuros.

6.1 – Conclusões

O objetivo geral foi alcançado, ou seja, o levantamento das aplicações voltadas a sistemas de controle de forma remota, em conjunto com a classificação estabelecida sobre a natureza das informações: envio de parâmetros ou de códigos de programas; e sobre a natureza do experimento: com equipamentos reais ou simulados, propiciou uma melhor compreensão das características dos laboratórios de experimentação remota, além de permitir a visualização de lacunas que justificaram o desenvolvimento de um novo laboratório.

Quanto aos objetivos específicos, as seguintes conclusões foram obtidas em relação ao estudo de caso abordado no capítulo 3:

- A flexibilidade tecnológica é fundamental na escolha de equipamentos para uso de forma remota, pois permite o acompanhamento para possível apropriação de novas tecnologias;
- O grau de intervenção do usuário é um fator restritivo no uso de equipamentos de forma remota, principalmente pela exigência da garantia das suas condições de operação, implicando num aumento da complexidade do problema de autonomia;
- A experimentação remota não se aplica facilmente a qualquer equipamento;
- Alguns equipamentos devem ser utilizados exclusivamente de forma presencial;
- Além da seleção de equipamentos com potencial de uso de forma remota, deve-se adotar critérios para escolha do equipamento a ser utilizado (no trabalho adotou-se o seguinte: permitir tanto o uso de técnicas simples como sofisticadas de algoritmos de controle);
- Deve-se criar condições mais favoráveis para a aprendizagem, como por exemplo, permitindo a possibilidade de trabalhos colaborativos e interfaces integradas ;
- Os custos associados à experimentação remota são quando muito equivalentes a soluções presenciais;

- A questão de segurança existe, tanto quanto em situações presenciais. É mais abrangente, do que no caso presencial, contudo, sua solução passa a ser muito mais técnica;
- A experimentação remota é uma solução viável para o problema de acesso aos meios, e proporciona um compartilhamento mais racional de recursos dentro de uma instituição de ensino ou, entre instituições de ensino.

Em relação ao modelo, ele acrescenta contribuições no que se refere a uma maior exploração dos conteúdos e conceitos relacionados a sistemas de controle, e no que se refere à incorporação de mídias consideradas relevantes no processo ensino – aprendizagem, dentre as quais destacam-se:

- Uso de um algoritmo de controle arbitrário, pela disponibilização de um protótipo de uma *função* a partir do qual o usuário pode criar a sua própria.;
- Uso de um controlador contínuo ou discreto, considerando que o período de amostragem é uma variável a ser especificada pelo usuário o que lhe permite trabalhar com os dois tipos de controladores.;
- Inclusão de novos experimentos, a partir da estratégia de se associar a cada experimento um programa aplicativo específico para sua realização, e não de um único programa central que permitisse a realização de vários experimentos;
- Experimentos com equipamentos reais ou simulados, considerando que, se por um lado pode ser impraticável o uso de equipamentos reais, por outro, o uso de experimentos simulados pode dentre outras coisas trazer muito conhecimento sobre um problema específico;
- Transferência de parâmetros ou de códigos de programa, dando mais liberdade ao usuário por exemplo, em pesquisar novos algoritmos de controle;
- Autonomia para a realização de experimentos específicos, a partir da inclusão de recursos da própria teoria de controle, para resolver problemas relacionados a inicialização de um experimento;
- Autonomia para garantir a integridade de equipamentos reais, pela inclusão de *software* de supervisionamento;
- Comunicação entre usuários, como uma maneira de garantir a possibilidade de trabalhos colaborativos;
- Visualização de mensagens geradas pelo compilador em caso de erro, permitindo portanto, usar o erro como fonte de conhecimento;

- Permite aprender rudimentos de uma linguagem de programação, o que é importante do ponto de vista de uma formação mais completa em sistemas de controle.

Ainda em relação ao modelo, observam-se as seguintes restrições:

- Não abordou a questão do uso ou desenvolvimento de bibliotecas pré-definidas;
- Não abordou os efeitos de perturbações externas;
- Não abordou recursos para auxiliar o usuário na fase de projeto de um controlador, por exemplo, permitindo o traçado de um *root locus*;
- A especificação do sistema operacional *Windows* foi feita em virtude da compatibilidade com o *software* da placa conversora AD/DA, contudo, não é definitiva.

Em relação a implementação, as seguintes conclusões foram obtidas:

- Comprova que o modelo é implementável. Mesmo considerando que durante a fase de estabelecimento do modelo, já se incluíam também alternativas para a implementação, ainda assim, enquanto um protótipo não existisse de fato, sempre existiria uma dúvida sobre sua viabilidade de implementação;
- A co-orientação da dissertação de mestrado citada no capítulo 5, permitiu, principalmente, a incorporação de novas especificações ao modelo, sua materialização num tempo menor e um aprendizado muito grande, em relação a uma atividade que no futuro poderá ser vivenciada com mais frequência;
- O uso da programação distribuída facilita a inclusão de novos experimentos e expansão do laboratório, visto que, sempre se tem a possibilidade de incorporar novos PC's à rede local existente, e conseqüentemente, novos experimentos;
- A inclusão de novos experimentos é simples, no sentido de ter etapas bem definidas e um aplicativo já desenvolvido para tratar da primeira fase;
- O desempenho na *Internet* durante a realização de experimentos, em termos de tempo, pode ser considerado aceitável;
- Uma possível desvantagem é que para cada nova versão do programa cliente, o usuário deverá realizar um *download* para obtê-la.

6.2 – Sugestões para trabalhos futuros

As sugestões para trabalhos futuros serão divididas em dois blocos, um relativo ao modelo, outro a sua implementação.

Quanto ao modelo, um trabalho importante é a inclusão de recursos que auxiliem o usuário na fase de projeto de um controlador e a disponibilização de bibliotecas pré definidas, que também possam auxiliá-lo, por exemplo, na solução de problemas específicos, como o de identificação dos parâmetros do motor usando mínimos quadrados. Outro trabalho sugerido é a replicação do modelo para outra área do conhecimento, por exemplo biologia, com as devidas adaptações, e expansão da concepção de gerenciamento para administrar o modelo atual e sua réplica.

Quanto à implementação, um trabalho necessário é a implementação de todas as especificações do modelo, particularmente as referentes a segurança além de se fazer uma revisão e eventuais correções em relação aos textos de apoio. Sugere-se a substituição do sistema operacional *Windows* pelo *Unix* tendo em vista que este último, por ser de domínio público, permitirá a expansão do laboratório sem a necessidade custos adicionais. Ainda em relação a custos, outra sugestão é o uso de novas tecnologias que permitam diminuir e aumentar a capacidade de expansão do laboratório remoto.

Até o momento da edição deste texto, teve-se acesso a dois trabalhos que podem servir como pontos de partida para o desenvolvimento de algumas das sugestões anteriores e estão citados nas referências [47] e [48].

REFERÊNCIAS

- [1] COMER, E. D. **Interligação em rede com TCP/IP, Princípios, protocolos e arquitetura**. 3 ed. Rio de Janeiro: Campus, 1998. 1 v.
- [2] TRENTIN, M. A. S.; SANTOS, A.V. dos. The virtual laboratory. In: ICECE'2000, CONFERÊNCIA INTERNACIONAL DE ENSINO DE ENGENHARIA E COMPUTAÇÃO, 2000, São Paulo. **Anais...** São Paulo: SENAC, 2000. CD-ROM.
- [3] GOMES, A.V.; DE SOUZA, F. da F. Uma ferramenta de auxílio a educadores no processo de ensino-aprendizagem a distância via Web. In: ICECE'2000, CONFERÊNCIA INTERNACIONAL DE ENSINO DE ENGENHARIA E COMPUTAÇÃO, 2000, São Paulo. **Anais...**, São Paulo: SENAC, 2000. CD-ROM.
- [4] GOMES, A.V.; GOTTGROY, M. de P. B. Um modelo conceitual baseado no processo de educação a distância via Web. In: ICECE'2000, CONFERÊNCIA INTERNACIONAL DE ENSINO DE ENGENHARIA E COMPUTAÇÃO, 2000, São Paulo. **Anais...**, São Paulo: SENAC, 2000. CD-ROM.
- [5] CARNEGIE MELLON UNIVERSITY. The Virtual Lab. **Engineering the future**. Disponível em: <<http://www.ece.cmu.edu/afs/ece/usr/stancil/web/virtual-lab/virtual-lab.html>> .Acesso em: 03 mar. 1999.
- [6] RENATO E. SABBATINI. **Telemedicina: A Assistência a distância**.Disponível em : <<http://www.nib.unicamp.br/papers/reporter-medico-03.htm>,>. Acesso em: 02 dez 2001.
- [7] MILGRAM, P.; ZHAI, S. A telerobotic virtual control system Proc. SPIE, vol. 1612, Cooperative intelligent robotics in space II, Boston, Nov. 11-13, 1991.
- [8] BARTHOLOMEU, M., LIMA, E.R.; BASTOS-FILHO, T. **Learning robotics through the Internet** ICECE'2000, Conferência Internacional de ensino de Engenharia e Computação, SENAC, São Paulo, Brasil, Agosto, 2000.

- [9] SABER ELETRÔNICA. n. 306, p. 6-12. julho, Saber Ltda., São Paulo: Brasil, 1998.
- [10] ZEILMANN, R. P.; GOMES da Silva Jr. ,J.; PEREIRA, C. E.; BAZANELA, A. S. **Uma estratégia para controle e supervisão de plantas industriais através de Internet.** Disponível em: <<http://www.delet.ufrgs.br/vsbai/Abstracts/119.html>> Acesso em: 02 dez. 2001.
- [11] OGATA, K. **Engenharia de Controle Moderno.** Rio de Janeiro: Prentice Hall do Brasil, 1982.
- [12] PHILLIPS, C. L.; NAGLE JR, H.T. - **Digital Control System Analysis and Design,** Englewood Cliffs: Prentice-Hall, Inc. 1993.
- [13] FRANKLIN, G.F.; POWELL, D. **Digital Control of Dynamic Systems.** Addison [s.l.], Wesley Publishing Company, 1980.
- [14] WISINTAINER, M. A. **RexLab: Laboratório de Experimentação Remota com o Microcontrolador 8051,** 1999. Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 1999.
- [15] UNIVERSITY OF TEXAS AT AUSTIN. **Department of Mechanical Engineering the Dynamic Systems and Controls lab.** Disponível em : <<http://www.me.utexas.edu/~lotario/me244/labs/simulation/pendulum/uwpend.html>> Acesso em: 22 mar. 1999.
- [16] UNIVERSITY OF WESTERN AUSTRALIAN. **Telerobot on the Web.** Disponível em: <<http://telerobot.mech.uwa.edu.au/>>. Acesso em: 22 jun. 1999
- [17] UNIVERSIDADE DE BRASÍLIA. Faculdade de Tecnologia. **RobotWebCam,** Disponível em: <<http://www.graco.unb.br/robwebcam>>. Acesso em: 22 jun. 1999.

- [18] UNIVERSIDADE FEDERAL DE SANTA CATARINA. **RexLab – Laboratório de Experimentação Remota**. Disponível em: <<http://www.inf.ufsc.br/~jbosco/labvir.htm/>>. Acesso em: 05 jan. 1999.
- [19] UNIVERSITY OF TENNESSEE AT CHATTANOOGA. **UTC, Internet Laboratory Server in Engineering Systems Laboratory**. Disponível em: <<http://chem.engr.utc.edu/>>. Acesso em: 08 out. 1999.
- [20] UNIVERSIDADE FEDERAL DO PARÁ. **Laboratório de controle e sistemas do departamento de engenharia elétrica**. Disponível em: <<http://www.ufpa.br/dee/lacos/histo.html>>. Acesso em: 07 fev. 1999.
- [21] Feedback Instruments. **Operational Amplifier Trainer – OAT 343**. [1994]. (Manual 343, Eda 0689).
- [22] Feedback Instruments. **Process Control Simulator – PSC 327**. [1994], (Manual 327-1 Eda 0392 e 327-2 Eda 0790).
- [23] Feedback Instruments. **Servo Fundamental Trainer – SFT 154**. [1994], (Manual 154 Eda 0193).
- [24] Feedback Instruments. **Process Trainer – PT 326**. [1994], (Manual 356 Eda 0275).
- [25] SOUZA, L. M.; EIKAWA, D. S. **Reestruturação e controle do processo físico placa e ar a partir de ganhos escalonados**. 1997. Trabalho de Conclusão de curso. Curso de Engenharia Elétrica, Universidade Federal do Pará, Belém, 1997.
- [26] NEVES, M.R. **Construção de Protótipo para Laboratório de Ensino e Pesquisa: Sistema “Ball beam”**. 1999. Trabalho de Conclusão de curso. Curso de Engenharia Elétrica, Universidade Federal do Pará, Belém, 1999.
- [27] GOMES, A. C. G.; Barbosa M. R. R.; Silva, O. F.; Oliveira, L. A. H. **A Didatic Prototype for Automation and control**. In: ICECE'2000, CONFERÊNCIA

INTERNACIONAL DE ENSINO DE ENGENHARIA E COMPUTAÇÃO, 2000, São Paulo. **Anais...** São Paulo: SENAC, 2000. CD-ROM.

- [28] ATOS. **Controlador Lógico Programável MPC2002**. Belém: DEE/LACOS, manual, 1997.
- [29] Educational Control Products. **Inverted Pendulum ECP**. Belém: DEE/LACOS, (Manual for model 505, 1997).
- [30] Educational Control Products. **Rectilinear Control System ECP**. Belém: DEE/LACOS, manual for model 210/210a 1994.
- [31] Advantech Co. **PCL – 812PG Enhanced Multi-Lab Card**. Belém: DEE/LACOS, manual do usuário, 1993.
- [32] HANSELMAN, D; LITTLEFIELD, B. **MATLAB Versão do estudante e guia do usuário**. São Paulo: Makron Books do Brasil, 1999.
- [33] SOBRINHO, E. G.de A. **Manual de utilização do SIMULACOS: ambiente computacional para simulação de sistemas dinâmicos**. 1993. Trabalho de Conclusão de curso. Curso de Engenharia Elétrica, Universidade Federal do Pará, Belém, 1993.
- [34] BARRA JUNIOR, W. **Ambiente de software para programação de controladores lógico programáveis**. In: CONGRESSO DE EDUCAÇÃO TECNOLÓGICA DOS PAÍSES DO MERCOSUL, 1998, Pelotas. **Anais...** Pelotas: [s.l.], 1998. p.
- [35] AGAH, A. **Human interactions with intelligent systems: research taxonomy**. Computers and Electrical Engineering p.71-107, 2001.
- [36] SENA, J. A. da S. - **Biblioteca de classes para a simulação de sistemas de potência – BCSSP & SimuSinc 2.0**. 1998. Trabalho de conclusão de curso – Curso de Engenharia Elétrica, Universidade Federal do Pará, Belém, 1998.

- [37] SILVA, O. F., BRANDÃO, J. T. R. **VLS1ST – Programa para simulação do primeiro estágio do VLS**. São José dos Campos: CTA/IAE, 1989. Relatório técnico.
- [38] BARRA JUNIOR, W. **Estratégias Neuro-Fuzzy adaptativas aplicadas ao controle de Sistemas de Potência**, 2001. Tese (Doutorado) – Programa de Pós Graduação em Engenharia Elétrica, Universidade Federal do Pará, Belém, 2001.
- [39] SILVA, O. F.; NEVES, M. R.; ALVES, J. B. Metodologia para a Construção de Protótipos didáticos: Um estudo de caso. In: ICECE'2000, CONFERÊNCIA INTERNACIONAL DE ENSINO DE ENGENHARIA E COMPUTAÇÃO, 2000, São Paulo. **Anais...**, São Paulo: SENAC, 2000. CD-ROM.
- [40] BECKER, F. **Aprendizagem e ensino: contribuições da epistemologia genética**. In: **FORMAÇÃO do engenheiro: desafios da atuação docente, tendências curriculares e questões contemporâneas da educação tecnológica**, Florianópolis: Editora da UFSC, 1999. p. 179-196.
- [41] RAMOS, E. M. F. **O papel da avaliação educacional nos processos de aprendizados autônomos e cooperativos**. In: **FORMAÇÃO do engenheiro: desafios da atuação docente, tendências curriculares e questões contemporâneas da educação tecnológica**, Florianópolis: Editora da UFSC, 1999. p. 208-230.
- [42] BARRA JUNIOR., W.; SILVA, O. F.; BARREIROS, J. A. L.; SOUZA, L. M. Identificação *fuzzy* adaptativa de uma planta não linear do tipo placa túnel de vento. In: **CONGRESSO BRASILEIRO DE AUTOMÁTICA, 12.**, 1998, Uberlândia.. **Anais...**, Uberlândia: SBA, 1998. v.5. p. 1779-1784.
- [43] BARRA Jr., W. **Desenvolvimento de um controlador fuzzy self-learning (inteligente)**. 1997. Dissertação (Mestrado) – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Pará, Belém. 1997.
- [44] REISDORPH, K. **Borland C++ Builder**. [s.l.] : SAMS Publishing, 1998.

- [45] RAMOS, E. M. F. **Análise ergonômica do sistema hipernet buscando o aprendizado da cooperação e da autonomia.** 1996. Tese (Doutorado) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis.1996.
- [46] OSCAR, A.K. **Ambiente Para Simulação Remota de Sistemas Dinâmicos Baseado na Arquitetura Cliente-Servidor.** 2001. Dissertação (Mestrado) – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Pará, Belém. 2001.
- [47] BERNHARDT, M. de F.; PALADINI, S. **Microservidor CHIRON** 2001. Trabalho de Conclusão. Curso de Ciência da Computação, UNISUL, Campus de Araranguá, Araranguá, 2001.
- [48] PONT, L. D.; CRISTIANO, M. A. da S. **LAMSTER to Scientific Research,** 2001. Trabalho de Conclusão. Curso de Ciência da Computação, UNISUL, Campus de Araranguá, Araranguá, 2001.

ANEXO A – SISTEMAS DE CONTROLE

A Figura 30 ilustra alguns dos principais elementos de um sistema de controle em malha fechada. A planta é um sistema dinâmico que pode ser desde um servomotor conectado a uma carga ou um foguete. O bloco sensor representa elementos capazes de realizar medições de sinais provenientes da planta $y(t)$. Por exemplo, num servomotor um sensor normalmente existente é o tacômetro que fornece um sinal de tensão proporcional a velocidade angular do eixo do motor. O bloco de soma algébrica fecha a malha e gera um erro, $e(t)$, entre a resposta desejada $r(t)$ ou referência e a resposta atual da planta. O bloco controlador é um elemento que a partir do valor do erro, determina qual o valor da entrada a ser aplicada na planta, tal que ocorra uma diminuição do erro, isto é, a resposta a ser gerada pela planta se aproxima cada vez mais da resposta desejada. Por fim, como via de regra o sinal gerado pelo controlador, $m(t)$, é de baixa potência, o atuador é um dispositivo que visa elevar a potência do sinal do controlador, gerando o sinal $u(t)$ que será efetivamente aplicado na planta [11].

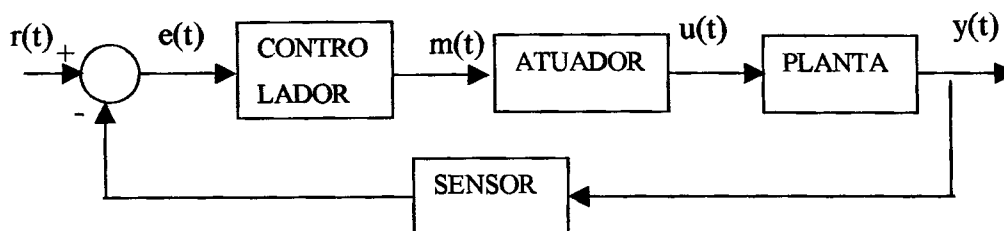


Figura 30 – Sistema de controle em malha fechada.

Geralmente um controlador processa o sinal de erro de modo que um certo critério de controle ou especificação seja satisfeito. Este critério pode envolver por exemplo:

- rejeição de distúrbios;
- erro em regime permanente;
- resposta transitória;
- sensibilidade a mudanças de parâmetros na planta.

E a solução do problema de controle normalmente envolve:

- a escolha de sensores e atuadores adequados para a planta;
- o desenvolvimento (construção) de modelos para a planta, para os sensores e para os

atuadores;

- o projeto do controlador baseado nos modelos desenvolvidos e no critério de controle (especificação);
- as simulações digitais para validação do controlador;
- os testes no sistema real.

Dentre as diversas estruturas de controladores existentes, apenas duas são apresentadas a seguir, uma representante da teoria de controle clássico, outra, da teoria de controle moderno que são, respectivamente, o controlador PID (proporcional – integral - derivativo) e a realimentação de estados.

O controlador PID é constituído pela combinação de três ações de controle: a proporcional, a integral e a derivativa. Possuindo as vantagens de cada uma das três ações de controle individuais. Seu modelo é dado pela equação (1):

$$m(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt} + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (1)$$

ou em termos de função de transferência pela equação (2):

$$\frac{M(s)}{E(s)} = K_p \left(1 + T_d s + \frac{T_i}{s} \right) = G_c(s) \quad (2)$$

onde K_p representa a sensibilidade proporcional, T_d representa o tempo derivativo e T_i o tempo integral. Um diagrama de blocos deste controlador é mostrado na Figura 31.

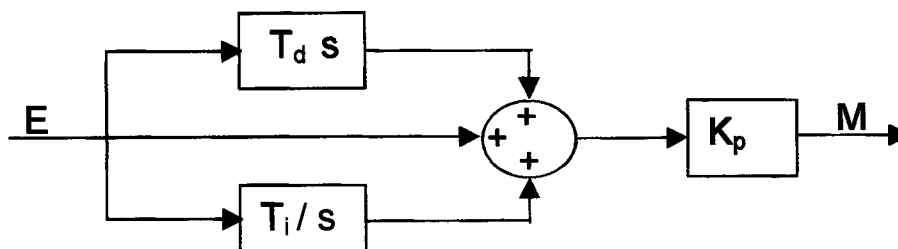


Figura 31 – Controlador PID.

Para se obter um controlador usando realimentação de estados, a planta deve ser representada por suas equações de estado e de saída, como mostrado na equação (3), para o caso linear e invariante no tempo.

$$\begin{aligned}\dot{\mathbf{X}}(t) &= \mathbf{A}\mathbf{X}(t) + \mathbf{B}u(t) \\ y(t) &= \mathbf{C}\mathbf{X}(t) + \mathbf{D}u(t)\end{aligned}\quad (3)$$

Se o vetor de estados, $\mathbf{x}(t)$ é totalmente disponível, ou seja, se todas as variáveis de estado podem ser medidas (ou pelo menos estimadas), realimenta-se cada uma das variáveis de estado devidamente ponderadas por ganhos (valores reais), para gerar um controlador por realimentação de estados.

Dependendo da natureza do sinal de referência, duas estruturas são definidas para este controlador: sendo nula a referência tem-se o problema de regulação e sendo a referência um sinal dependente do tempo, tem-se o problema do rastreamento.

Para a regulação pode-se propor como ação de controle a equação (4),

$$u = -\mathbf{K} \cdot \mathbf{x} = -[K_1 \quad K_2 \quad K_3 \quad K_4] \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}\quad (4)$$

e o diagrama de blocos do sistema realimentado é mostrado na Figura 32.

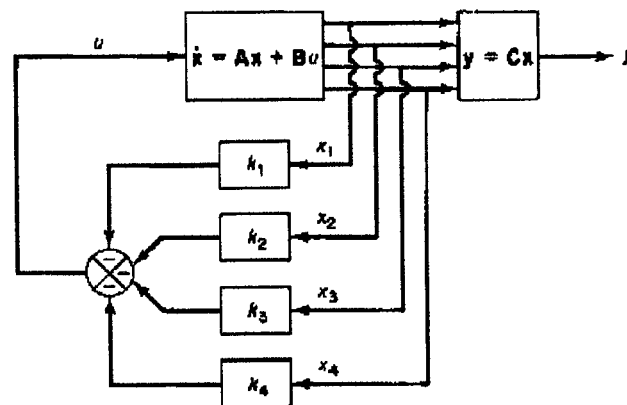


Figura 32 – Diagrama de blocos para o problema de regulação.

Para o problema de rastreamento, onde a saída y , igual a variável de estado x_1 , deve acompanhar o sinal de referência variante no tempo $r(t)$, injetado na entrada do sistema, pode-se propor como ação de controle a equação (5),

$$u = -\mathbf{K} \cdot \mathbf{x} + K_1(r - y) = -[K_1 \quad K_2 \quad K_3 \quad K_4] \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + K_1(r - C\mathbf{x}) \quad (5)$$

e o diagrama de blocos do sistema realimentado é mostrado na Figura 33.

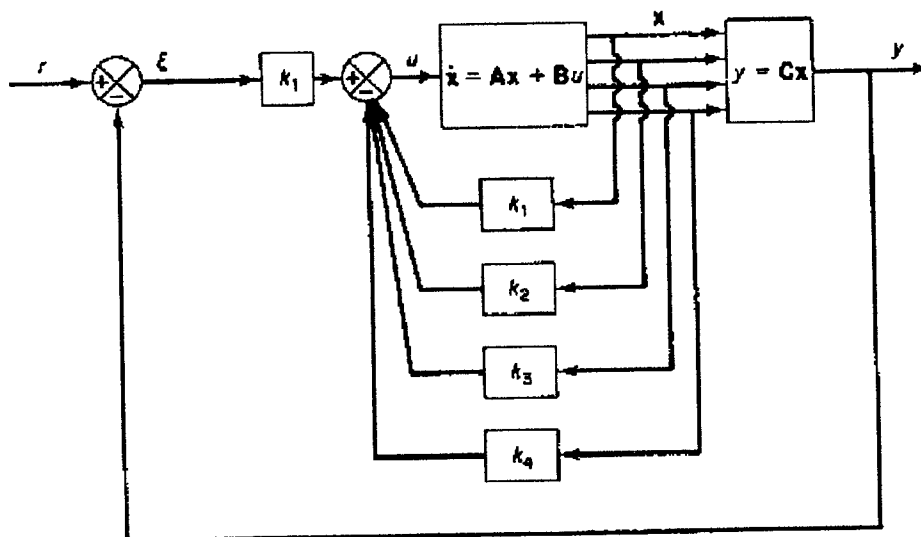


Figura 33 – Diagrama de blocos para o problema de rastreamento.

ANEXO B – SINAIS DE TESTE TÍPICOS

Alguns dos sinais de teste que se costuma utilizar em sistemas de controle são mostrados nas Figuras 34, 35 e 36, onde: “A” corresponde a amplitude do sinal e “T” o período, para o caso de sinal periódico [11].

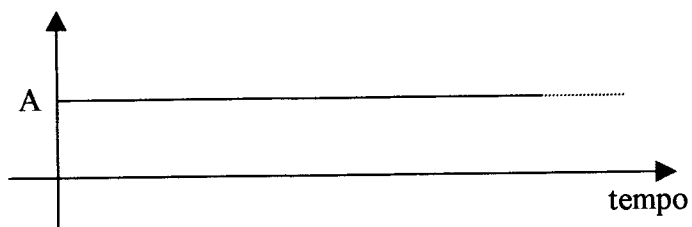


Figura 34 – Sinal degrau.

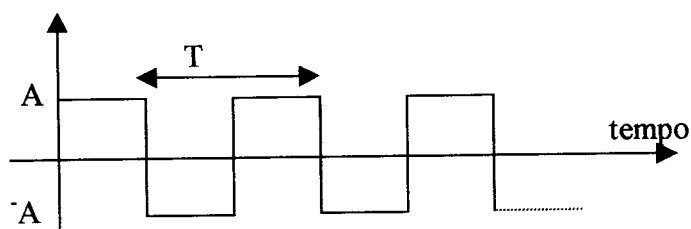


Figura 35 – Sinal onda quadrada.

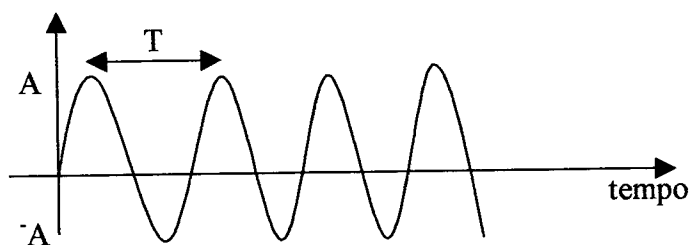


Figura 36 – Sinal senoidal.

ANEXO C – MODELO MATEMÁTICO DE SISTEMAS FÍSICOS

A maioria dos sistemas dinâmicos reais sejam, mecânicos, elétricos, térmicos, hidráulicos, econômicos ou biológicos, podem ser descritos matematicamente por equações diferenciais. Deste modo, a resposta de um sistema dinâmico real quando sujeito a uma dada entrada (ou função de excitação), pode ser obtida a partir da solução das equações diferenciais que o descrevem [11].



Figura 37 – Representação de sistemas físicos.

As equações diferenciais que descrevem as características dinâmicas de um dado sistema físico, são também denominadas modelo matemático, e podem ser obtidas a partir das leis físicas que governam um sistema em particular, como exemplo, as lei de Newton para sistemas mecânicos e as de Kirchhoff para sistemas elétricos.

No processo de construção de um modelo matemático para a solução de um problema, deve-se estabelecer um compromisso entre a simplicidade do modelo e a precisão dos resultados por ele gerado. É recomendável que inicialmente se construa um modelo simplificado, de modo a se ganhar um conhecimento básico e geral, ainda que se comprometa a precisão, e posteriormente, um modelo mais completo poderá ser elaborado e utilizado para a obtenção de resultados com maior precisão.

Uma classe de sistemas dinâmicos que apresenta modelos simples são os lineares e invariantes no tempo. O modelo para tais sistemas se constitui em equações diferenciais lineares a parâmetros constantes. Alguns exemplos são apresentados a seguir.

Um sistema translacional mecânico é constituído de uma massa m , sobre a qual pode-se aplicar a força $u(t)$, provocando um deslocamento $y(t)$ e tendo como reações, a força de uma mola $f_k(t)$ com constante de elasticidade K e a de um amortecedor viscoso $f_b(t)$ com constante de amortecimento B , conforme a Figura 38.

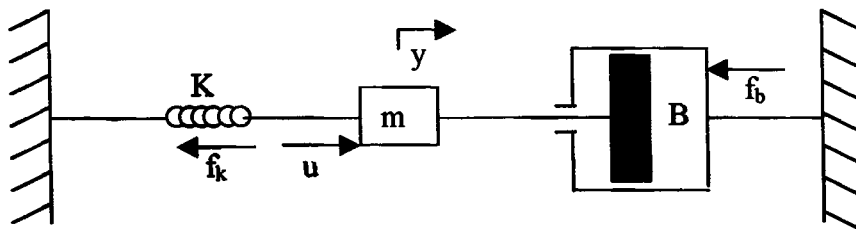


Figura 38 – Sistema translacional mecânico.

A lei de Newton para sistemas de translação é dada pela equação (6),

$$\Sigma F_i = ma \quad (6)$$

ou seja, a resultante das forças é igual ao produto da massa pela aceleração, assim, para o sistema da Figura 38, a equação (6) resulta na equação (7),

$$u(t) - f_k(t) - f_B(t) = m \frac{d^2 y(t)}{dt^2} \quad (7)$$

e como a força na mola e do amortecedor viscoso são dadas pela equação (8), então,

$$\begin{aligned} f_k(t) &= K \cdot y(t) \\ f_B(t) &= B \frac{dy(t)}{dt} \end{aligned} \quad (8)$$

substituindo a equação (8) na equação (7), o modelo final é dado pela equação (9).

$$u(t) = m \frac{d^2 y(t)}{dt^2} + B \frac{dy(t)}{dt} + Ky(t) \quad (9)$$

Um sistema rotacional mecânico é constituído de uma carga com momento de inércia J , sobre a qual pode-se aplicar um torque $T(t)$ provocando um deslocamento angular $\theta(t)$ com velocidade $w(t)$, tendo como reações um torque $T_b(t)$, devido a um amortecedor viscoso com constante de fricção viscosa B , como mostrado na Figura 39.

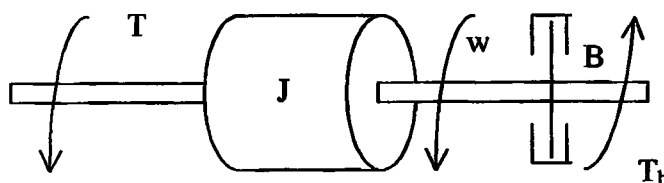


Figura 39 – Sistema rotacional mecânico.

A lei de Newton para sistemas de translação é dada pela equação (10), ou seja, a

$$\Sigma T_i = J\alpha \quad (10)$$

resultante dos torques é igual ao produto do momento de inércia da carga pela aceleração angular, assim para o sistema da Figura 39, a equação (10) resulta na equação (11).

$$T(t) - T_b(t) = J \frac{d\omega(t)}{dt} \quad (11)$$

Como o torque devido ao atrito viscoso é dado pela equação (12),

$$T_b(t) = B\omega(t) \quad (12)$$

o modelo final é dado pela equação (13).

$$T(t) = J \frac{d\omega(t)}{dt} + B\omega(t) \quad (13)$$

Um Circuito L-R é constituído de um indutor com indutância L , em série com um resistor de resistência R , sobre os quais pode-se aplicar uma tensão $e_i(t)$ provocando uma corrente $i(t)$ e quedas de tensão $e_L(t)$ e $e_R(t)$ no indutor e resistor respectivamente, conforme é mostrado na Figura 40.

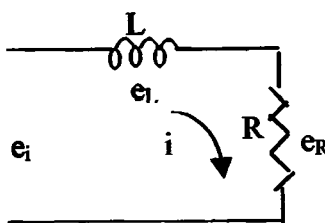


Figura 40 – Circuito L-R.

A lei de Kirchoff para quedas de tensão é dada pela equação (14),

$$\sum V_i(t) = 0 \quad (14)$$

ou seja a soma algébrica das quedas de tensão num circuito fechada é igual a zero, então, para o circuito da Figura 40, a equação (14) resulta na equação (15),

$$e_i(t) - e_R(t) - e_L(t) = 0 \quad (15)$$

e como as tensões no resistor e indutor são dadas pela equação (16),

$$\begin{aligned} e_R(t) &= Ri(t) \\ e_L(t) &= L \frac{di(t)}{dt} \end{aligned} \quad (16)$$

e o modelo final é mostrado na equação (17).

$$e_i(t) = L \frac{di(t)}{dt} + Ri(t) \quad (17)$$

Se a saída, a ser observada, for a tensão no resistor $e_R(t)$, e não a corrente de circulação $i(t)$, então o modelo resultante é o da equação (18),

$$e_i(t) = \frac{L}{R} \frac{de_R(t)}{dt} + e_R(t) \quad (18)$$

e se for a tensão no indutor, então, o modelo é dado pela equação (19)

$$e_i(t) = e_L(t) + \frac{R}{L} \int e_L(t) dt \quad (19)$$

De uma maneira genérica, os sistemas lineares invariantes no tempo são descritos por equações diferenciais da forma mostrada na equação (20),

$$a_0 y^{(n)}(t) + a_1 y^{(n-1)}(t) + \dots + a_{n-1} \dot{y}(t) + a_n y(t) = b_0 u^{(m)}(t) + b_1 u^{(m-1)}(t) + \dots + b_{m-1} \dot{u}(t) + b_m u(t); m \geq n \quad (20)$$

onde $y(t)$ é a saída do sistema, $u(t)$ é a entrada e os sobre escritos correspondem a derivadas, como indicado na equação (21),

$$\begin{aligned} \dot{u}(t) &= \frac{du(t)}{dt}, \dots, u^{(m)}(t) = \frac{d^m u(t)}{dt^m} \\ \dot{y}(t) &= \frac{dy(t)}{dt}, \dots, y^{(n)}(t) = \frac{d^n y(t)}{dt^n} \end{aligned} \quad (21)$$

e os a_i 's ($i = 0, 1, \dots, n$) e b_j 's ($j = 0, 1, \dots, m$) são coeficientes reais constantes.

ANEXO D – REPRESENTAÇÕES DE MODELOS MATEMÁTICOS DE SISTEMAS FÍSICOS

Para o caso de sistemas lineares e invariantes no tempo, duas representações alternativas podem ser obtidas a partir das equações diferenciais que descrevem o sistema. A primeira se baseia no uso da transformada de Laplace e é denominada função de transferência, a segunda, se baseia no uso de variáveis de estado e é denominada representação de estados. Suas definições são [11]:

- a função de transferência é definida como sendo a relação entre a transformada de Laplace da saída e a transformada de Laplace da entrada de um sistema, considerando-se nulas todas as condições iniciais;
- o espaço de estados de um sistema dinâmico é o menor conjunto de variáveis (chamadas variáveis de estado) tal que, o conhecimento destas variáveis no instante $t = t_0$, juntamente com a entrada para $t \geq t_0$, determina completamente o comportamento do sistema para qualquer instante $t \geq t_0$.

Para sistemas lineares e invariantes no tempo, estas representações assumem as seguintes formas: equação (22) para a função de transferência,

$$\frac{Y(s)}{U(s)} = \frac{N(s)}{D(s)} = G(s) \quad (22)$$

onde:

- s é uma variável complexa da forma $\sigma + j\omega$;
- $Y(s)$: é a resposta do sistema;
- $U(s)$: é a entrada do sistema;
- $N(s)$: é um polinômio em s de grau m ;
- $D(s)$: é um polinômio em s de grau n , ($n \geq m$);
- $G(s)$: é a função de transferência.

e equação (23) para o espaço de estados,

$$\begin{aligned}\dot{X}(t) &= AX(t) + Bu(t) \\ y(t) &= CX(t) + Du(t)\end{aligned}\quad (23)$$

onde:

- $X(t) = [x_1(t) \ x_2(t) \dots \ x_n(t)]^t$, é vetor de estados ($n \times 1$);
- $u(t)$: é a entrada do sistema (para apenas uma entrada e uma saída (1×1));
- A : é uma matriz de coeficientes reais ($n \times n$);
- B : é um vetor de coeficientes reais ($n \times 1$);
- C : é um vetor de coeficientes reais ($1 \times n$);
- D : é um vetor de coeficientes reais (1×1);
- n : é a ordem do sistema.

As duas representações serão obtidas para um sistema translacional mecânico, cujo modelo é desenvolvido no ANEXO C, e mostrado na equação (24).

$$u(t) = m \frac{d^2 y(t)}{dt^2} + B \frac{dy(t)}{dt} + Ky(t) \quad (24)$$

Para determinação da função de transferência, toma-se a transformada de Laplace, usando tabelas, de ambos os lados da equação D.3, resultando na equação (25),

$$\begin{aligned}Ms^2 Y(s) + BS Y(s) + KY(s) &= U(s) \\ \frac{Y(s)}{U(s)} &= \frac{1}{Ms^2 + Bs + K}\end{aligned}\quad (25)$$

que é a representação do sistema translacional mecânico por função de transferência, e que pode ainda, ser representada pelo diagrama de blocos da Figura 41.

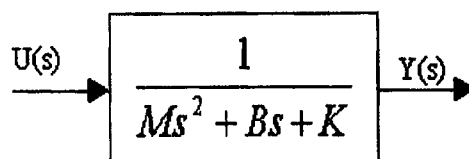


Figura 41 – Diagrama de blocos do sistema translacional mecânico.

Para a representação por espaço de estados, o número de variáveis de estado é sempre igual a derivada de maior ordem do modelo do sistema, neste caso dois (equação 24). Assim, deve-se definir duas variáveis de estado, contudo, esta definição não é única. Sejam então, por exemplo, x_1 e x_2 , definidos pela equação (26).

$$\begin{aligned} \dot{x}_1 &= y \\ \dot{x}_2 &= y \end{aligned} \quad (26)$$

Tomando-se a primeira derivada das definições anteriores, ou seja da equação (26), resulta na equação (27).

$$\begin{aligned} \ddot{x}_1 &= \dot{y} \\ \ddot{x}_2 &= \dot{y} \end{aligned} \quad (27)$$

Substituindo a segunda definição da equação (26) na primeira definição da equação (27), e na segunda definição da equação (27) o modelo da equação (24) considerando neste as definições da equação (26), resulta na equação (28).

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\frac{K}{M} & -\frac{B}{M} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1/M \end{bmatrix} u(t) \\ y &= [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \quad (28)$$

A equação (28) pode ser representada de forma matricial como na equação (29),

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\frac{K}{M} & -\frac{B}{M} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1/M \end{bmatrix} u(t) \\ y &= [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \quad (29)$$

que é a representação no espaço de estados do sistema translacional mecânico.

ANEXO E – IDENTIFICAÇÃO DE SISTEMAS

A identificação de sistemas, consiste no processo de construção de um modelo e estimação dos melhores valores de parâmetros desconhecidos, a partir de dados experimentais, sendo estes dados obtidos basicamente pela aplicação de um sinal de entrada pré definido no sistema a ser identificado, e o registro de sua resposta. A partir dessa massa de dados (entrada - resposta) e do estabelecimento de um modelo, procura-se determinar os parâmetros do modelo, baseando-se em algum método [11], [12].

Dentre os métodos possíveis considera-se um extremamente simples que consiste na análise de resposta em regime permanente e transitória. Tal método não é de uso geral, aplicando-se principalmente para sistemas onde se tem conhecimento de um modelo a partir de leis físicas e para o qual deseja-se estimar os parâmetros. Em particular, considera-se como exemplo um servomotor representado no diagrama de blocos da Figura 42,

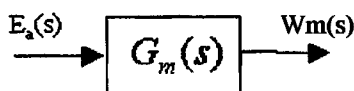


Figura 42 – Servomotor.

onde a função de transferência do motor é dada na equação (30),

$$G_m(s) = \frac{K}{Ts + 1} = \frac{W_m}{E_a} \quad (30)$$

sendo:

- E_a : a tensão aplicada na armadura do motor [V];
- W : a velocidade angular no eixo do motor [rad/s];
- K : ganho do motor;
- T : constante de tempo do motor.

A análise de regime permanente permite a determinação do ganho K do motor. O procedimento consiste na aplicação de uma tensão E_a , de magnitude E (constante) no motor. Em regime, o motor atingirá uma velocidade W_m de magnitude W (constante). Pelo teorema do valor final, “ s ” tende a zero na equação E.1, resultando a equação (31),

$$K = \frac{W}{E} \quad (31)$$

ou seja, o ganho do motor passa a ser simplesmente o quociente entre a magnitude do sinal de entrada e a magnitude da resposta.

A análise da resposta transitória permite a determinação da constante de tempo T do motor. O procedimento consiste no registro da resposta transitória quando da aplicação de uma tensão de magnitude constante (E). Neste caso, utiliza-se a seguinte definição de constante de tempo: o intervalo de tempo necessário para a resposta atingir 63% do seu valor de regime. Dado que é conhecido o valor de regime (e_{ss}), pode-se calcular o ponto da curva de resposta transitória correspondente a 63% deste valor e o instante de tempo em que ocorre, conseqüentemente tem-se a constante de tempo, como é ilustrado na Figura 43.

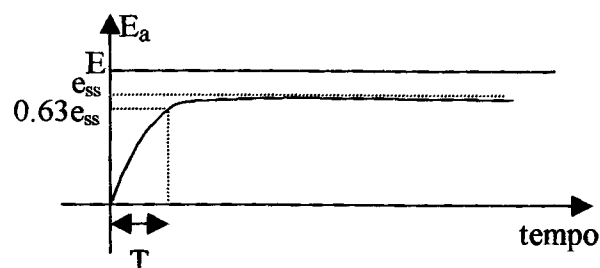


Figura 43 – Constante de tempo.

ANEXO F – MODELO MATEMÁTICO DE UM SERVOMOTOR

Na Figura 44, tem-se o diagrama esquemático de um motor D.C. controlado por armadura, com tensão de campo (e_f) constante [11], [12].

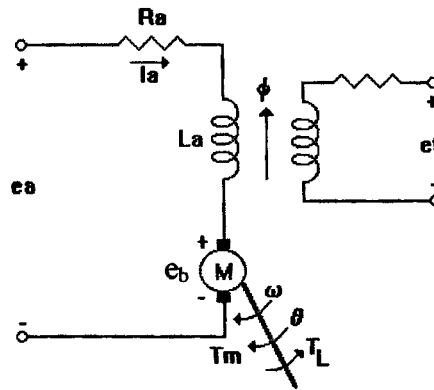


Figura 44 – Esquema de um motor D.C. com excitação independente.

Suas equações dinâmicas são dadas pelas equações (32), (33) e (34):

$$e_b(t) = K_b \frac{d\theta_m(t)}{dt} = K_b W_m(t) \quad (32)$$

$$e_a(t) = e_b(t) + R_a I_a(t) + L_a \frac{dI_a(t)}{dt} \quad (33)$$

$$T_m(t) = K_t I_a(t) = B_m \frac{d\theta_m}{dt} + J_m \frac{d^2\theta_m(t)}{dt^2} + T_L(t) \quad (34)$$

onde:

- e_b : é a força contra eletromotriz [V];
- K_b : é a constante de força contra eletromotriz [V/rad/s];
- θ_m : é o deslocamento angular do eixo do motor [rad];
- W_m : é a velocidade angular no eixo do motor [rad/s];
- e_a : é a tensão de armadura [V];

- I_a : é a corrente no enrolamento de armadura [A];
- R_a : é a resistência do enrolamento de armadura [ohms];
- L_a : é a indutância do enrolamento de armadura [H];
- T_m : é o torque fornecido pelo motor [N.m];
- K_i : é a constante de torque do motor [N.m/A];
- J_m : é o momento de inércia do motor [Kg.m²];
- B_m : é o coeficiente de atrito viscoso do motor [Kg.m/rad/s];
- T_L : é o torque da carga [N.m].

Considerando: desprezível a indutância do enrolamento de armadura, o motor a vazio ($T_L(t) = 0$) e as equações (32), (33) e (34), o seguinte modelo é obtido:

$$J_m \frac{d^2\theta_m(t)}{dt^2} + \frac{B_m R_a + K_i K_b}{R_a} \frac{d\theta_m(t)}{dt} = \frac{K_i}{R_a} e_a \quad (35)$$

que pode ser representado em termos de função de transferência, tomando sua transformada de Laplace, resultando na equação (36):

$$G_m(s) = \frac{K_m}{T_m s + 1} = \frac{W_m}{E_a} \quad (36)$$

onde:

- $K_m = K_i / (R_a B_m + K_i K_b)$, é a constante de ganho do motor;
- $T_m = R_a J_m / (R_a B_m + K_i K_b)$, é a constante de tempo do motor.

ou, em termos de espaço de estados, com a definição das seguintes variáveis de estado da equação (37),

$$\begin{aligned} x_1 &= \theta \\ x_2 &= \dot{\theta} \end{aligned} \quad (37)$$

por:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\left(\frac{1}{T_m}\right) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_m}{T_m} \end{bmatrix} e_a \quad (38)$$

O modelo apresentado anteriormente, equação (36) ou (38), será utilizado para representar um motor D.C. real, no caso, o servomotor da unidade mecânica do equipamento *servo fundamental trainer* da *Feedback*. Neste caso, além do servomotor D.C., têm-se: um sensor tacométrico acoplado diretamente ao eixo do motor que gera uma tensão proporcional a velocidade angular, um sistema de redução de velocidade por correias com relação 1/32 e após esta redução de velocidade, os seguintes sensores: um potenciômetro que gera uma tensão proporcional a posição angular, um disco de acrílico graduado que permite uma leitura visual da posição angular em graus e um *display* onde se pode fazer a leitura visual da velocidade angular em rotações por minuto (RPM). Desta maneira, um diagrama de blocos para o servomecanismo é ilustrado na Figura 45.

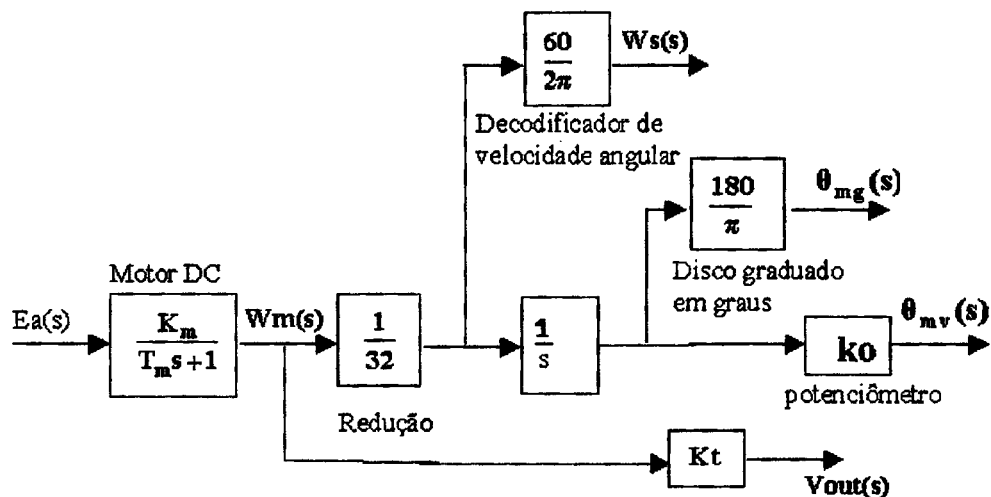


Figura 45 – Diagrama de blocos do servomotor da *Feedback*.

Onde:

- $W_s(t)$: é a velocidade angular do eixo do motor, após redução [RPM];
- k_o : é a constante potenciométrica [V/rad];
- $\theta_{mv}(t)$: é o deslocamento angular do eixo do motor [V];
- $\theta_{mg}(t)$: é o deslocamento angular do eixo do motor [°];
- k_t : é a constante tacométrica [V/rad/s];
- $V_{out}(t)$: é a tensão de saída do tacômetro [V].

Para experimentos com o motor real, os sinais $\theta_{mv}(t)$ e $V_{out}(t)$ provenientes dos sensores, são enviados a um PC via placa de aquisição de dados para processamento, em

seguida, do PC, é enviado também via placa, a tensão de armadura E_a a ser aplicada no motor. Esta leitura e escrita de sinais têm de ocorrer a cada período de amostragem (T_s), como ilustrado na Figura 46. Um período de amostragem extremamente pequeno, da ordem de vinte vezes menor que a constante de tempo do motor, corresponde a implementação de um controlador contínuo no PC, caso contrário, com um período de amostragem entre cinco e dez vezes menor que a constante de tempo do motor, tem-se a implementação de um controlador discreto.

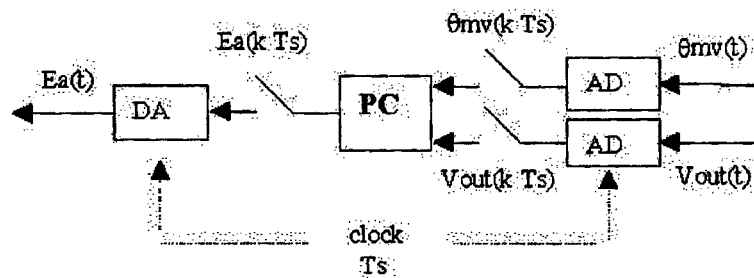


Figura 46 – Sinais lidos dos sensores e sinal enviado para o motor.

Por fim, desde que os sinais $\theta_g(t)$ e $W_s(t)$ podem ser lidos visualmente no equipamento real, estes também são disponibilizados nos experimentos a serem realizados de forma remota apenas para leitura ou visualização gráfica, ou seja, não podem ser usados para efeitos de realimentação.

Para experimentos simulados, a equação da dinâmica do motor (38), que é um sistema contínuo, são integradas com um passo de integração fixo e da ordem de vinte vezes menor que a constante de tempo do motor. O sinal de entrada E_a e os de saída, θ_{mv} e V_{out} são gerenciados por uma lógica que garante suas atualizações apenas a cada período de amostragem, ou seja, enquanto a equação (38) é integrada numericamente de acordo com o passo de integração, a entrada e saídas do motor só são atualizadas a cada período de amostragem. Deste modo, permite-se implementar no PC tanto controladores contínuos, período de amostragem igual ao passo de integração, como discretos, período de amostragem maior que o passo de integração.

DESCRIÇÃO DO MODELO EXPERIMENTAL:

O modelo para a experiência se baseia no servomecanismo da *Feedback* representado no diagrama de blocos da Figura 48. Como entrada aplica-se uma tensão $E_a(t)$ na armadura do motor D.C. que então vai girar com uma certa velocidade $W_m(t)$. Como sensores para as saídas têm-se: um tacômetro acoplado diretamente ao eixo do motor para indicação de sua velocidade angular $V_{out}(t)$, expressa em volts, um potenciômetro de giro livre (após um sistema de redução de velocidade), para indicação da posição angular $\theta_{mv}(t)$, expressa em volts. Por fim, pode-se ler visualmente: a velocidade angular $W_s(t)$ do motor em RPM e a posição angular $\theta_{mg}(t)$ do eixo do motor em graus, ambas após o sistema de redução de velocidade.

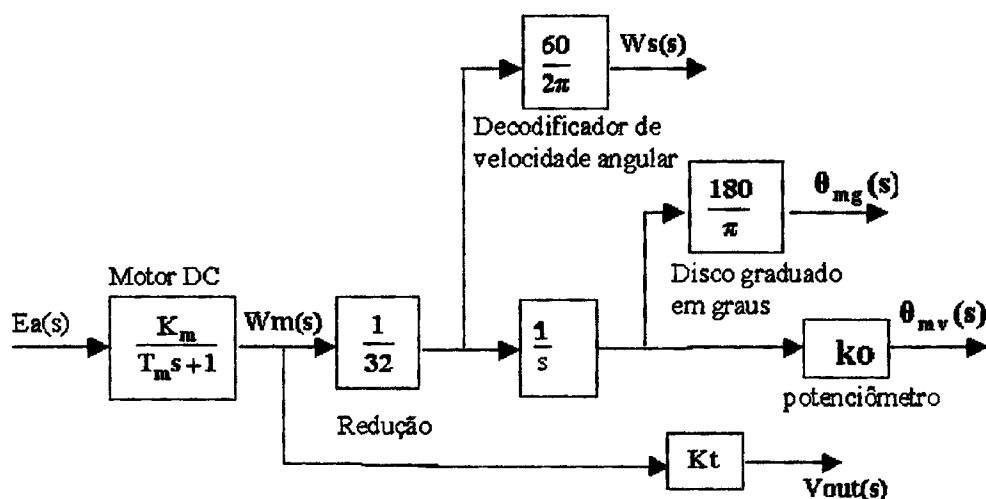


Figura 48 – Modelo para a experiência 1.

SIGNIFICADO E RESTRIÇÕES DE VARIÁVEIS:

1 – Condições iniciais do motor:

$$[\text{teta inicial } \text{omegainicial}]^t = [X0[0] \ X0[1]]^t = [\theta_{mg} \ W_m]^t$$

teta inicial => posição angular inicial em graus; $-180 \leq \text{teta inicial} \leq 180$

omega inicial => velocidade angular inicial em rad/s ; 0 (para motor real)

2 – Tempo total para o experimento: Instante final ≤ 10 (segundos)

3 – Tipo de sinal de entrada a ser aplicado no motor: E_a

Se for onda quadrada ou senoidal, especificar sua amplitude e período, se for degrau apenas sua amplitude, onde: $-5 \leq \text{amplitude (V)} \leq 5$ e $1 \leq \text{período (s)} \leq 10$.

4 – Período de amostragem: período d amostragem = $k \cdot 5$ (milisegundos), $k = 1, 2, 3, \dots$, para $k = 1$, tem-se um sistema contínuo.

PROCEDIMENTO EXPERIMENTAL:

Para o Servomotor simulado ou real, especifique:

$[t_{\text{etainicial}} \ \omega_{\text{megainicial}}]^t = [0 \ 0]^t$; $t_f = 10$; sinal de entrada (E_a) do tipo onda quadrada, com amplitude = 3 e período = 10; e período d amostragem = 5. Em seguida execute a experiência.

1. Com as medidas de E_a [V], ω_s [RPM], V_{out} [V], θ_{mr} [rad] e θ_{mv} [V] em regime permanente, determine K_m , K_t e K_o em concordância com a Figura 48.

Monte uma tabela conforme a indicada a seguir.

Tabela 5 – Determinação de parâmetros do motor e sensores

E_a [volt]	ω_s [RPM]	ω_m [Rad/Seg]	K_m	V_{out} [volt]	K_T	θ_{mv} [volt]	θ_{mg} [graus]	θ_{mr} [rad]	K_o
3									

2. Com o gráfico V_{out} x tempo (ou os resultados numéricos), determine a constante de tempo do motor de acordo com a definição usada para constante de tempo.

3. Repetir (1) e (2) para outras amplitudes da onda quadrada.

ANEXO H – EXPERIÊNCIA 02: CONTROLE DE POSIÇÃO DE UM MOTOR D.C.

OBJETIVOS

Familiarização com o Laboratório Remoto e realizar o controle da posição angular no eixo de saída de um servomotor D.C., em malha fechada, utilizando uma estrutura de controle P (Proporcional) contínua e realimentação apenas da posição angular.

RECURSOS NECESSÁRIOS

- 1) PC
- 2) Conexão com a Internet
- 3) Programa Cliente

DESCRIÇÃO DO MODELO

O modelo inicial para o experimento é mostrado na Figura 49.

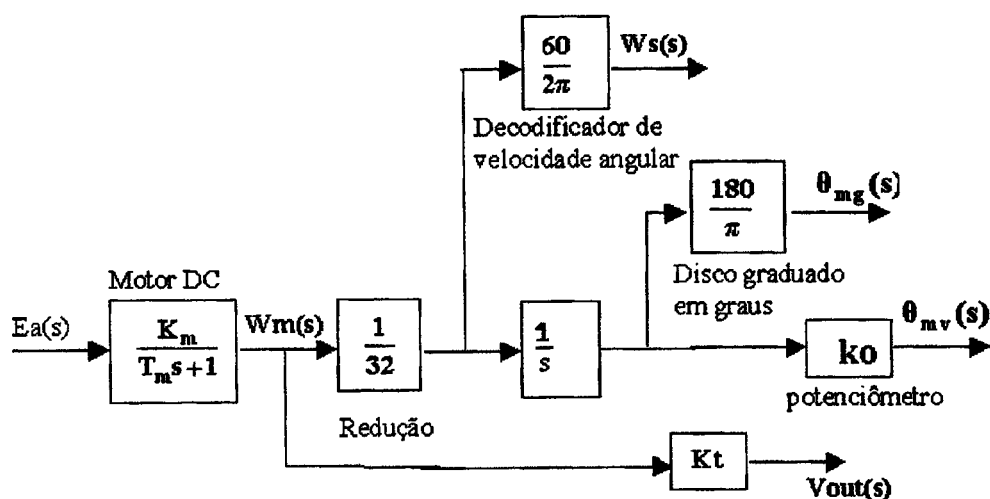


Figura 49 – Modelo para a experiência 2.

RESUMO TEÓRICO

Para o controle de posição, usando apenas realimentação da posição angular no eixo do motor (θ_{mv}) e uma estrutura de controle do tipo P contínua, o laboratório remoto disponibiliza um sistema como o da Figura 50, ou seja, ao invés de uma ação de controle P, tem-se uma PID, neste caso para se ter apenas P deve-se fazer: $T_i = T_d = 0$ e para que este controlador seja contínuo deve-se especificar o menor período de amostragem possível ($T_s = 0.005$). Dois problemas de controle serão abordados nesta experiência: o de rastreamento (a posição

angular no eixo de saída do motor deve acompanhar um sinal de referência pré definido) e o de regulação (a posição angular no eixo de saída do motor deve ser mantida em zero (graus)).

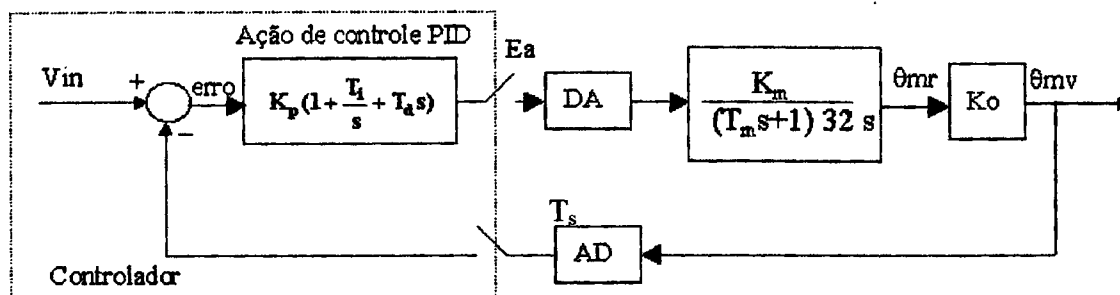


Figura 50 – Controle de posição com ação de controle PID.

SIGNIFICADO E RESTRIÇÕES DE VARIÁVEIS

1- Condições iniciais do motor: $[\text{teta inicial } \omega \text{ inicial}]^t = [X0[0] \ X0[1]]^t = [\theta_{mg} \ W_m]^t$

teta inicial \Rightarrow posição angular inicial em graus; $-180 \leq \text{teta inicial} \leq 180$

omega inicial \Rightarrow velocidade angular inicial em rad/s ; 0 (para motor real)

2 - Tempo total para o experimento: Instante final ≤ 10 (segundos)

3 - Tipo de sinal de referência a ser aplicado no motor: V_{in}

Se for onda quadrada ou senoidal, especificar sua amplitude e período, se for degrau apenas sua amplitude, onde: $-5 \leq \text{amplitude (V)} \leq 5$ e $1 \leq \text{período (s)} \leq 10$.

4 – Período de amostragem, T_s : período d amostragem = $5 \cdot K$, $K = 1, 2, 3 \dots$ (milissegundos)

5 – Ganho da ação de controle K_p . Será estipulado no procedimento experimental

PROCEDIMENTO EXPERIMENTAL

(1) Para o servomotor simulado, e o problema de rastreamento, especifique:

$[\text{teta inicial } \omega \text{ inicial}]^t = [0 \ 0]^t$; $t_f = 10$; sinal de referência do tipo onda quadrada, com amplitude = 5 e período = 10; período d amostragem = 5 (controlador contínuo), e os ganhos da ação de controle: $K_p = 1$, $T_i = T_d = 0$. Observe V_{in} , θ_{mv} e E_a .

(2) Para o servomotor real e o problema de rastreamento, use as mesmas especificações de (1).

(3) Para o servomotor simulado e o problema de regulação, especifique:

$[\text{teta inicial } \omega \text{ inicial}]^t = [45 \ 0]^t$; $t_f = 10$; referência do tipo degrau, com amplitude = 0

período de amostragem = 5, ganhos do controlado $K_p=1$, $T_i=T_d=0$. Observe V_{in} , θ_{mv} e E_a

(4) Para o servomotor real, e o problema de regulação, use as mesmas especificações de (3).

(5) Repita (1) e (2) ou (3) e (4) modificando apenas os ganhos da ação de controle para:

$$K_p = 2, T_i = T_d = 0.$$

ANEXO I – EXPERIÊNCIA 03: CONTROLE DE POSIÇÃO DE UM MOTOR D.C.

OBJETIVOS:

Familiarização com o Laboratório Remoto e realizar o controle da posição angular no eixo de saída um servomotor D.C., em malha fechada, utilizando uma estrutura de controle (contínuo) do tipo relé com zona morta e realimentação da posição e velocidade angular.

RECURSOS NECESSÁRIOS:

- 1) PC 2) Conexão com a Internet 3) Programa Cliente

DESCRIÇÃO DO MODELO:

O modelo inicial para o experimento é mostrado na Figura 51

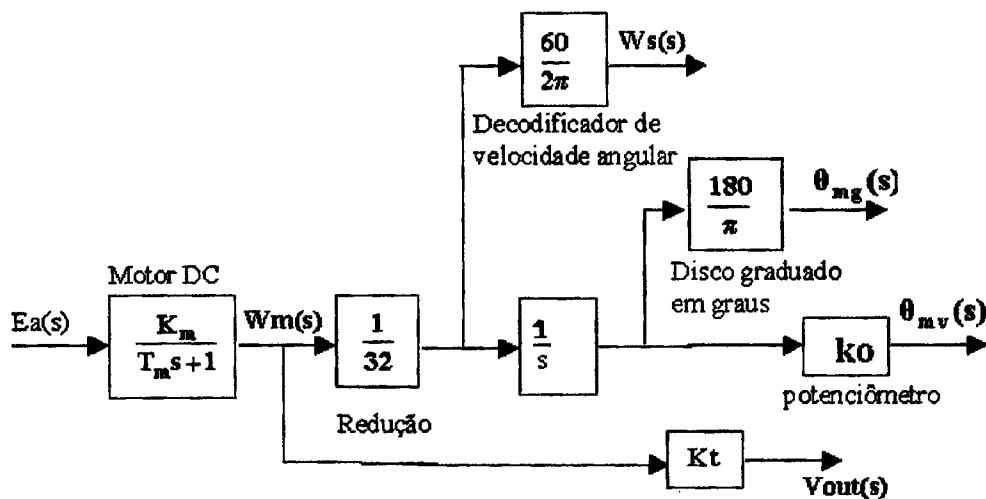


Figura 51 – Modelo para a experiência 3.

RESUMO TEÓRICO:

Para o problema de controle de posição, usando realimentação da posição e velocidade angular (θ_{mv} , V_{out}) e uma estrutura de controle contínuo do tipo relé com zona morta, o laboratório remoto disponibiliza um sistema como o da Figura 52, ou seja, uma *function* que permite a implementação de um controlador qualquer. Neste caso, a ação de controle qualquer deverá implementar um relé com zona morta como mostrado na Figura 53 e o código de programa sugerido para tanto é mostrado na Tabela 6.

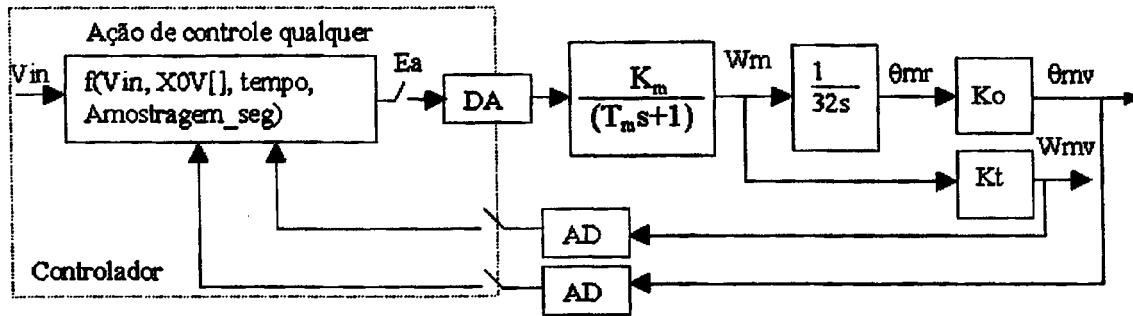


Figura 52 – Controle de posição com ação de controle qualquer.

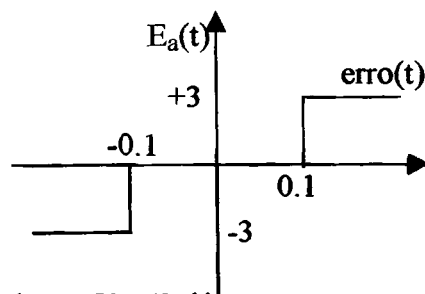


Figura 53 – Relé com zona morta.

Tabela 6 – Listagem para implementação de um relé.

```

Float contoqq(float ref, float X0V[], float tempo, float Amostragem_seg)
{
float u ;
float erro_k;
float Kp = 1 ;
erro_k = ref - X0V[0] - Kp*X0V[1];
u = 0.0;
if(erro_k > 0.1) { u = 3.0 ; }
if(erro_k < -0.1) { u = -3.0 ; }
return(u);
}

```

Este código deve ser editado na janela correspondente do laboratório remoto, e a primeira linha nunca pode ser alterada (note que é a mesma do protótipo que aparece na janela do laboratório remoto).

Dois problemas de controle serão abordados nesta experiência: o de rastreamento (a posição angular no eixo de saída do motor deve acompanhar um sinal de referência pré

definido) e o de regulação (a posição angular no eixo de saída do motor deve ser mantida em zero (graus)).

SIGNIFICADO E RESTRIÇÕES DE VARIÁVEIS

1- Condições iniciais do motor: $[\text{teta inicial } \omega \text{ inicial}]^t = [X0[0] \ X0[1]]^t = [\theta_{mg} \ W_m]^t$

teta inicial \Rightarrow posição angular inicial em graus; $-180 \leq \text{teta inicial} \leq 180$

omega inicial \Rightarrow velocidade angular inicial em rad/s ; 0 (para motor real)

2- Tempo total para o experimento: Instante final ≤ 10 (segundos)

3- Tipo de sinal de referência a ser aplicado no motor: V_{in}

Se for onda quadrada ou senoidal, especificar sua amplitude e período, se for degrau apenas sua amplitude, onde: $-5 \leq \text{amplitude (V)} \leq 5$ e $1 \leq \text{período (s)} \leq 10$.

4- Período de amostragem, T_s : período d amostragem = $5 \cdot K$, $K = 1, 2, 3 \dots$ (milisegundos).

PROCEDIMENTO EXPERIMENTAL

(1) Para o servomotor simulado e o problema de rastreamento, especifique:

$[\text{teta inicial } \omega \text{ inicial}]^t = [0 \ 0]^t$; $t_f = 10$; referência do tipo onda quadrada, com amplitude = 5 e período = 10; e período d amostragem = 5 e edite o código sugerido.

(2) Para o servomotor real e o problema de rastreamento, use as mesmas especificações de (1).

(3) Para o servomotor simulado e o problema de regulação, especifique:

$[\text{teta inicial } \omega \text{ inicial}]^t = [45 \ 0]^t$; $t_f = 10$; referência do tipo degrau, com amplitude = 0 e período d amostragem = 5 e edite o código sugerido.

(4) Para o servomotor real e o problema de regulação, use as mesmas especificações de (3).

ANEXO J – PROGRAMA ADMINISTRADOR

Sua janela principal é mostrada na Figura 54. O botão usuário ativa outra janela que permite: incluir/excluir um usuário (Figura 55) e excluir resultados de experimentos já realizados por um usuário (Figura 56). O botão plantas ativa uma janela para inclusão/exclusão de um experimento (Figura 57 e Figura 58), especificação das variáveis que o descrevem e seus valores iniciais (Figura 59 e Figura 60) e do protótipo de rotina de controle a serem disponibilizados ao usuário (Figura 61) [46].

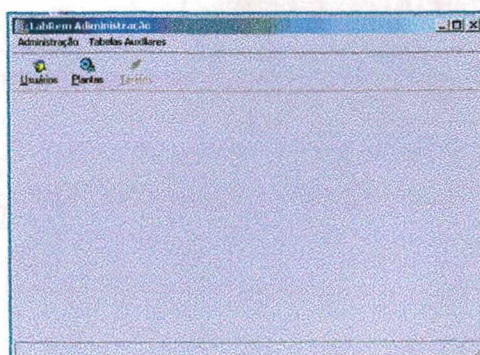


Figura 54 – Janela principal do administrador.

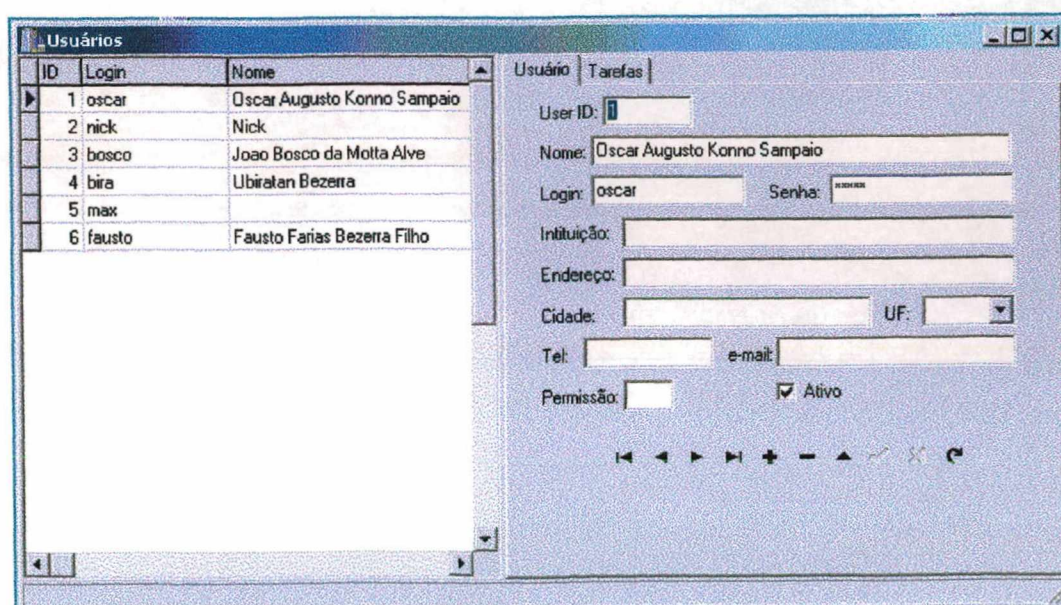


Figura 55 – Janela para inclusão/exclusão de usuários.

Para visualizar as experiências já realizadas por um usuário deve-se ativar a legenda “Tarefas” na janela mostrada na Figura 55, resultando na janela mostrada na Figura 56.

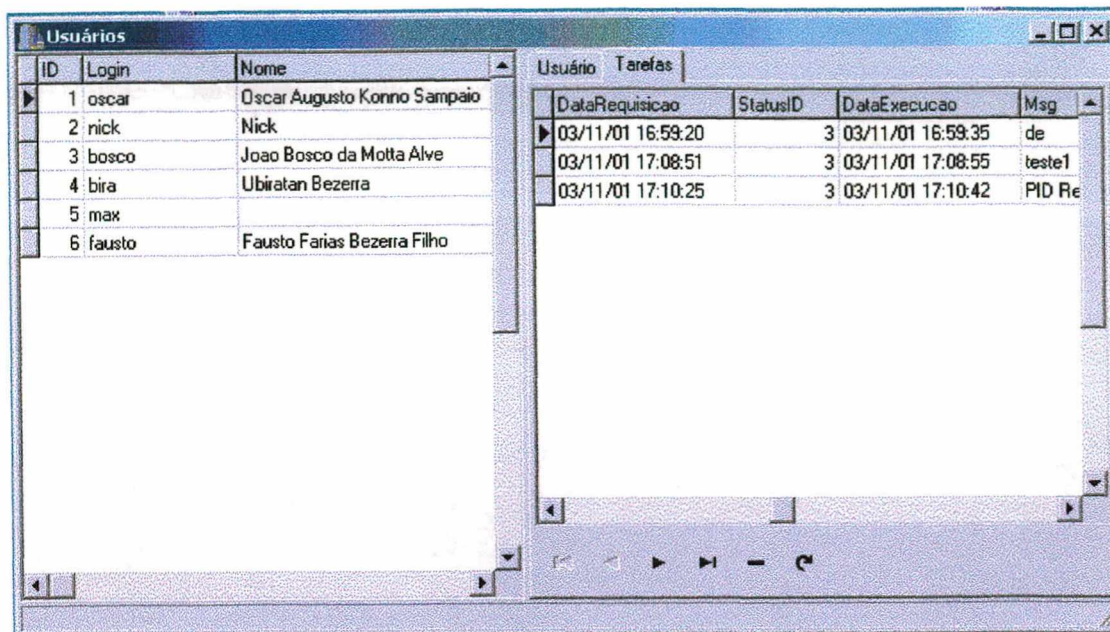


Figura 56 – Tarefas realizadas pelo usuário.

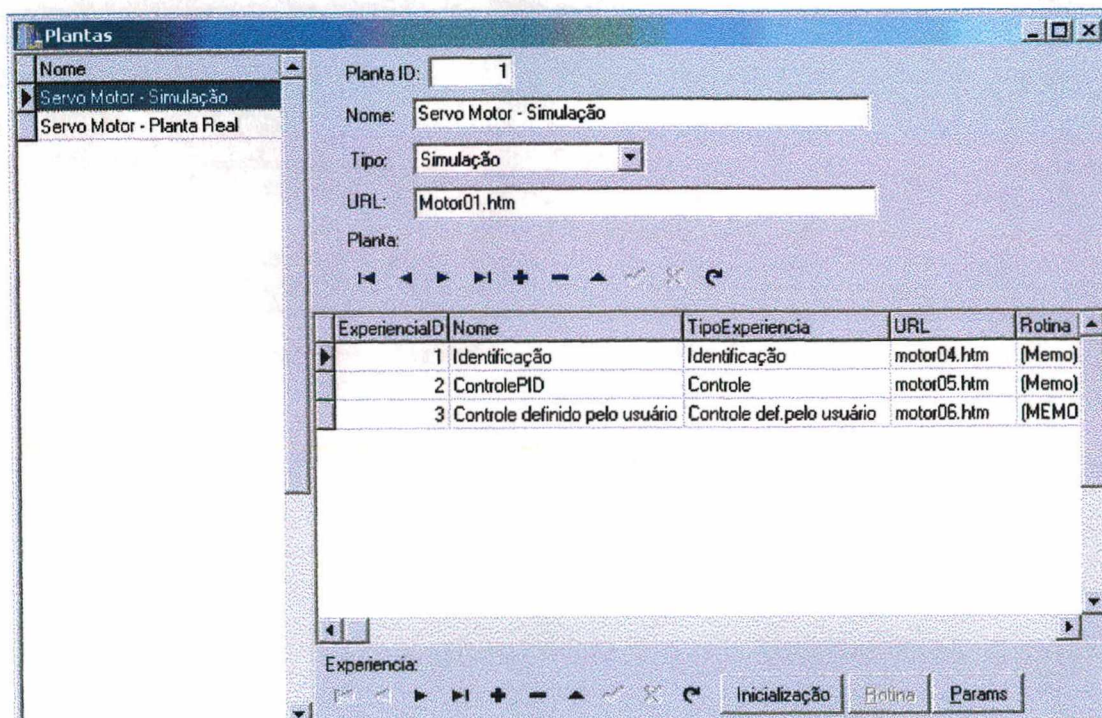


Figura 57 – Janela para inclusão/exclusão de experimentos mostrando os três experimentos que se pode realizar com o motor via simulação.

Selecionando “Servo Motor - Planta real” (canto superior esquerdo da Figura 57), resulta na Figura 58.

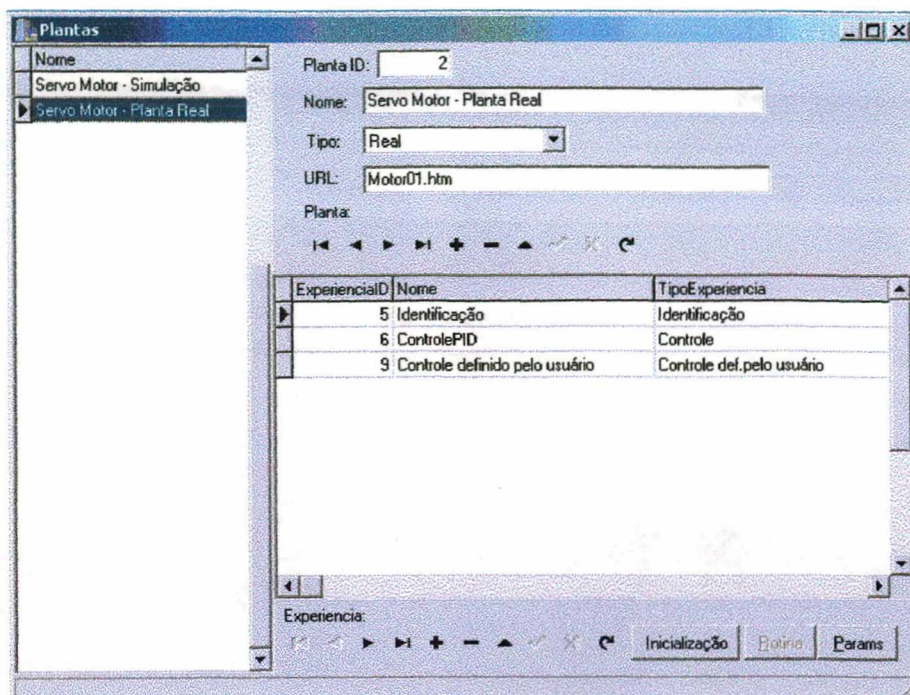


Figura 58 – Janela para inclusão/exclusão de experimentos mostrando os três experimentos que se pode realizar com o motor real.

O botão “Inicialização” (Figura 57), resulta na janela mostrada na Figura 59,

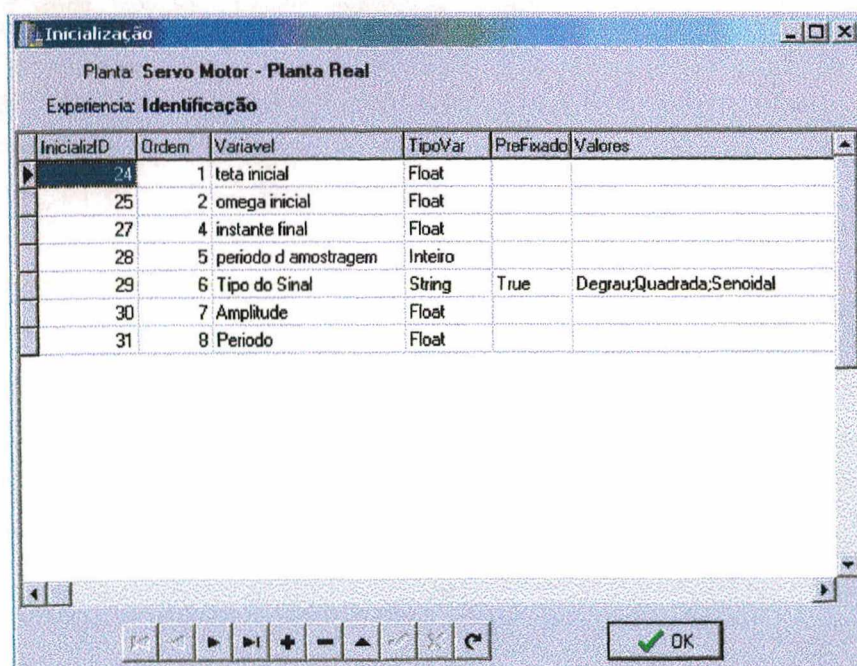


Figura 59 – Definição dos nomes/tipos de variáveis/parâmetros de um experimento.

onde se especifica o nome das variáveis e parâmetros relacionados a um experimento e seus valores iniciais.

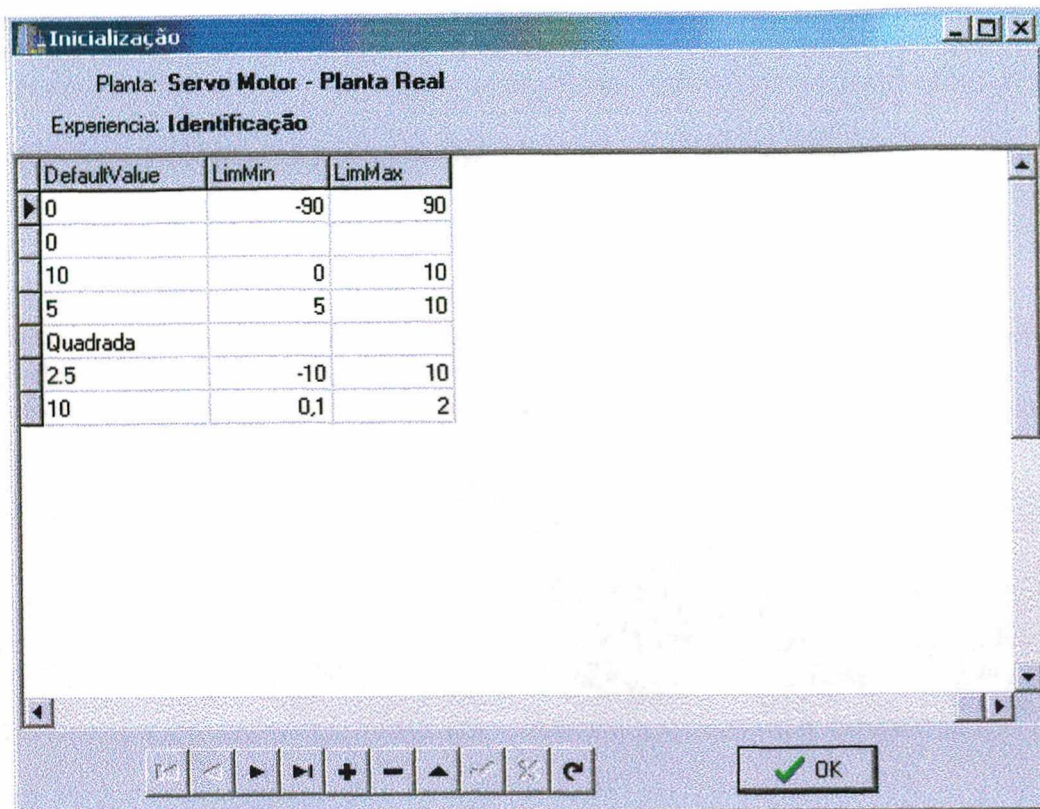


Figura 60 – Continuação da Figura 59.

Selecionando o botão rotina na tela da Figura 57, resulta na tela da Figura 61.

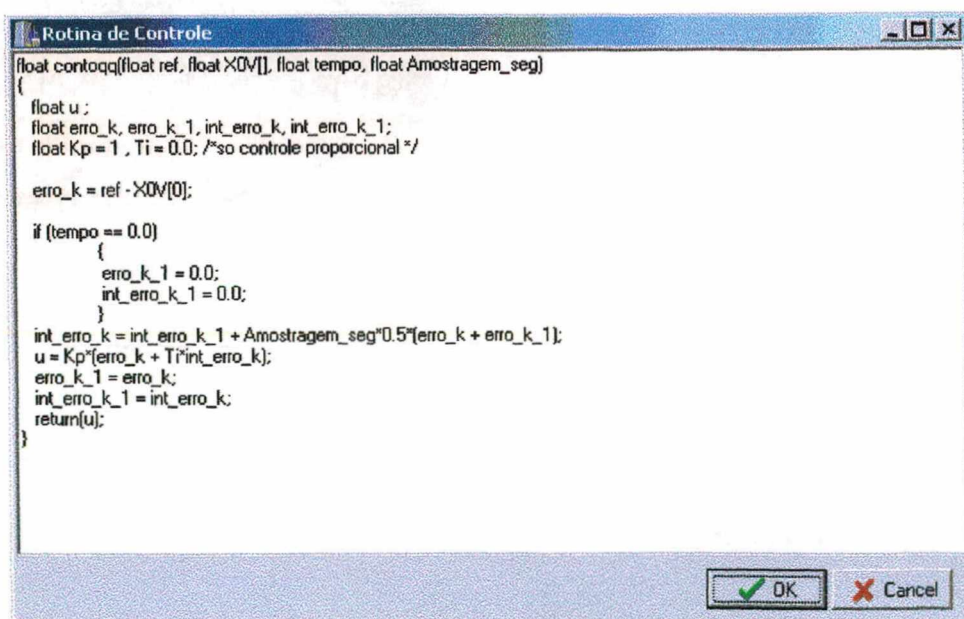


Figura 61 – Protótipo da *function* disponibilizada ao usuário.

Para que se complete a inclusão de um novo experimento, no que se refere ao servidor, deve-se ainda:

- atualizar o arquivo de inicialização “LabRem.ini”, incluindo neste qual o PC da rede irá realizá-la;
- criar as páginas html que serão disponibilizadas durante a experiência e inclui-las no sub diretório HTTP.

O arquivo “LabRem.ini” atual é listado na Tabela 7, com alguns comentários na segunda coluna.

Tabela 7 – Arquivo de inicialização atual do servidor.

<p>[HTTP] RootDir=F:\Tese\Bin\Server\Http</p>	<p>Sub diretório onde devem estar as páginas html</p>
<p>[PARAMS] PAGEMAIN= PAGELAB=LabRem00.htm PAGEPLANTA=LRAAdd01.htm PAGEEXPERIENCIA=LRAAdd02.htm PAGEVARS=LRAAdd03.htm PAGECONTROL=LRAAdd04.htm PAGEFINAL=LRAAdd05.htm</p>	
<p>[PCS] NoPC=6</p>	<p>número de PC's que o servidor vê (inclusive ele) X número de portas</p>
<p>[PC1] Host=localhost Port=5055 Planta=1 Experiencias=1</p>	<p>é o próprio PC onde está o servidor porta a ser usada na experiência é o número associado a PlantaID na Figura 57 é o número associado a ExperienciaID na Figura 57</p>
<p>[PC2] Host=localhost Port=5054</p>	

<p>Planta=1 Experiencias=2</p> <p>[PC3] Host=localhost Port=5056 Planta=1 Experiencias=3</p> <p>[PC4] Host=10.57.1.215 port=5055 Planta=2 Experiencias=5</p> <p>[PC5] Host=10.57.1.215 port=5056 Planta=2 Experiencias=6</p> <p>[PC6] Host=10.57.1.215 port=5054 Planta=2 Experiencias=9</p>	<p>IP do PC que está conectado ao servomotor porta do PC a ser usada na experiência é o número associado PlantaID na Figura 58 é o número associado a ExperienciaID na Figura 58</p>
--	--

A inclusão de um novo experimento termina no PC responsável pela realização do mesmo. Neste PC deve-se instalar: o aplicativo que realizará o experimento em si, o programa gerenciador de experiências, cuja janela principal é mostrada na Figura 62, e seu respectivo arquivo de inicialização que é listado na Tabela 8.

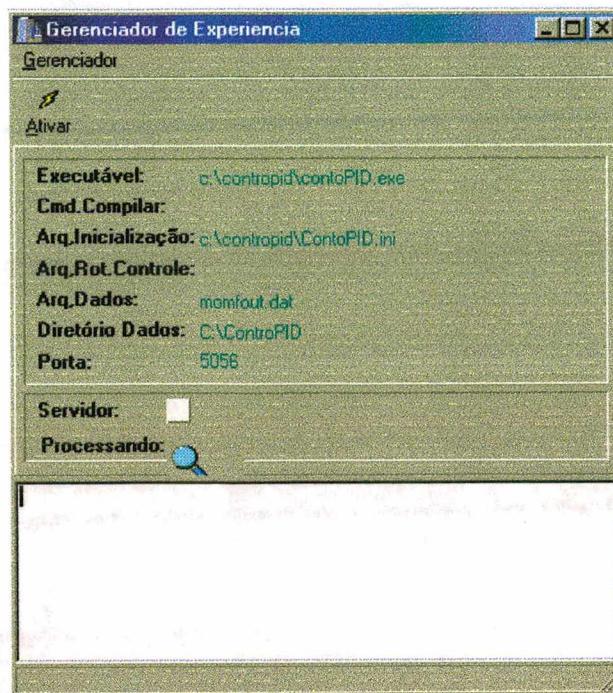


Figura 62 – Janela do gerenciador de experiências para o experimento do PC5 do arquivo de inicialização “LabRem.ini”.

A listagem da Tabela 8, corresponde ao arquivo de inicialização do gerenciador de experiências para o experimento do PC5 do arquivo de inicialização “LabRem.ini” atual.

Tabela 8 – Um arquivo de inicialização.

[GERENCIADOR]	
Executavel=c:\contropid\contoPID.exe	Nome do executável que realizará o experimento.
rotctrl=false	Variável lógica: <i>true</i> chama o compilador, <i>false</i> não.
CmdComp=	Linha de comando para compilação.
ArqIni=c:\contropid\ContoPID.ini	Inicializações especificadas pelo usuário.
ArqCtrl=	<i>Function</i> de controle definida pelo usuário.
ArqDados=momfout.dat	Nome do arquivo de dados .
DirDados=C:\ControPID	Diretório onde os arquivos serão manipulados.
Porta=5056	Porta de acesso do PC para realizar o experimento.

Por fim, uma listagem parcial do código fonte, do aplicativo que realiza a experiência: controle PID com o motor real, ou seja PC5 do arquivo de inicialização “LabRem.ini” atual é mostrada na Tabela 9.

Tabela 9 – Listagem de um aplicativo.

```

/* Programa para controle do servomotor da Feedback usando um PID, */
/* Ko : constante do sensor de posicao angular, [V/rad] */
/* Kt : constante do sensor de velocidade angular, [V/(rad/seg)] */
/* crpg : constante para conversao de radianos para graus */
/* cgpr : constante para conversao de graus para radianos */
/* ref : referencia , [V] */
/* u=ea : acao de controle, [V] */
/* h : intervalo de integracao ,[s] */
/* tf : tempo total de simulacao, [s] */
/* amplitude : amplitude do sinal de referencia,[V] */
/* periodo : periodo do sinal de referencia, [s] */
/* x1=X[0] : posicao angular do eixo do motor, [rad] */
/* x2=X[1] : velocidade angular do eixo do motor, [rad/s] */
/* K : ganho do motor, [(rad/s)/V] */
/* tau : constante de tempo do motor, [s] */
/* Kp, Ti, Td: Ganhos da acao de controle PID */
/* Amostragem: Período de amostragem (um número inteiro), [milisegundos] */
//-----
#include <stdlib.h>
#include <math.h>
//-----
USELIB("../Advantech\ADSAPI\Examples\BCB\LIB\Adsapibc.lib");
//-----
int z, Amostragem, Delta = 2 ;//Amostragem e Delta em milisegundos
const float pi = 3.141592;
const float Ko = 3.326, Kt = 0.026 ;
float ref, ea_e, u_max = 10.0; //u_max em volts
float static X0[2];
float Kp, Ti, Td, X0V[2];
int tipo_do_sinal ;
//-----
int main(int argc, char *argv[])
{
//-----Inicio do programa-----
if (argc<3) {
printf("\nUso MotorMA.exe [INIFILE] [DATAFILE] [ERRORFILE]\n");
return 1;
}

// Constantes para uso posterior
crpg = 180.0/pi; doispi32 = 2.0*pi*32.0 ; cgpr = pi/180.0;
printf("Lendo arquivo de inicialização...\n");
GetPrivateProfileString("INIT", "teta inicial", "0.0",buffer,256,argv[1]);
X0[0]=atof(buffer);
printf("x1=%f\n",X0[0]);

// Verifica o tempo de execução da rotina de controle (para rotina definida pelo usuário)
X0V[0] = 0.0; X0V[1] = 0.0; ref = 2.5; tempo = 0.0;
t_real_i = clock();
u = contopid(ref,X0V,tempo,Amostragem_seg,Kp,Ti,Td);

```

```

t_real = (clock() - t_real_i)/CLK_TCK;
if (t_real > ( (Amostragem - Delta)/1000.0 )
{
    fprintf(fpmferr, "tempo para executar rotina de controle= %f seg\n", t_real);
    fprintf(fpmferr, "maior que o periodo de amostragem= %f seg\n", Amostragem_seg);
    exit(1);
}

// Le (via AD) sinais da fonte para avaliar se não há problemas na fonte** ESTA FALTANDO ***
// Inicializa placa
Inicializa();

EscrDA(0, 5.0);    //canal0 => controle. Anula tensão no motor.
EscrDA(1, 10.);   //canal1 => rele. Fecha rele.
Sleep(10);        //Atraso(miliseg) garante fechamento do relé

//-----  REGULAÇÃO  -----
// Le estado inicial do motor via AD. Caso igual a (0,0) ativa :
// 1- Programa para provocar ligeiro movimento angular (contínuo)
// 2- Programa de controle em malha fechada para regulação (contínuo)
Ler_MAD();
x1_volt = voltage[0];
// 1- Evita que a referência defase de 180 (triângulo no acrílico da UM)
if ( fabs(x1_volt) <= 0.21 ) // Corresponde aproximadamente a 2 graus
{
    z = 0;
    tempo = z*0.005;
    while (tempo <= 0.05)    // Em segundos
    {
        EscrDA(0, 7.5);    // canal0 => controle
        Ler_MAD();
        x1_volt = voltage[0]; //Utilizado na próxima fase(nunca retirar!)
        z = z + 1;
        tempo = z*0.005;
        Sleep(3);        //Amostragem-delta,fixos,independe do usuário
    }
    EscrDA(0, 5.0);        // Para de movimentar
}
// 2- Regulação
Ler_MAD();
x1_volt = voltage[0];
if ( fabs(x1_volt) >= 0.12 ) // Corresponde aproximadamente a 1 grau
{
    z = 0;
    tempo = z*0.005;
    while (tempo <= 2.7)    // Em segundos
    {
        Ler_MAD();
        x1_volt = voltage[0];
        erro = - x1_volt ;    // Cálculo da ação de controle Proporcional
        u = 2.1*erro;
        if ( fabs(u) > u_max ) //Se u > u_max,funciona como relé de +/- 5 V
        {
            u = sinal(u)*u_max; //u:tensao de armadura gerada,-10<=u<=+10
        }
        ea_e = (u + 10.)/2.; //ea_e: " enviada,0<=ea_e<=+10
        EscrDA(0, ea_e);    // Envia tensao de armadura pelo DA
        z = z + 1;        //Incrementa contadores de tempo
        tempo = z*0.005;
        Sleep(3);        //Amostragem-delta,fixo independente do usuário
    }
}

```



```

} // fim do laço while
} // fim do if (fabs(x1_volt) >= 0.12)
//----- Fim da regulação -----

//----- POSICIONAMENTO INICIAL -----
// Se X0[0] (teta inicial) especificado pelo usuário for diferente de zero
// 1) posicionar o motor de acordo com a especificação do usuário, senão,
// 2) ativar controle PID com parâmetros especificados pelo usuário.
// 1- Programa de controle em malha fechada para posicionamento(contínuo)
X0V[0] = X0[0]*cgpr*Ko; // Converte X0[0] de graus para volts
Ler_MAD();
x1_volt = voltage[0];
if ( fabs(X0V[0] - x1_volt) >= 0.12) //aproximadamente 1 grau
{
z = 0; erro = 0.; // Zera contador de tempo e erro
tempo = z*0.005;
while (tempo <= 1.5) // Em segundos
{
Ler_MAD();
x1_volt = voltage[0];
// x2_volt = voltage[1];
erro = X0V[0] - x1_volt ;
u = 1.5*erro;
if ( fabs(u) > u_max ) //Para u > u_max, funciona como
{ // relé de +/- 10 Volts
u = sinal(u)*u_max;
}
ea_e = (u + 10.)/2.;
EscrDA(0, ea_e);
z = z + 1;
tempo = z*0.005;
Sleep(3); //Amostragem-delta=fixo. Independe do usuário
} //fim do laço while
printf("fim posicionamento inicial\n");
} // fim do if(fabs(X0[0]-x1_volt)>=0.12)

// Ativa câmera para gravação de imagens** ESTA FALTANDO **

/** INICIO DO CONTROLE PID COM PARÂMETROS DO USUÁRIO *****
// 2 - Programa de controle em malha fechada com PID
if ( fabs(X0V[0] - x1_volt) <= 0.12) //aproximadamente 1 grau
{
z = 0; erro = 0.; // Zera contador de tempo e erro
Amostragem_seg = Amostragem/1000.; //Amostragem do usuário em segundos
tempo = z*Amostragem_seg;
fprintf(fpmfout,"tempo[s];ref[volt];Ea[volt];teta[grau];teta[volt];omega[volt];omega[RPM]\n");
while (tempo <= tf) // Em segundos
{
Ler_MAD();
X0V[0] = voltage[0];
X0V[1] = voltage[1];
switch (tipo_do_sinal)
{
case 0: ref = degrau(amplitude);break;
case 1: ref = senoide(amplitude,periodo,tempo);break;
case 2: ref = quadrada(amplitude,periodo,tempo);break;
default: fprintf(fpmferr,"Tipo de sinal invalido\n");
exit(1);
}
}
u = contopid(ref,X0V,tempo,Amostragem_seg,Kp,Ti,Td);

```

```

if ( fabs(u) > u_max )           // Avalia valor da amplitude do
{                                 //sinal a ser aplicado no motor;
    fprintf(fpmferr,"Amplitude maior que a admissível,(+/-10 V). Programa abortado\n");
    // Faz todas a saidas DA nulas;
    EscrDA(0, 5.0);              //5 garante V na saída da placa
    EscrDA(1, 0.);
    // Fecha arquivos de dados
    // Desabilita a placa
    exit(1);
}
ea_e = (u + 10.)/2.;
EscrDA(0, ea_e);
// Posição e velocidade em: rad; rad/s; grau; rps
X0[0] = X0V[0]/Ko;              //converte para radiano
X0[1] = X0V[1]/Kt;              //converte para radiano por segundo,
x1_grau = X0[0]*crpg;           //converte para grau
x2_rps = X0[1]/doispi32;        //converte para RPS após a redução,
x2_rpm = x2_rps*60.;            //converte para RPM após a redução,

// Salva: tempo[seg]; ea[volt]; x1[grau]; x1[volt]; x2[volt]; x2[rpm]
fprintf(fpmfout,"%f;%f;%f;%f;%f;%f;%f\n", tempo, ref, u, x1_grau, X0V[0], X0V[0], x2_rpm);

z = z + 1;
tempo = z*Amostragem_seg;
Sleep(Amostragem - Delta);
}                                 // fim do laço while
}                                 //fim do if (fabs(X0[0]-x1_volt)<=0.12)

// Faz todas a saidas DA nulas;
EscrDA(0, 5.0);                  //5 garante V na saída da placa
EscrDA(1, 0.);

// Desativa câmera para gravação de imagens
// Salva em arquivo imagens do experimento
// Fecha arquivos de dados
// Desabilita a placa
return 0;
}                                 //do main

```