

Universidade de Brasília
Faculdade de Tecnologia
Departamento de Engenharia Elétrica

Arquitetura de Laboratório Remoto com Banco de Dados e
Controle no Espaço de Estados

Alfredo Américo de Freitas
Orientador: Dr Ing. Adolfo Bauchspiess
Projeto Final de Graduação

Agradecimentos

Agradeço à Deus por tudo de bom em minha vida.

Agradeço ao meu orientador Adolfo Bauchspiess, sempre presente nos momentos difíceis deste trabalho e por quem tenho enorme admiração.

Aos professores Geovany Borges e Marco Egito pela ajuda incondicional e paciência com minhas dúvidas.

Aos professores de laboratório de Controle Dinâmico, Filomeno, Pedro e Marcelino pela confiança e credibilidade.

Aos técnicos do GAT, que muito me ajudaram neste trabalho.

Aos colegas Hugo Gosmann, Bruno Guimarães, Alexandre Silva, Rafael Jabuonski e Luis Vicente pelos trabalhos que antecederam a este e sem o quais o Sistema de Nível de Líquidos não seria possível.

Aos colegas do LAVSI, Ronaldo de Jesus, Fábio Lobo, Gustavo, e do LCVC Flávio Vidal e Bruno Mariani pela ajuda e apoio durante este trabalho e a todos os colegas da Universidade de Brasília.

A minha mãe Francisca Eliza e minhas irmãs Ocirema e Nathália Eliza de Freitas, por todo o apoio e confiança.

As minhas tias Maria José e Nucleine Freitas pela ajuda incondicional durante minha vida.

À todos meus amigos especialmente a Daniel Veloso e Hebert França por estarem sempre presentes em momentos difíceis.

À todos que em mim acreditaram e aos professores que contribuíram para que eu atingisse meus objetivos.

Por fim, mas não menos importante, agradeço a Alexandra Belchior pela dedicação, amor e confiança.

Este trabalho é dedicado à minha namorada Alexandra Belchior e à minha afilhada Luisa França.

Índice

1. INTRODUÇÃO	I
1.1. OBJETIVO DO TRABALHO.....	2
1.2. ORGANIZAÇÃO DO TRABALHO.....	3
2. SITUAÇÃO ANTERIOR DO SISTEMA DE CONTROLE DE NÍVEL DE LÍQUIDOS .	4
2.1. ARQUITETURA DO LABORATÓRIO.....	5
2.2. ELETRÔNICA DE CONTROLE.....	6
3. SISTEMAS REMOTOS.....	ERRO! INDICADOR NÃO DEFINIDO.
3.1. INTRODUÇÃO	ERRO! INDICADOR NÃO DEFINIDO.
3.2. MOTIVAÇÃO	ERRO! INDICADOR NÃO DEFINIDO.
3.3. USO DA EXPERIMENTAÇÃO REMOTA	ERRO! INDICADOR NÃO DEFINIDO.
3.4. SIMULAÇÃO E EXPERIMENTAÇÃO REMOTA	ERRO! INDICADOR NÃO DEFINIDO.
3.5. REQUISITOS PARA IMPLEMENTAÇÃO DE LABORATÓRIOS REMOTOS	ERRO! INDICADOR NÃO DEFINIDO.
3.6. RESUMO DO CAPÍTULO	ERRO! INDICADOR NÃO DEFINIDO.
4. SISTEMA DE TANQUES ACOPLADOS	7
4.1. INTRODUÇÃO	8
4.2. ARQUITETURA DA PLANTA DE NÍVEL DE LÍQUIDOS	10
4.3. MODELO DOS SENSORES E ATUADORES.....	11
4.3.1. SENSORES DE NÍVEL	11
4.3.2. MOTO-BOMBAS	13
4.4. RESUMO DO CAPÍTULO	17
5. ARQUITETURA DO SERVIDOR	18
5.1. INTRODUÇÃO	18
5.2. SUPERVISÓRIO DO SISTEMA	18
5.3. O TOMCAT, CONTÊINER JSP/SERVLET.....	20
5.4. O PADRÃO DE ARQUITETURA <i>MVC2</i>	21
5.4.1. PADRÕES DE PROJETO E ARQUITETURA	21
5.4.2. O PADRÃO DE ARQUITETURA <i>MODEL VIEW CONTROLLER 2</i>	22
5.5. A <i>FRAMEWORK</i> JAKARTA STRUTS.....	23
5.5.1. A IMPLEMENTAÇÃO DO CONTROLLER E ARQUITETURA STRUTS	23

5.5.2.	A IMPLEMENTAÇÃO DOS <i>VIEWS</i>	25
5.5.3.	<i>TILES</i> COM <i>DESIGN</i> BASEADO EM <i>TEMPLATES</i>	26
5.5.4.	A IMPLEMENTAÇÃO DOS <i>MODELS</i>	28
5.5.5.	INTERNACIONALIZAÇÃO, I18N[32].....	29
5.5.6.	VANTAGENS E DESVANTAGENS DA ARQUITETURA DO LEARN.....	30
5.6.	A INTEGRAÇÃO DO SERVIDOR COM O CONTROLE DE PROCESSOS.....	31
5.7.	RESUMO DO CAPÍTULO.....	32
6.	O BANCO DE DADOS DO LEARN.....	33
6.1.	INTRODUÇÃO.....	33
6.2.	O MODELO DE ENTIDADE RELACIONAL - MER.....	34
6.2.1.	AS TABELAS DO BANCO DE DADOS DO LEARN.....	36
6.2.2.	AS VISÕES DO BANCO DE DADOS DO LEARN.....	39
6.3.	A EXPANSÃO PROJETADA PARA O BANCO DE DADOS.....	39
6.4.	RESUMO.....	40
7.	CONTROLE NO ESPAÇO DE ESTADOS.....	41
7.1.	INTRODUÇÃO.....	41
7.2.	ESPAÇO DE ESTADOS E MÉTODOS DE TRANSFORMADAS.....	42
7.3.	MODELAGEM MATEMÁTICA NO ESPAÇO-DE-ESTADOS.....	43
7.3.1.	MODELO NÃO LINEAR DO SISTEMA.....	43
7.3.2.	MODELO LINEARIZADO NO PONTO DE OPERAÇÃO.....	45
7.4.	SISTEMA DE CONTROLE DE NÍVEL DE LÍQUIDOS SISO.....	48
7.5.	SISTEMA DE CONTROLE DE NÍVEL DE LÍQUIDOS MIMO.....	49
7.6.	IMPLEMENTAÇÃO DO CONTROLADOR NO ESPAÇO DE ESTADOS.....	50
7.7.	RESUMO DO CAPÍTULO.....	51
8.	UTILIZAÇÃO NO LABORATÓRIO DE CONTROLE DINÂMICO.....	52
8.1.	EXECUÇÕES POR ALUNO.....	52
8.2.	EXECUÇÕES POR GRUPO.....	54
9.	CONCLUSÕES.....	56
10.	BIBLIOGRAFIA.....	59

Índice de Figuras

FIGURA 2-1 SUPERVISÓRIO ANTERIOR, LIMITA AO CONTROLE PID[30]	4
FIGURA 2-2 ARQUITETURA WEB DO LABORATÓRIO[31]	5
FIGURA 2-3 CIRCUITO DE CONDICIONAMENTO DE SINAL[1]	6
FIGURA 2-4 CIRCUITO DE ACIONAMENTO DAS MOTO-BOMBAS[1]	6
FIGURA 2-5 CURVAS DE CALIBRAÇÃO DAS MOTO-BOMBAS[1]	7
FIGURA 4-1 SISTEMA DE CONTROLE DE NÍVEL DE LÍQUIDOS	8
FIGURA 4-2 ESQUEMÁTICO DO SISTEMA DE CONTROLE DE NÍVEL DE LÍQUIDOS[1]	9
FIGURA 4-3- CONFIGURAÇÃO COM DOIS SISTEMAS INDEPENDENTES[1]	10
FIGURA 4-4 ARQUITETURA DE CONTROLE DE NÍVEL DE LÍQUIDOS[1]	11
FIGURA 4-5 SENSOR DE NÍVEL DE LÍQUIDOS, BASEADO EM [1]	12
FIGURA 4-6 CURVAS DE CALIBRAÇÃO DOS SENSORES DE NÍVEL DE LÍQUIDOS	13
FIGURA 4-7 DIAGRAMA ESQUEMÁTICO DO CIRCUITO DE ACIONAMENTO DAS MOTO-BOMBAS	14
FIGURA 4-8 CURVA DE CALIBRAÇÃO DAS MOTO-BOMBAS	16
FIGURA 5-1 SUPERVISÓRIO DO SISTEMA	19
FIGURA 5-2 COMUNICAÇÃO DO SUPERVISÓRIO COM O SERVIDOR	20
FIGURA 5-3 ARQUITETURA DO PADRÃO MVC	22
FIGURA 5-4 INTERFACE DE COMUNICAÇÃO ENTRE O CONTROLLER E O MODEL	24
FIGURA 5-5 EXEMPLO DE TEMPLATE TILE	26
FIGURA 5-6 ARQUITETURA <i>WEB</i> DO LEARN	27
FIGURA 5-7 ARQUITETURA <i>WEB</i> DO LEARN	29
FIGURA 5-8 COMUNICAÇÃO ENTRE SERVIDOR E CONTROLE DE PROCESSO	31
FIGURA 6-1 MODELO CONCEITUAL DO LEARN	34
FIGURA 6-2 MODELO DE ENTIDADE RELACIONAL	35
FIGURA 7-1 DIAGRAMA DE BLOCOS DE CONTROLE NO ESPAÇO DE ESTADOS[1]	41
FIGURA 7-2 DIAGRAMA DE BLOCOS DO CONTROLADOR	50
FIGURA 7-3 DIAGRAMA DE BLOCOS DO CONTROLADOR COM INTEGRADOR	51
FIGURA 8-1 NÚMERO DE EXECUÇÕES POR ALUNO	53
FIGURA 8-2 PERCENTUAL DE USO POR GRUPO	54

1. Introdução

O interesse por aplicações distribuídas na Internet aumentou consideravelmente nos últimos anos. Desses, uma alternativa barata e de qualidade são os laboratórios remotos com sistemas reais manipuláveis através da Internet.

O uso de laboratórios remotos em disciplinas curriculares deve ser feito de maneira a se controlar e identificar todos os usuários e seus respectivos experimentos. Isso permite uma avaliação mais próxima da que se conseguiria em um laboratório presencial. Nessa situação, um sistema gerenciador de banco de dados é a opção viável e necessária em projetos como o do Laboratório de Ensino de Automação Remoto da Universidade de Brasília, LEARn – UnB.

O Sistema de Controle de Nível de Líquidos do LEARn possui três tanques acoplados e pode ser modelado de diversas maneiras. Utilizando técnicas de Controle no Espaço de Estados o sistema pode ser modelado como sistema de uma ou várias entradas e saídas. O modelo conveniente utilizando controle no espaço de estados para aplicação do LEARn é como um sistema de segunda ordem com uma entrada e duas variáveis de estados, complementando aplicações na disciplina de Controle Dinâmico.

1.1. Objetivo do Trabalho

Este trabalho visa ao desenvolvimento de uma infra-estrutura para que o Laboratório de Ensino de Automação Remoto da UnB – LEARn possa disponibilizar experimentos na Internet de maneira rápida e segura. A infra-estrutura desenvolvida conta com um Sistema Gerenciador de Banco de Dados, um Sistema Supervisório independente de tipo de controle, arquitetura de servidor WEB portátil e robusta.

Além de prover um ambiente propício para disponibilização de experimentos remotos, este trabalho tem por finalidade a implementação de um controlador no Espaço de Estado para o Sistema de Controle de Nível de Líquidos do LEARn.

1.2. Organização do Trabalho

O Capítulo 2 apresenta o ponto onde se encontrava o Sistema de Controle de Nível de Líquidos antes deste trabalho. O servidor de páginas antigo era o IIS da Microsoft, substituído pelo Apache TomCat[9], contêiner JSP/Servlet de código livre e gratuito. A placa de controle foi alterada para propiciar uma melhor utilização dos atuadores. O Sistema de Supervisório era dependente do controle PID do processo dificultando o uso de outros controladores.

O Capítulo 3 trata de sistemas de experimentação remota, vantagens e motivações para tal tipo de implementação. Os sistemas remotos são sistemas reais controlados a distância e monitorados por supervisórios.

O Capítulo 4 mostra a planta de nível de líquidos. A eletrônica de controle dos atuadores e de sensoriamento utilizada. Os sensores estão projetados para fornecer uma máxima resolução de uso da placa conversora analógico digital.

O Capítulo 5 apresenta a organização da arquitetura para implementação do LEARN. O principal objetivo é criar uma arquitetura robusta que possa receber novos experimentos e diversificar a aplicação do laboratório.

O Capítulo 6 aborda o Sistema Gerenciador de Base de Dados utilizado no LEARN - SGBD FireBird[10] - e o Modelo de Entidade Relacional[10], MER, desenvolvido.

O Capítulo 6 trata o Controle no Espaço de Estados do Sistema de Controle de Nível de Líquidos, que pode ser implementado utilizando sistema SISO¹ ou MIMO².

O Capítulo 8 apresenta resultados de execuções

¹ Sistema com uma Entrada e uma Saída

² Sistema com Mais de uma Entrada e Mais de uma Saída

2. Situação anterior do Sistema de Controle de Nível de Líquidos

O Sistema de Nível de Líquidos do LEARN foi desenvolvido com um supervisor para controle PID, ver Figura 2-1, dependente do controlador e que dificultava a implementação de novas técnicas de controle como Avanço Atraso e Espaço de Estados.

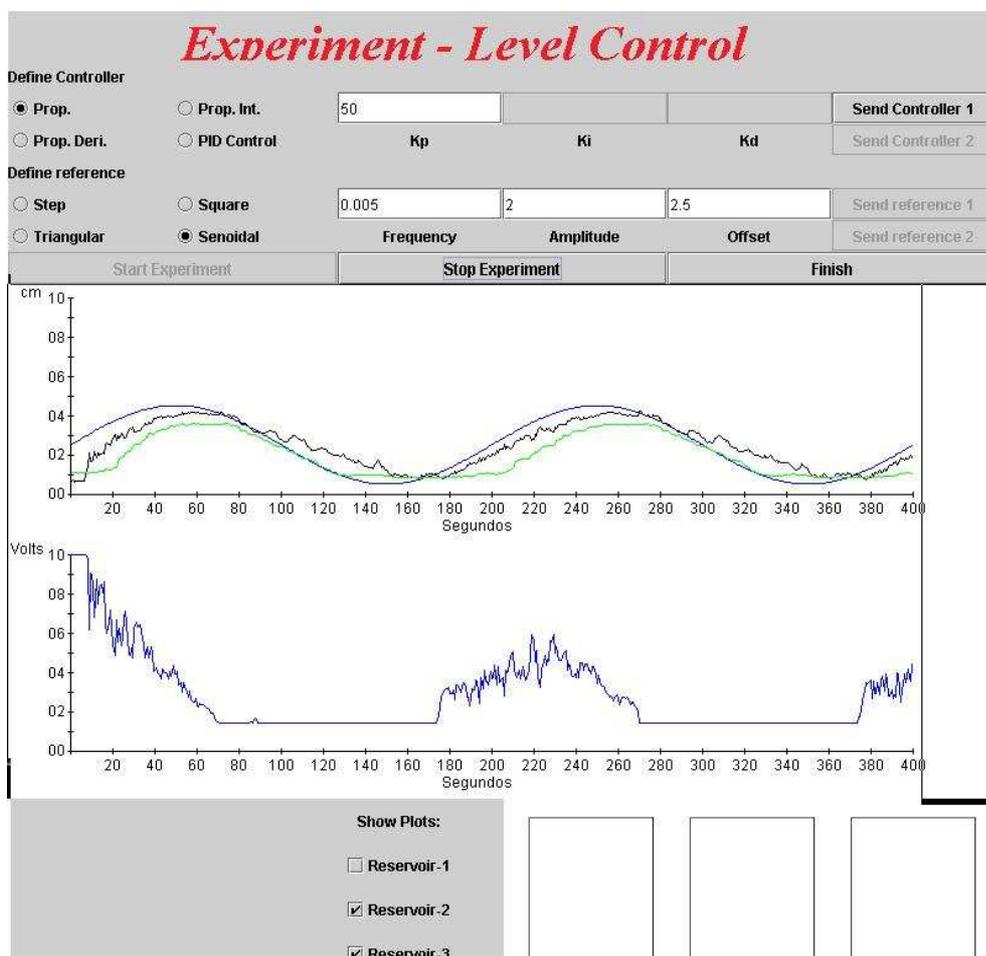


Figura 2-1 Supervisor Anterior, Limitado ao Controle PID[30]

O acesso ao experimento era livre e irrestrito, existia apenas uma interface que solicitava do usuário nome, e-mail e país, sem realizar qualquer verificação das

informações. Para um maior controle do laboratório era imprescindível o desenvolvimento de uma infra-estrutura que utilize Banco de Dados.

2.1. Arquitetura do Laboratório

A arquitetura do Servidor é dividida em duas partes: um servidor WEB, Microsoft Internet Information Service, IIS[30][28][31], e um programa Java, que envia e recebe informações do Applet³ Supervisor, como mostra a Figura 2-2.

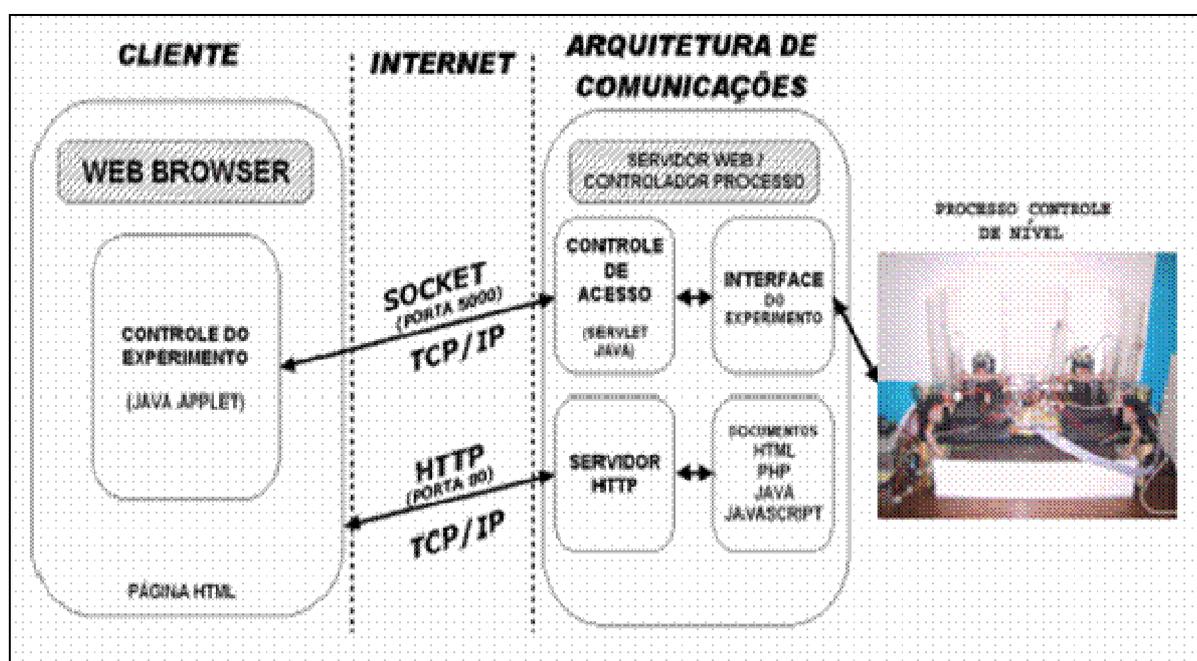


Figura 2-2 Arquitetura WEB do Laboratório[31]

A comunicação entre o servidor e o computador de controle do processo é implementada via RAM Driver⁴[29] no Servidor, o que permite o compartilhamento desta área. Esta forma de comunicação não sofreu alterações neste trabalho.

³ Programa Java executável em navegadores WEB.

⁴ Mapeamento de uma parte da memória RAM como se fosse uma área do disco rígido.

A calibração do conjunto circuito de acionamento moto-bombas está mostrada na Figura 2-5, onde nota-se que a vazão máxima atingida é de $50 \frac{cm^3}{s}$, isto se deve a baixa potência fornecida pelo circuito de acionamento.

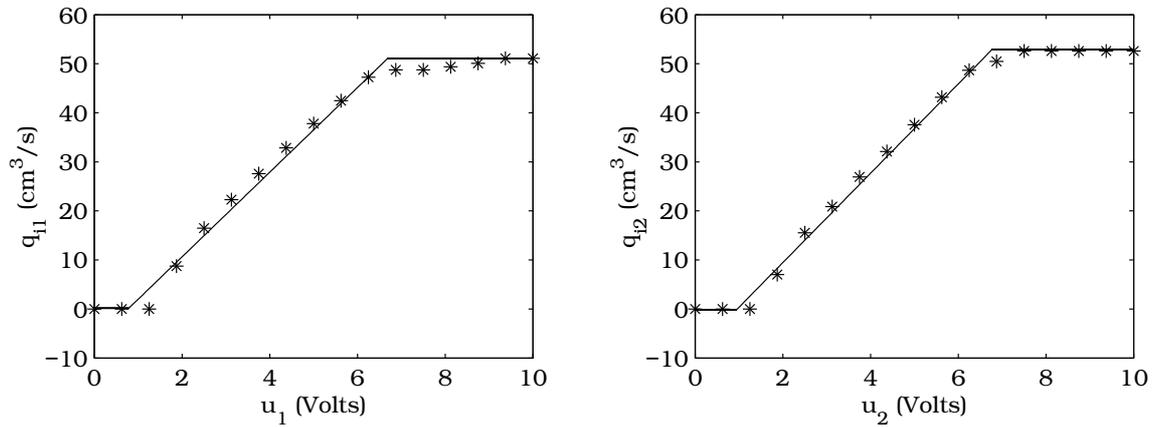


Figura 2-5 Curvas de Calibração das Moto-Bombas[1]

<C:\Documents and Settings\alfredo\Meus documentos\Sistemas Remotos.doc>

3. Sistema de Tanques Acoplados

3.1. Introdução

O Sistema de Controle de Nível de Líquidos - Figura 3-1 - do Laboratório de Ensino de Remoto de Automação da Universidade de Brasília - LEARN[25] vem sendo usado na disciplina de Controle Dinâmica dos cursos de Engenharia Elétrica e Mecatrônica, desde o 1º semestre de 2003.

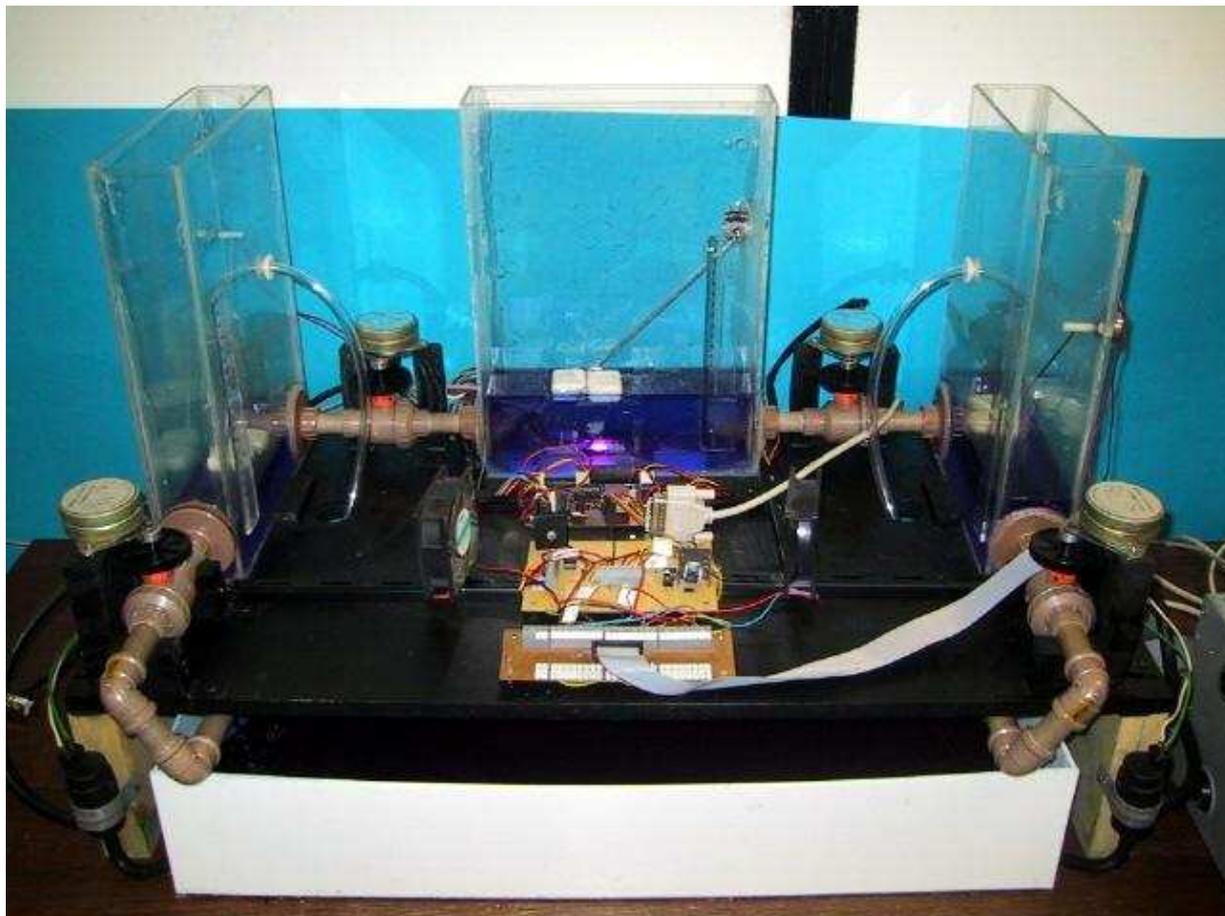


Figura 3-1 Sistema de Controle de Nível de Líquidos

O Sistema de Controle de Nível de Líquidos, da Figura 3-1, é composto por três

tanques acoplados e duas bombas. As especificações técnicas do sistema de tanques acoplados bem como a descrição do funcionamento de seus componentes estão detalhados em capítulos posteriores.

As variáveis do sistema e sua nomenclatura estão descritas na Figura 3-2. Pode-se observar que o Tanque 1 possui alimentação de líquidos através da Bomba 1 e está acoplado ao Tanque 3 por uma válvula regulável. O Tanque 2 possui alimentação de líquidos através da Bomba 2 e está acoplado ao Tanque 3 por uma válvula regulável. A saída de líquido para o reservatório ocorre através das válvulas de saída dos tanques 1 e 2.

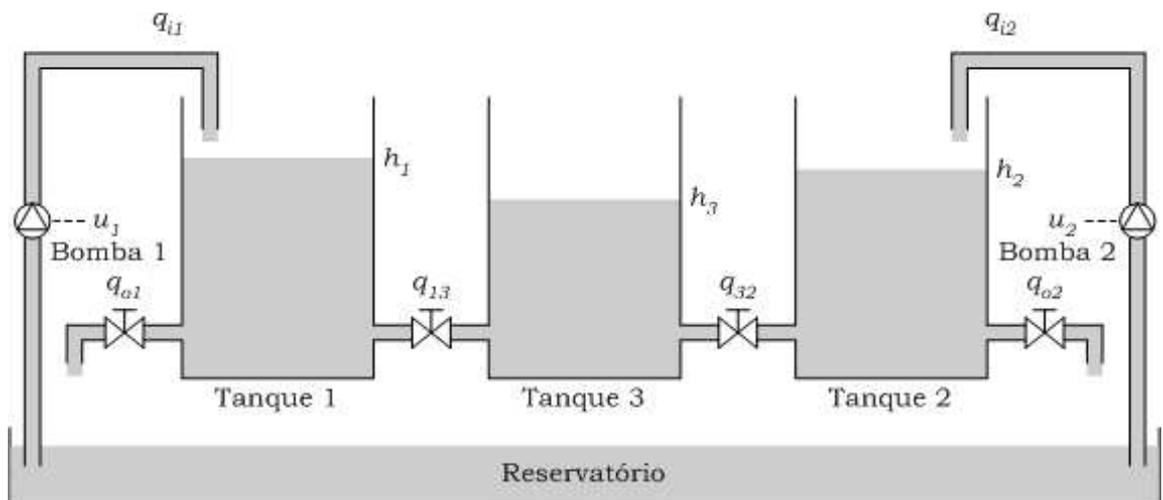


Figura 3-2 Esquemático do Sistema de Controle de Nível de Líquidos[1]

A altura do Tanque 1 é utilizada como variável de saída para um sistema de primeira ordem, a variável de entrada deste sistema é a vazão da Bomba 1, q_{i1} . A altura do Tanque 3 é utilizada como variável de saída para um sistema de segunda ordem, a variável de entrada deste sistema é a vazão da Bomba 2, q_{i2} . Esta configuração ocorre quando a válvula que liga o Tanque 1 ao Tanque 3 encontra-se fechada Figura 3-3 .

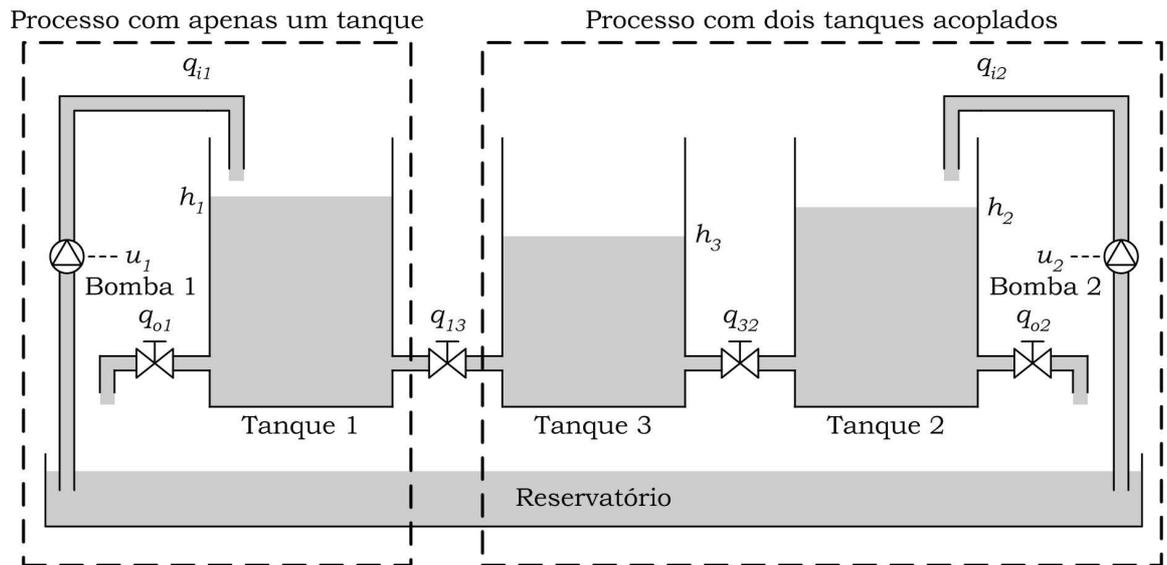


Figura 3-3- Configuração com dois sistemas independentes[1]

Na configuração MIMO, o Sistema de Controle de Nível de Líquidos terá como variáveis de estados as alturas dos tanques, duas delas variáveis de saída e as duas de entrada, as vazões das bombas, q_{i1} e q_{i2} .

3.2. Arquitetura da Planta de Nível de Líquidos

O controle da planta de nível de líquidos é realizado por um computador Intel Pentium 90Mhz. A placa AD/DA utilizada para comunicação com a planta possui 12bits de resolução e 16 entradas analógicas, das quais três estão em uso, e duas saídas digitais, ambas em uso para controle das moto-bombas. A Figura 3-4 mostra a arquitetura completa do controle e da planta de nível de líquidos. Para maiores detalhes sobre a planta veja os itens seguintes.

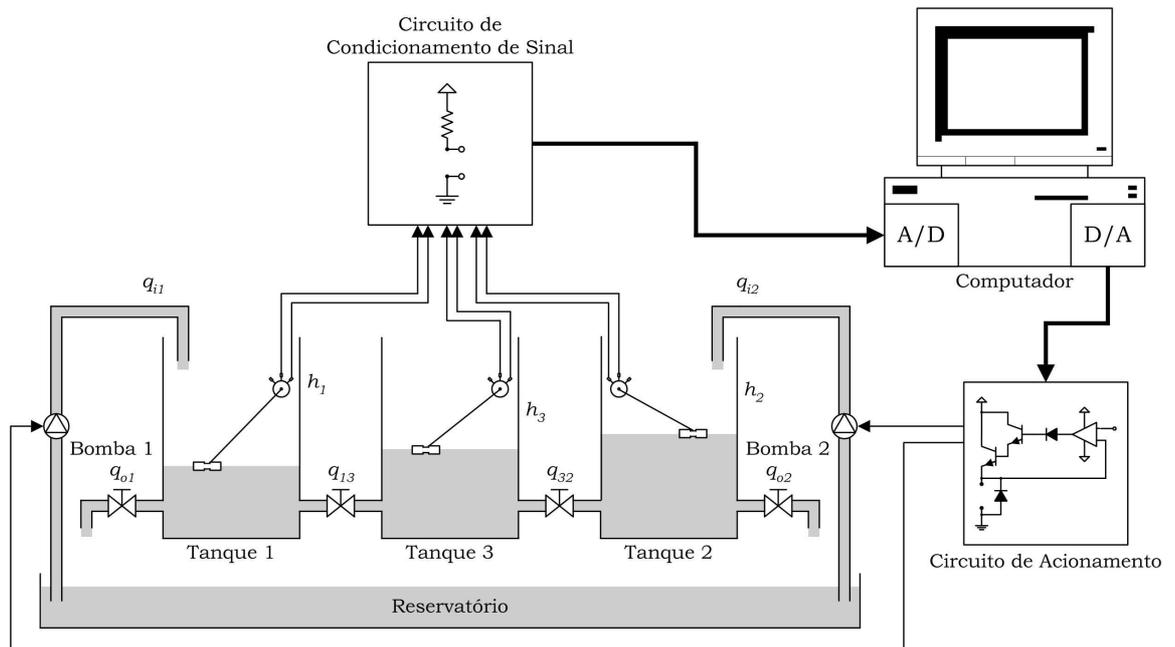


Figura 3-4 Arquitetura de Controle de Nível de Líquidos[1]

As entradas analógicas são utilizadas para receber os sinais do sistema de condicionamento de sinal, ver 3.3.

As saídas analógicas são utilizadas para o envio de sinais de controle para a placa de acionamento das moto-bombas.

3.3. Modelo dos Sensores e Atuadores

3.3.1. Sensores de Nível

Os sensores instalados nos três tanques foram construídos com potenciômetros lineares de valor nominal de $4.7k\Omega$. No cursor de cada potenciômetro está acoplado uma haste metálica com uma bóia na extremidade, como mostra a Figura 3-5.

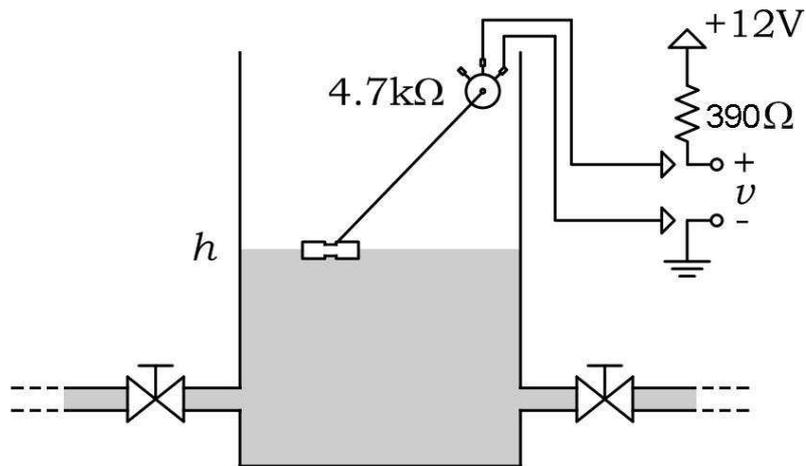


Figura 3-5 Sensor de Nível de Líquidos, baseado em [1]

Os potenciômetros estão ligados em série com um resistor de 390Ω , numa configuração de divisor de tensão, alimentado por $+12V$ [1]. A tensão sobre cada potenciômetro (sensor) está diretamente relacionada à altura de nível de água no tanque. Quando o nível de água no tanque atinge 16cm, o modelo matemático indicou o resistor de 390Ω como o ideal. A tensão será de 10V, tensão máxima da placa de aquisição de dados, maximizando a resolução do sistema.

O sucesso do controle é determinado pela certeza dos sensores, sendo assim, para cada Tanque da Figura 3-2 foi levantado um modelo que relaciona a altura do nível de líquidos - $h(\text{cm})$ - com a tensão - $v(\text{volts})$ - de cada conjunto sensor/circuito de condicionamento. A Figura 3-6 mostra as curvas de calibração para todos os conjuntos sensor/circuito de condicionamento de sinal.

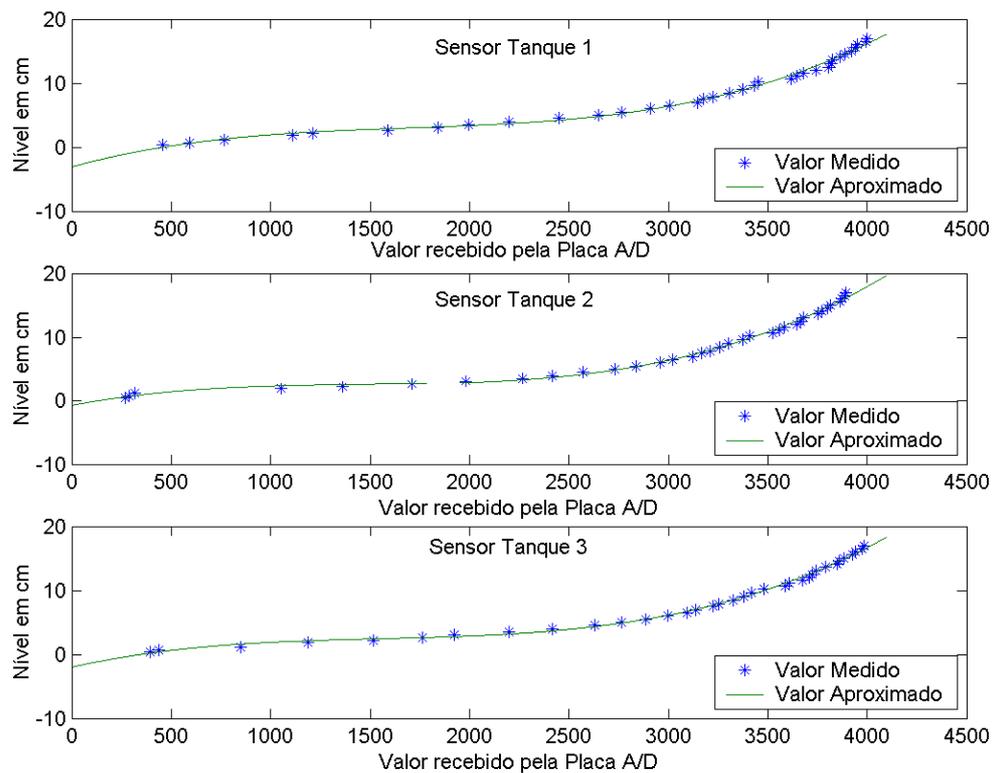


Figura 3-6 Curvas de Calibração dos Sensores de Nível de Líquidos

A calibração foi realizada para cada conjunto sensor/circuito de condicionamento de sinal, produzindo três equações, uma para cada sensor: Relação entre Valor de Digital e Altura do Sensor (1); Relação entre Valor de Digital e Altura do Sensor 2 (2); e Relação entre Valor de Digital e Altura do Sensor 3 (3).

$$h_1(v) = 8,5974 \cdot 10^{-10} v^3 - 4,3733 \cdot 10^{-6} v^2 + 8,5383 \cdot 10^{-3} v - 3,0927 \quad (1)$$

$$h_2(v) = 8,5974 \cdot 10^{-10} v^3 - 4,3733 \cdot 10^{-5} v^2 + 8,5383 \cdot 10^{-3} v - 3,0927 \quad (2)$$

$$h_3(v) = 8,5974 \cdot 10^{-10} v^3 - 4,3733 \cdot 10^{-5} v^2 + 8,5383 \cdot 10^{-3} v - 3,0927 \quad (3)$$

3.3.2. Moto-Bombas

Os dois atuadores do sistema são moto-bombas que funcionam com tensões entre 0 e 10V. O acionamento das moto-bombas é realizado através de um circuito de potência

que é comandado pelas saídas digital-analógicas da placa de aquisição de dados. A Figura 3-7 mostra o diagrama esquemático do circuito entre a placa de aquisição e as moto-bombas.

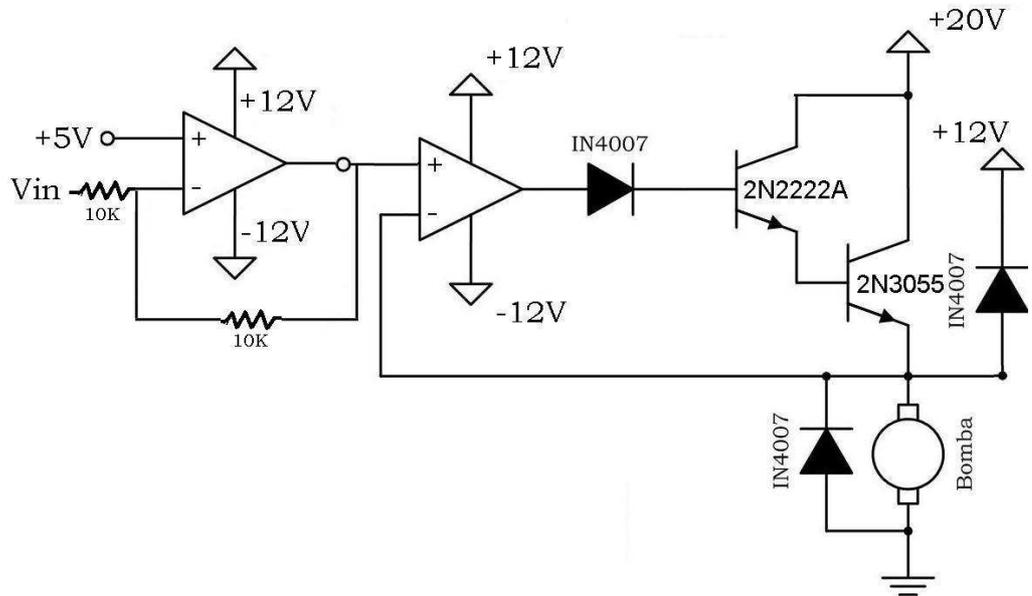


Figura 3-7 Diagrama esquemático do circuito de acionamento das moto-bombas

O transistor de potência 2N3055 foi usado para aumentar a potência dos atuadores, já que suporta corrente da ordem de 15A. A corrente máxima nos atuadores é da ordem de 3,5A em 10V. Como o transistor 2N3055 necessita de uma corrente de base na ordem de 300mA quando em potência máxima, foi acrescentado o transistor 2N2222A, que fornece corrente de emissor superior a este valor. Com os dois transistores foi realizado um par Darlington.

Antes do circuito de acionamentos dos atuadores propriamente dito foi incluído um circuito para realizar uma inversão de tensão em torno de 5V. Esta inversão era necessária porque a placa de aquisição tem por saída padrão o valor máximo de tensão quando a entrada esta em aberto. Sendo assim, o circuito ativaria as bombas em caso de falha no programa de controle ou no período anterior a sua execução. O algoritmo de controle envia o valor invertido para a placa de acionamento, logo o valor digital enviado será dado pela equação (4), relação entre valor digital desejado e enviado. A tensão no atuador em função

do valor digital enviado pelo algoritmo de controle é descrita pela equação (5) relação entre tensão no atuador e valor digital no algoritmo de controle

$$V_x = 4095 - V_d \quad (4)$$

$$V_g = 10 - \frac{10(4095 - V_d)}{4095} \quad (5)$$

O objetivo da calibração é encontrar uma relação entre vazão desejada e a saída digital que deve ser enviada pelo canal D/A. Para calibrar os conjuntos moto-bomba e circuito de acionamento – moto-bomba 1 e moto-bomba 2 - a tensão de entrada no circuito foi variada de 2 a 10V e a vazão calculada em vários níveis por um programa que utiliza os dados de altura de nível proveniente dos conjuntos sensores/circuitos de acionamento do tanque 1 e tanque 3. O procedimento foi realizado com todas as válvulas fechadas, ver Figura 3-2. A análise dos resultados mostrou pequenas variações entres os dois conjuntos moto-bomba e circuito de acionamento, isso possibilitou a utilização de uma única curva de calibração para os conjuntos. A Figura 3-8 mostra a curva encontrada.

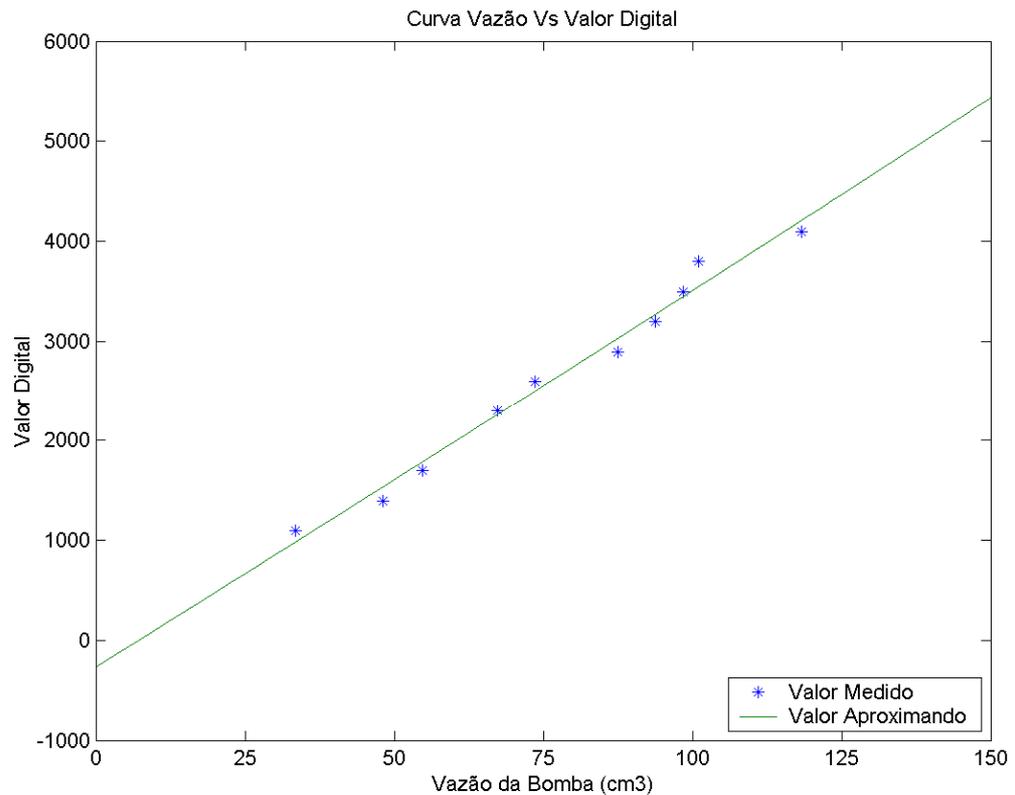


Figura 3-8 Curva de Calibração das Moto-Bombas

A análise da curva da Figura 3-8 revela que não há saturação no intervalo de trabalho, 0-10V, diferente do que ocorria em [1], ver Figura 2-5. Este fato se deve as mudanças realizadas nos circuitos de acionamento dos atuadores.

Utilizando o método de interpolação de Taylor foi determinada a equação (6) que relaciona o valor desejado de vazão e o valor que deve ser enviado pela a canal D/A para o circuito de acionamento. Com a equação (5) pode-se determinar o valor que será recebido pelo circuito de acionamento, Figura 3-7.

$$q_i = 0.0259 * u + 8.8889 \quad (6)$$

3.4. Resumo do Capítulo

Neste capítulo foi apresentada a parte física do Sistema de Controle de Nível de Líquidos. A planta e seus componentes foram resumidos em 3.1, a arquitetura da planta foi mostrada na seção 3.2, onde é utilizada a placa de aquisição de dados. O item 3.3 trabalhou os sensores e atuadores, os circuitos de condicionamento e acionamento bem como as calibrações utilizadas.

4. Arquitetura do Servidor

4.1. Introdução

Este capítulo descreve a implementação do servidor de experiência remota do LEARn.

O item 4.2 mostra o Supervisório desenvolvido com base em [30], cuja principal vantagem é a independência do tipo de controle utilizado, ver capítulo 2. A utilização do Apache TomCat[9][13], contêiner JSP/Servlet, utilizado como servidor WEB, é descrita em 4.3. Já o padrão de arquitetura WEB utilizado para disponibilização e interface foi o *Model View Controller*[7], referenciado no item 4.4.

4.2. Supervisório do Sistema

O Sistema Supervisório desenvolvido tem como objetivo mostrar os resultados de experimentos em execução no Sistema de Controle de Nível de Líquidos, Figura 4-1. A tecnologia Java Applet [32] foi a escolhida porque permite a troca de informações entre o servidor e o cliente web em tempo real. Este applet foi desenvolvido baseado em [30][31].

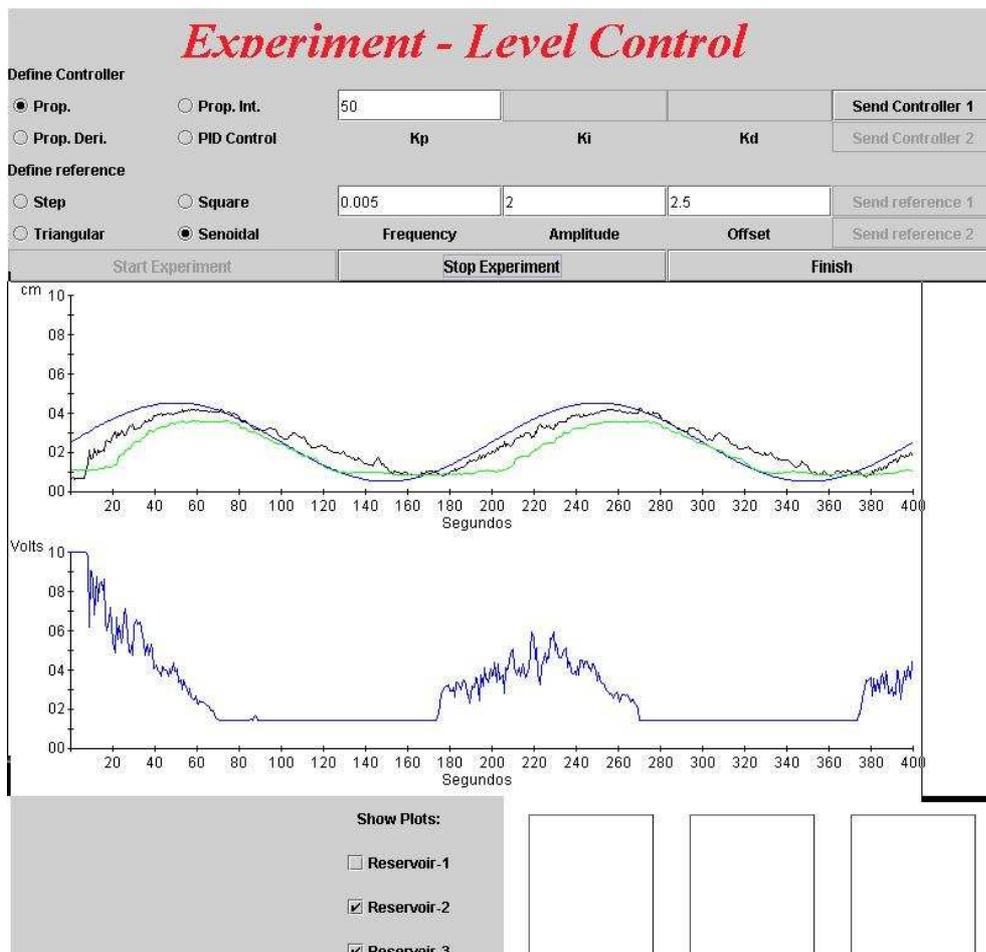


Figura 4-1 Supervisório do Sistema

No presente projeto, o supervisório inicia a comunicação, via TCP/IP, com um sistema servidor de dados que roda constantemente no computador servidor. Assim, temos uma conexão para troca de informações entre servidor e cliente. O applet envia comandos de início e término, enquanto o servidor envia resultados on-line do experimento, como mostra a Figura 4-2.

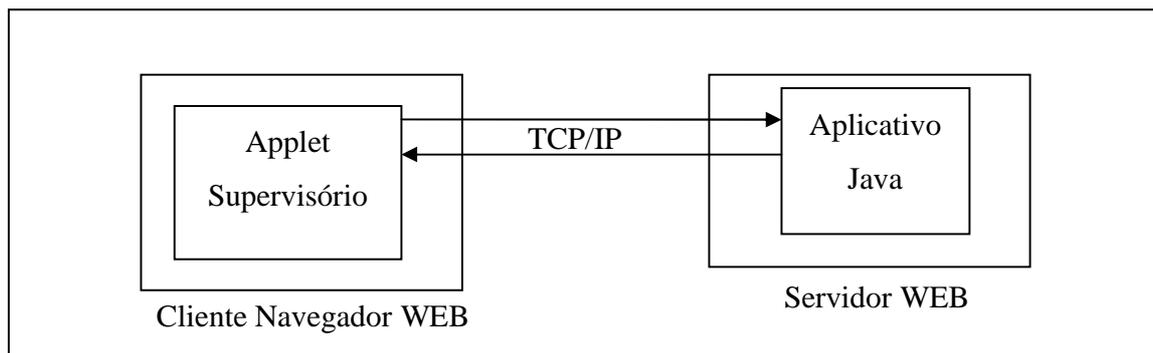


Figura 4-2 Comunicação do Supervisório com o Servidor

O envio de parâmetros de controladores e referências não é realizado pelo supervisório, para isso páginas html, ou jsp devem ser criadas para cada modelo de controle utilizado. Esse padrão torna o supervisório⁵ independente do tipo de controle utilizado. Sendo assim, para incluir novos métodos de controle basta criar uma página e uma classe de encapsulamento de parâmetros.

Com este supervisório pode-se realizar *multicasting*⁶ de informações, em que vários clientes têm acesso aos resultados de forma visual, mas apenas um pode enviar parâmetros de controle.

O supervisório gera os gráficos acessando métodos de leitura de dados de classes Java beans[32], a mesma classe é utilizada pelo servidor e pelo supervisório. Este padrão de projeto é implementado através de herança da classe Controle⁷, logo, todos os métodos de controle disponíveis devem possuir uma implementação herdada de Controle.Java.

4.3. O TomCat, Contêiner JSP/Servlet

O Contêiner JSP/Servlet TomCat[9], do grupo Jakarta Apache[13], é uma referência mundial em implementações das especificações JSP/Servlet[32]. A versão utilizada é 4.27, que implementa as especificações Servlets 2.3 e Java Server Pages 1.2 da Comunidade de

⁵O supervisório anterior era dependente do controle PID, isso dificultava a manutenção deste.

⁶ Comunicação de um servidor com vários clientes ao mesmo tempo

⁷ Arquivo de Classe Java com método pré-definidos que podem ou não ser sobrescritos.

Software Java. Características adicionais o tornam uma excelente plataforma de desenvolvimento e utilização de aplicações web e *web services*.

No LEARN, os serviços ativos do TomCat são:

- HTTP/1.1, com alguns módulos extras;
- JNDI⁸[32] e ;
- Logs .

O *Connector* HTTP/1.1, implementado por *org.apache.coyote.tomcat4.CoyoteConnector*, é responsável pela comunicação através do protocolo HTTP/1.1. Tal comunicação possui como cliente os navegadores mais modernos. Este *connector* é executado no LEARN nas portas TCP/IP 80 e 8080[26].

O serviço de JNDI foi implementado para fornecer a conexão com o banco de dados, ver capítulo 5. Tal implementação utiliza a *framework Jakarta commons*[33] para fornecer conexões com o banco de dados FireBird[10]. A conexão gerenciada é requisitada pelas classes de negócio para armazenar e recuperar informações no banco de dados. Para fornecer esta conexão às classes do LEARN, foi criada uma classe que gerencia o uso e liberação das conexões. Normalmente, somente uma conexão está ativa, porém em momentos de muitos acessos o gerenciador pode criar mais conexões.

O serviço de *logs* informa e grava em arquivos os erros e outras informações enviados pelas classes de negócio para posterior análise e verificação de funcionamento do sistema.

4.4. O padrão de Arquitetura MVC2

4.4.1. Padrões de Projeto e Arquitetura

A Engenharia de Software possibilitou a criação e o desenvolvimento de padrões de arquitetura e de padrões de projetos de softwares. Os padrões de arquitetura indicam como

⁸ *Java Naming Directory and Interface*

dividir e comunicar classes, já os padrões de projetos indicam herança e codificação de classes. O uso destes padrões visa à melhoria da performance, manutenibilidade e documentação de soluções.

4.4.2. O padrão de Arquitetura *Model View Controller 2*

O padrão *Model View Controller*[32] propõe a divisão de sistema em três grandes grupos: o grupo das regras de negócio, denominado *model*; o grupo da apresentação para o usuário, denominado *view*; e um controlador para mediar a comunicação entre estes dois grupos, ver Figura 4-3.

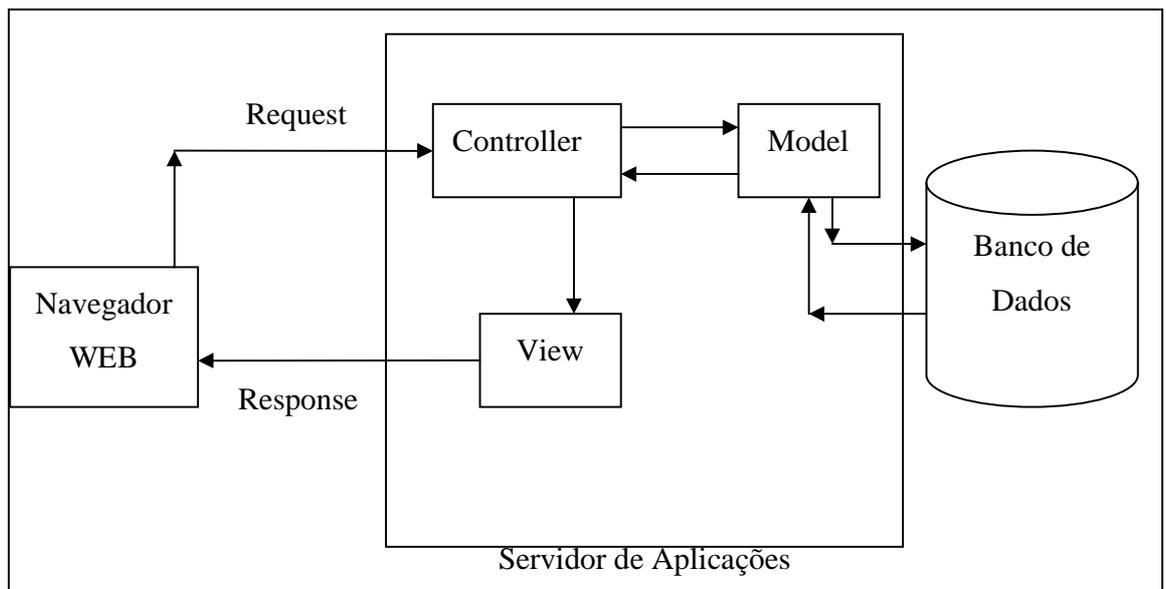


Figura 4-3 Arquitetura do Padrão MVC

O *controller* é responsável por receber requisições, no LEARN requisições do tipo http, repassá-las depois de tratadas a um *model* responsável e retornar uma apresentação, uma página html ou JSP.

A comunicação entre *view* e *model* é realizada através de classes que transportam informações entre os “dois mundos”. Com isso, a apresentação - *view* - torna-se

independente do negócio - *model*. No laboratório, estas classes são sub-classes de *ActionForm*.

4.5. A *framework* Jakarta Struts

4.5.1. A implementação do Controller e Arquitetura Struts

A *framework* Jakarta Struts implementa o *Controller* do padrão MVC. A configuração é realizada através de arquivos xml, em que estão indicados os *models* e *views*. Podendo haver vários *views* para um *model* e um *view* pode ser utilizado por mais de um *model*. Esta configuração cria vários objetos da classe *ActionMappings*, que são inicializados pelos *servlet* de controle.

A configuração abaixo do arquivo *web.xml*⁹ do contexto indica que o *servlet* pode receber requisições:

```
<servlet>
    <servlet-name>action</servlet-name>
    <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
</servlet>
```

Todas as requisições com o padrão *.web serão redirecionadas ao

```
<servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.web</url-pattern>
</servlet-mapping>
```

Um mapeamento – *ActionMapping* - de ação típico é:

⁹ Arquivo padrão da especificação JSP para configuração de aplicações web, este arquivo deve estar no diretório WEB-INF da aplicação.

```

<action attribute="usuarioForm" input=".usuario.cadastro"
name="usuarioForm" parameter="action" path="/saveUsuario" scope="request"
type="com.actions.SaveUsuarioAction">
  <forward name="cadastro" path=".usuario.cadastro" />
  <forward name="dados" path=".usuario.dados" />
  <forward name="listagem" path=".usuario.listagem" />
</action>

```

Este mapeamento indica que requisições como `saveUsuario.web` serão redirecionadas para a classe `com.actions.SaveUsuarioAction`. O redirecionamento ocorrerá através da chamada do método `execute` desta classe. Os retornos possíveis deste mapeamento são: `cadastro`, `dados` e `listagem`, além dos retornos comuns a toda a aplicação. O `usuarioForm` é um objeto Java que será passado por referência ao método `execute` e possui as propriedades e métodos da classe associada a este *formbean*¹⁰. O parâmetro de valor `request` indica que o objeto somente existirá até a próxima requisição do usuário. A Figura 4-4 mostra a interface de comunicação entre o Model e Controller.

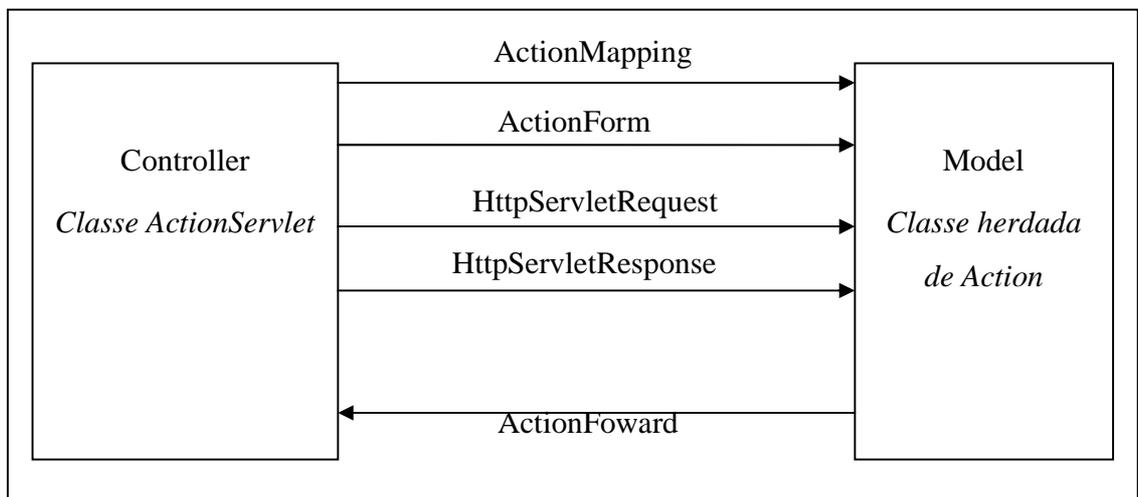


Figura 4-4 Interface de Comunicação entre o Controller e o Model

É possível ainda definir outras propriedades para cada mapeamento de ações, como em qual escopo¹¹ deve ser armazenado o retorno do negócio e quais *views* podem ser

¹⁰ Referência utilizada para classes que são intermediárias entre view e model do padrão MVC.

¹¹ Os escopos disponíveis na especificação JSP são página, requisição, seção e aplicação.

utilizados por este *action*, além dos *views* compartilhados. No LEARN, alguns *views* e mesmo classes de *model* estão compartilhados entre ações mapeadas.

No LEARN, existem, ainda outros módulos como validação de entrada de dados pelos usuários, centralização de exceções e *tiles*.

4.5.2. A implementação dos Views

A implementação de interface em ambientes que utilizam padrões de projetos é realizada por código html e *tags* personalizadas¹², além das JSTL¹³. Sendo assim, não é utilizado código Java com tags html, o que facilita o desenvolvimento e a manutenção do *designer*, podendo este ser realizado por pessoas que não conheçam a tecnologia Java..

No LEARN estão em uso alguns pacotes de *tags* personalizadas da Tabela 4-1.

<i>Tags</i>	Resumo
Html	Geram códigos de interface como componentes de formulários html. Principal vantagem: já incluem dados em caso de edição. Exemplo: escreve o nome de um usuário em campo de alteração.
Beans	Trabalham com objetos Java em vários escopos ¹¹ . Exemplo: escreve o nome do usuário conectado na tela.
Logic	Geram códigos dependentes de condições lógicas. Exemplo: se houver usuário válido na seção, mostre o link.
Tiles	Gerencia o retorno para o navegador web dinamicamente.
datetime	Manipulam data e hora. Exemplo: formatar uma data na língua do navegador web.

¹² *Tags* que se assemelham a códigos html, porém são classes Java reutilizáveis, gerenciadas pelo contêiner JSP/Servlet.

¹³ As *tags* da especificação JSP são referenciadas como JSTL, *JSP Standard Tag Library*, *tags* personalizadas padrões da especificação JSP/J2EE. As *tags* Java geram o código html em tempo de execução através da chamada de métodos específicos pelo contêiner TomCat[32]

ControleExecucao	Controlam e apresentam os dados de execuções de experimentos do LEARN. Exemplo: mostrar o código da execução atual.
JSTL	Especificação JSP.

Tabela 4-1 Tags Personalizadas em Uso no LEARN

4.5.3. Tiles com design baseado em Templates

O desenvolvimento de *designer* de páginas é algo que pode sofrer alterações e variações dinâmicas em sistemas web. A *framework tiles*[8] retorna ao usuário o *designer* de maneira dinâmica. O uso de *tiles* fornece resposta para os usuários da web criando, ao início da aplicação, uma fábrica de *tiles* disponíveis para utilização, configurados no arquivo.

Os *templates* são arquivos jsp que utilizam as tags *tiles* para indicar onde cada área do *designer* deve ser inserida em uma página padrão. Estes *templates* centralizam a manutenção da apresentação, veja a Figura 4-5.

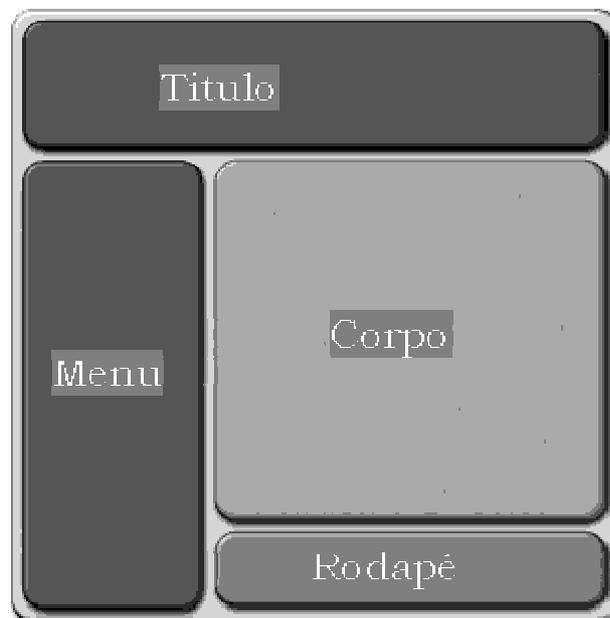


Figura 4-5 Exemplo de Template Tile

Um *tile* é um conjunto de arquivos ou propriedades que serão inseridos em um *template*, com o da Figura 4-5. O uso de *tiles* no LEARN é realizado por herança, facilitando assim o designer.

No LEARN, encontram-se em uso dois *templates* básicos: um onde existem quatro áreas, o menu, o rodapé, o título e o “corpo”, como mostra a Figura 4-5. Outro em que há somente o corpo. Este último, é utilizado em páginas que contêm o supervisor e roteiros de experimentos. No laboratório, um dos *tiles* básicos é formado pela área de título, menu e rodapé, os demais são herdados deste e somente incluem outro arquivo como corpo da página.

Os arquivos de uso exclusivo em *tiles* do LEARN estão em áreas do disco onde não é possível acesso pela Internet, aumentando assim a segurança do laboratório. Com o uso de *tiles* a arquitetura do servidor *web* do laboratório é alterada para a forma apresentada na Figura 4-6.

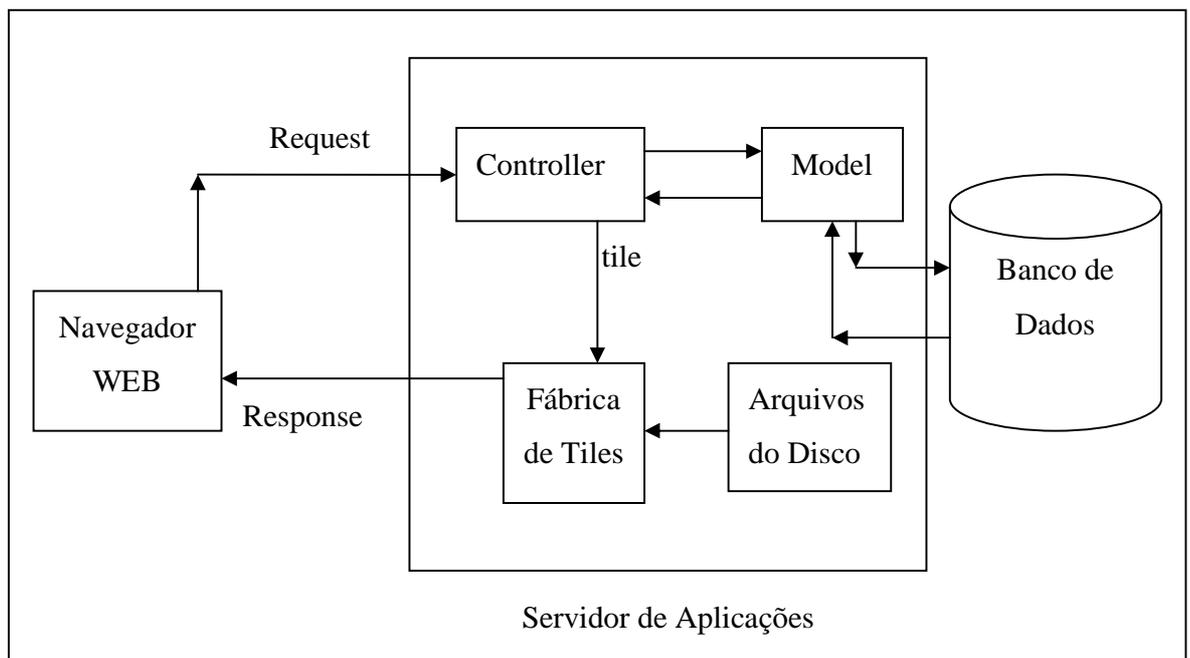


Figura 4-6 Arquitetura *web* do LEARN

4.5.4. A Implementação dos *Models*

A camada *model*, também conhecida como camada de negócio, da arquitetura *Model View Controller* encontra-se no LEARN dividida em duas partes: uma que referencia os estados do sistema; e outra que gerencia ações executadas sob tais estados.

Na subcamada de ações¹⁴ ocorre a comunicação com o *controller*, recebendo as requisições, tratando com uma ou mais subcamadas de estados¹⁵¹⁶ e, dependendo do retorno, informa um *ActionForward* para o controller. Esta subcamada é responsável por indicar o que fazer, para responder a determinada requisição de um usuário do LEARN.

A subcamada de estados é responsável pelo armazenamento e recuperação de informações requisitadas ou enviadas pelo usuário. Esta subcamada conecta-se com o banco de dados do LEARN, gravando e retornando informações deste. A subcamada de estados responde a pergunta como fazer. A Figura 4-7 mostra a camada *model* utilizada no laboratório.

¹⁴ Normalmente referenciada na literatura como *Action*

¹⁵ Implementadas no LEARN através de Java *Beans*, classes Java que são padronizadas segundo padrões da Comunidade Java.

¹⁶ Referenciada na literatura como *bussines-logic bean*.

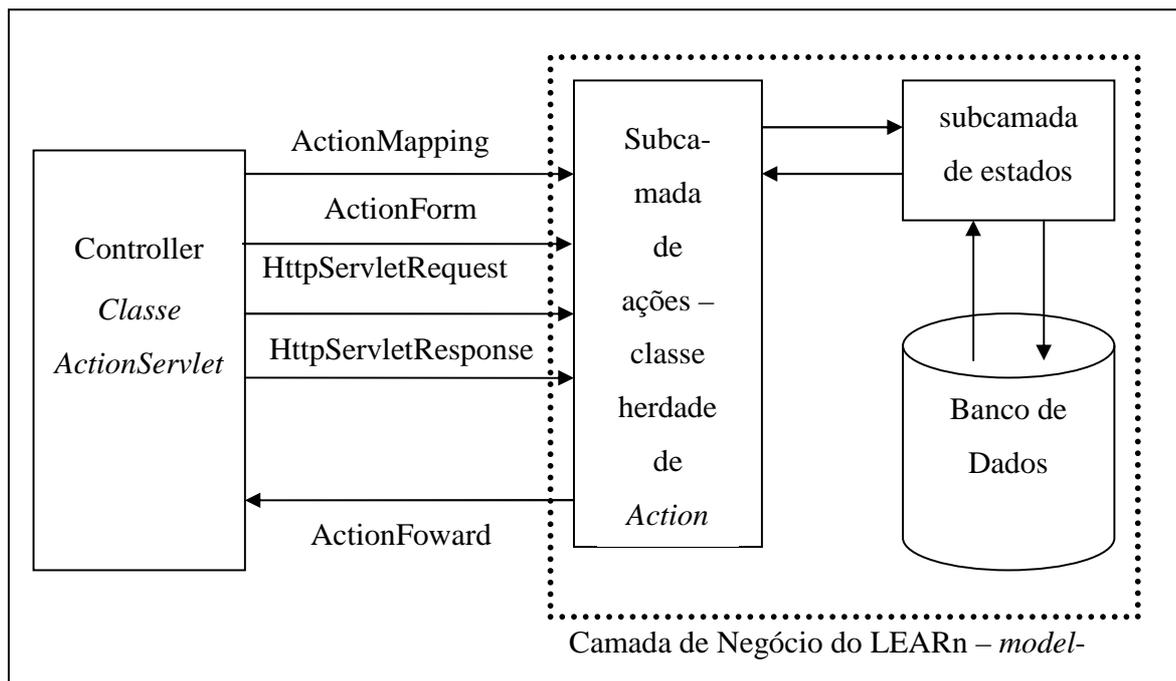


Figura 4-7 Arquitetura web do LEARN

4.5.5. Internacionalização, I18N¹⁷¹⁸[32]

O padrão de internacionalização define como um sistema deve ser desenvolvido para que sua disponibilização ocorra em várias línguas e modelos de representações diferentes. A arquitetura utilizada no LEARN segue esse padrão, criando um ambiente internacionalizável.

No LEARN, todos os *links* e pequenas descrições estão em arquivos de recursos, que podem ser traduzidos para outras línguas de maneira fácil e rápida, bastando gerar um novo arquivo com sufixo do país e língua¹⁹. Arquivo cujo nome deve possuir traço baixo, as iniciais da língua e, opcionalmente, pode-se incluir as iniciais do país, exemplo `_pt_Br`.

¹⁷ Padrão de Projeto especificado pela SUN Microsystems.

¹⁸ A sigla I18N faz referência às 18 letras existentes entre o I inicial e o N final da palavra *Internationalization*.

¹⁹ Sufixos padronizados pela Comunidade Java.

A internacionalização de arquivos de textos de grande extensão é realizada com a *framework tiles*. Para isso, basta incluir um arquivo no mesmo diretório, com o nome em formato de arquivo de propriedade Java.

A utilização de arquivos de recurso não gera demora perceptiva na resposta enviada ao usuário, pois as informações são armazenadas em memória no contêiner. No LEARn, as *tags* do grupo *beans*, ver Tabela 4-1, utilizam esse padrão para apresentar as mensagens e os *labels* do sistema de maneira internacionalizada.

4.5.6. Vantagens e Desvantagens da Arquitetura do LEARn

O padrão *Model View Controller* é uma referência mundial em arquiteturas de desenvolvimento, tanto *web* quanto cliente-servidor.

O uso de *threads* melhora a performance do sistema mantendo componentes em memória e reutilizando-os. Isto se deve ao conjunto J2EE, Apache TomCat, Struts e FireBird.

Esta arquitetura resolve o problema de manutenção e dificuldade de controle de fluxo em aplicações, utilizando para isso arquivos de configuração.

A distinção entre interface e negócio aumenta a qualidade dos produtos, possibilitando o trabalho de maneira independente em tais áreas e gerando uma especialização dos envolvidos no projeto.

A manutenção é facilitada, com pouca ou nenhuma interferência em outras partes do sistema, aumentando assim sua robustez.

Em ambientes como a *web*, internacionalização é um fator muito importante, quase necessário.

O tempo de desenvolvimento da infra-estrutura inicial é alto, se comparado com outras arquiteturas. Já a ampliação e o desenvolvimento de novos módulos são facilitados e mais rápidos.

A curva de aprendizagem de arquiteturas como a do LEARN, padrão MVC, não é tão rápida quanto a de arquiteturas simples, sem camadas definidas e com linguagens estruturadas. Exemplo ASP e PHP.

A geração de muitos arquivos, classes Java, é vista por muitos desenvolvedores como um obstáculo à utilização de arquiteturas como a implementada no LEARN. Porém, o reduzido tempo de manutenção e a facilidade de identificação de problemas, compensam essas supostas desvantagens.

4.6. A Integração do Servidor com o Controle de Processos

A comunicação entre o computador Servidor e o de Controle de Processos é realizada através de *RAM Driver*, compartilhado no Servidor. Esta comunicação não sofreu alterações neste trabalho, para mais informações sobre esta comunicação veja[29].

Através de arquivos de controle e de resultados os computadores trocam informações como mostra a Figura 4-8.

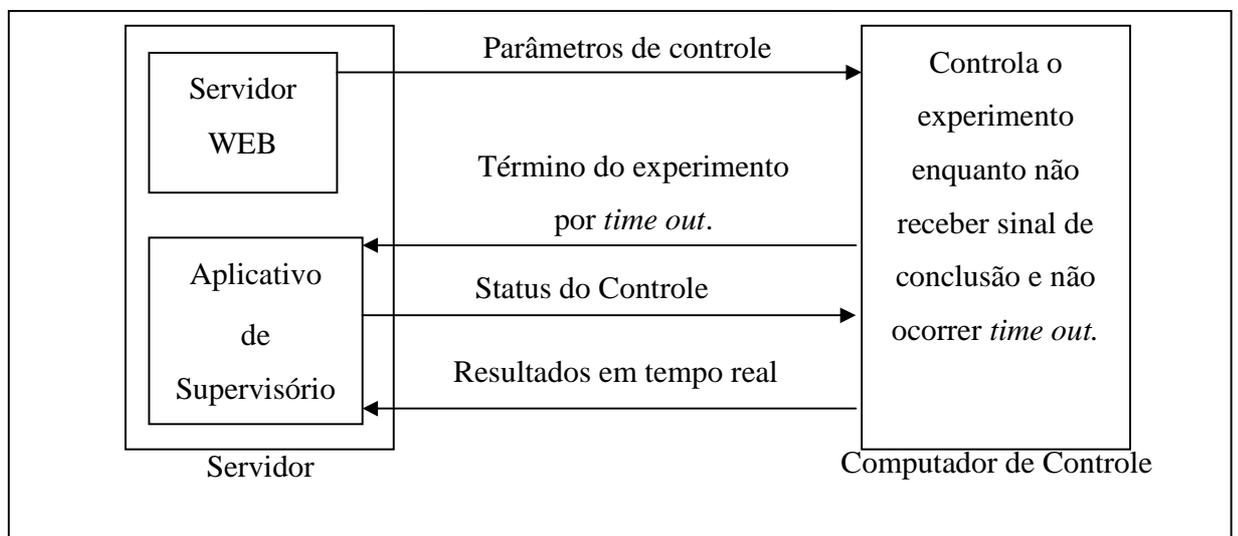


Figura 4-8 Comunicação entre Servidor e Controle de Processo

A arquitetura completa de comunicação do Sistema de Nível de Líquidos do LEARN é mostrada no diagrama de blocos da Figura 4-9.

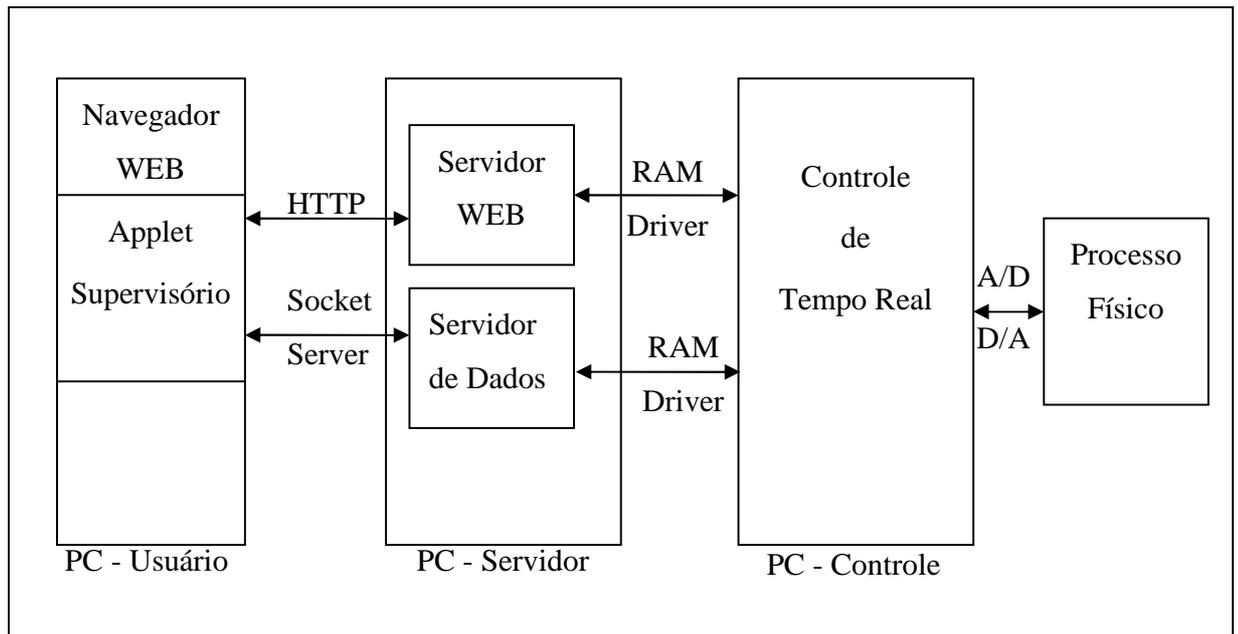


Figura 4-9 Arquitetura do Sistema de Nível de Líquidos do LEARN

4.7. Resumo do Capítulo

Este capítulo mostrou como o ambiente do LEARN está preparado para receber novos experimentos com uma arquitetura robusta, barata e baseada em ambientes testados mundialmente.

O recurso de internacionalização é algo muito importante, pois facilita a apresentação em congressos internacionais e utilização por pessoas de diferentes nacionalidades.

Todo o aparato tecnológico mostrado neste capítulo é de código fonte aberto e gratuito, mantido ou pela comunidade Java ou pelo grupo Apache, principalmente os sub-projetos do grupo Apache Jakarta que fornecem uma orientação para implementação de aplicações web, bem como infra-estrutura para disponibilização de tais aplicações com o TomCat.

5. O Banco de Dados do LEARn

5.1. Introdução

A utilização de um Sistema Gerenciador de Base de Dados - SGDB,[15], possibilita a recuperação e análise de informações de maneira rápida e fácil, através de interfaces *web* ou ferramentas de administração para confecção de relatórios.

O SGDB utilizado no LEARn é o FireBird[10][11][12], que possui características de bancos de dados robustos e confiáveis como:

- Integridade Referencial Declarativa²⁰;
- Procedimentos Armazenados²¹, Visões²² e Gatilhos²³ de Banco;
- Forte uso de índices;
- Compatibilidade com as especificações SQL²⁴, ANSI-92 e 99;
- Linguagem de programação robusta e simples, semelhante ao Pascal;
- Gratuito e de código aberto;
- Possui versões com ou sem uso de *threads*.
- Possui uma grande comunidade mundial de suporte e manutenção, com fundação e membros votantes para decisões de implementações.
- Disponível em várias plataformas: Windows, Linux, UNIX, MAC-OS e outros;

²⁰ Na literatura também é referenciada como *DRI* ou *Constraint*.

²¹ Normalmente referenciados em inglês como *Stored Procedures*.

²² Normalmente referenciados em inglês como *Views*.

²³ Normalmente referenciados em inglês como *Triggers*

²⁴ *Structured Query Language*

5.2. O Modelo de Entidade Relacional - MER

O principal objetivo da base de dados do LEARN é controlar o uso dos experimentos pelos alunos e fornecer um conjunto de acesso diferenciado por grupos de usuários, como professores, alunos e membros do laboratório, ver Figura 5-1 e Figura 5-2.

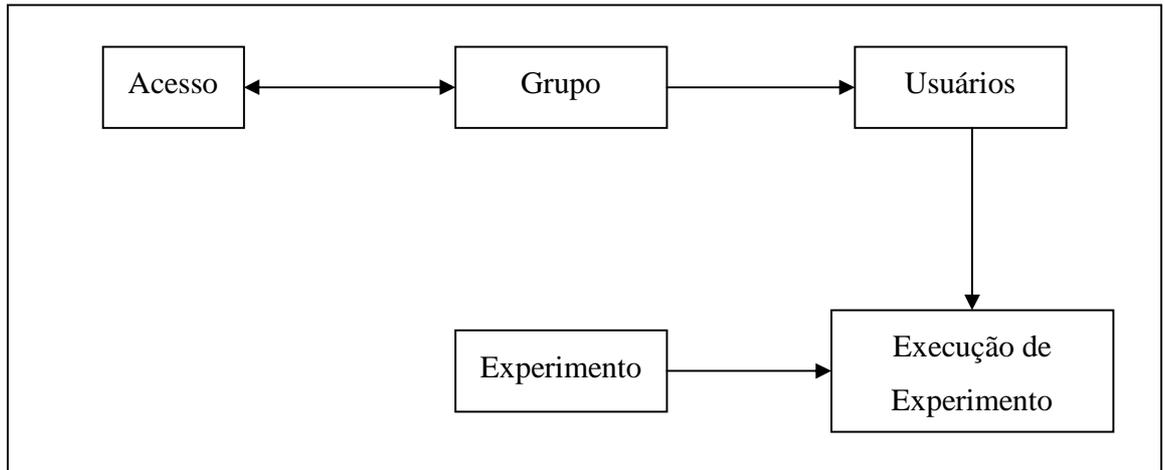


Figura 5-1 Modelo Conceitual do LEARN

A partir do Modelo Conceitual do LEARN, Figura 5-1, obteve-se o Modelo de Entidade Relacional da Figura 5-2. O relacionamento entre Grupo e Acesso é do tipo n para n, já que um grupo pode ter mais de um Acesso e um acesso pode ser relacionado a mais de um grupo. Sendo assim, surgirá uma tabela entre Acesso e Grupo no Modelo Lógico do Banco de Dados.

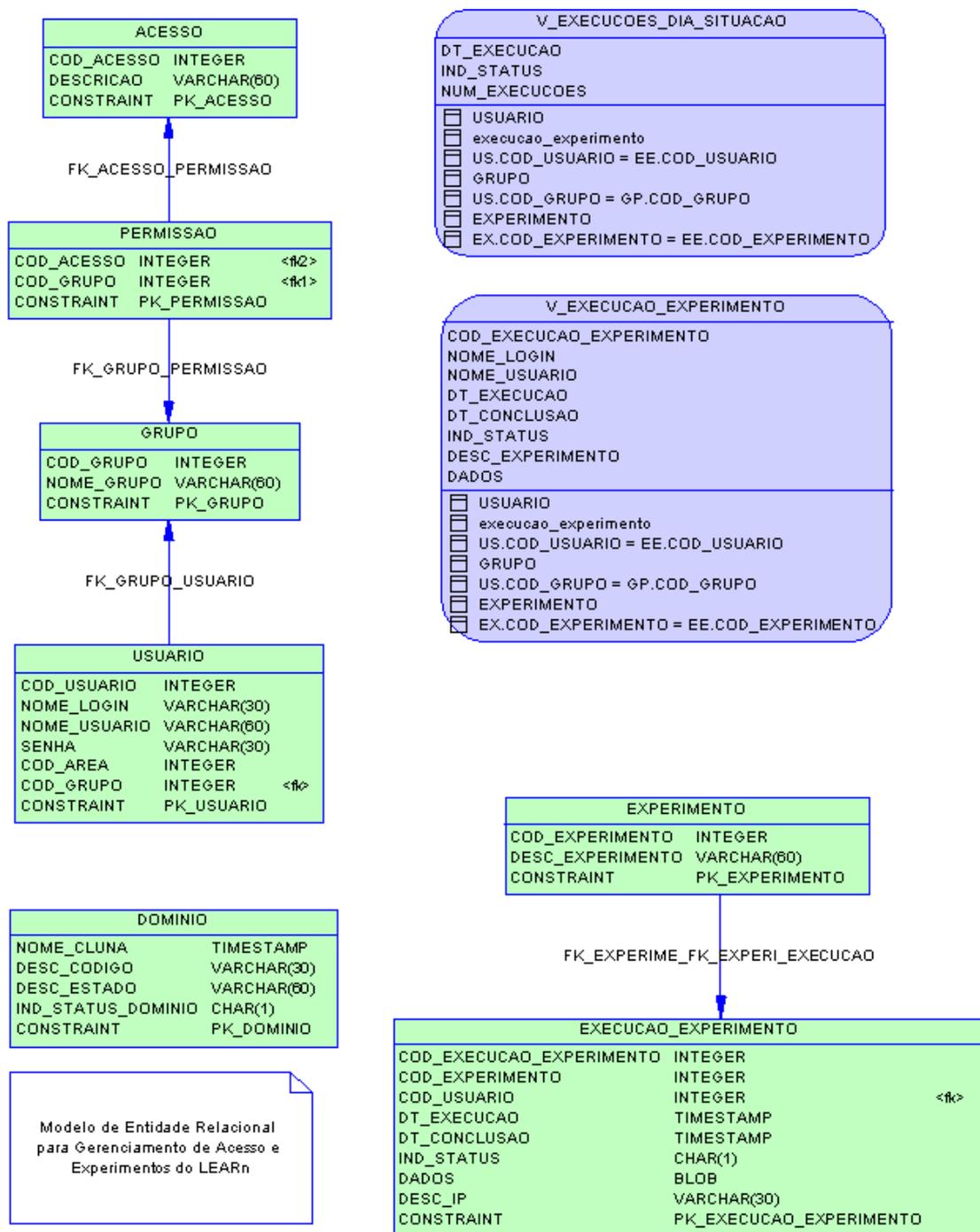


Figura 5-2 Modelo de Entidade Relacional

O Banco de Dados do LEARN possui dois grupos de ações principais: um para gerenciamento de usuários e acessos; e outro para gerenciamento de execuções de experimentos.

5.2.1. As tabelas do Banco de Dados do LEARN

Os bancos de dados armazenam informações em forma de linhas em tabelas com estrutura definidas. As tabelas do banco de dados do LEARN estão mostradas no modelo de entidade relacional da Figura 5-2. O MER está na terceira forma normal e possui as chaves primárias como códigos internos gerados de maneira auto-incremental.

A tabela de acessos possui o cadastro de áreas do *site* que não podem ser acessadas por todos os usuários cadastrados. Armazena as seguintes informações:

- *COD_ACESSO* – Código interno gerado pelo sistema para integridade referencial, é a chave primária da tabela.
- *DESCRICAÇÃO* – Descrição da área de acesso restrito.

A tabela grupos possui o cadastro dos grupos de usuários disponíveis no LEARN, como professores, alunos, pesquisadores, alunos de controle dinâmico, convidados e outros. Armazena as seguintes informações:

- *COD_GRUPO* – Código do grupo, gerado automaticamente de forma incremental.
- *NOME_GRUPO* – Nome do grupo, utilizado para indicar o significado de forma que as pessoas compreendam.

A tabela permissões indica quais grupos possuem acesso a determinadas áreas do *site*. Por exemplo, o grupo de professores pode gerenciar o cadastro de usuários. Armazena as seguintes informações:

- *COD_ACESSO* – Código interno do acesso relacionado à permissão, é uma chave estrangeira que vem da tabela acesso.
- *COD_GRUPO* - Código interno do grupo relacionado à permissão, é uma chave estrangeira que vem da tabela grupo.

A tabela de usuários possui o cadastro de todos os usuários do LEARN, cada usuário pertence a apenas um grupo. Armazena as seguintes informações:

- *COD_USUARIO* - Código do grupo, gerado automaticamente de forma incremental.

- *NOME_LOGIN* – Nome de login que deve ser utilizado para acessar o LEARN pelo usuário.
- *NOME_USUARIO* – Nome completo do usuário.
- *SENHA* – Senha que será utilizada pelo usuário para acessar o LEARN.
- *COD_GRUPO* - Código interno do grupo ao qual o usuário pertence, é uma chave estrangeira que vem da tabela grupo.
- *DESC_EMAIL* – E-mail do usuário.
- *IND_STATUS* – Situação do usuário, o cadastro das possíveis situações está na tabela de DOMINIO. Os valores possíveis estão descritos na Tabela 5-1:

Valor	Significado
1	Ativo
2	Temporariamente Inativo
3	Inativo ²⁵

Tabela 5-1 Conjunto de Valores de Situação de um Usuário

A tabela de experimento possui o cadastro dos experimentos disponíveis no LEARN, exemplo Nível de Líquidos. Armazena as seguintes informações:

- *COD_EXPERIMENTO* – Código do grupo, gerado automaticamente de forma incremental.
- *DESC_EXPERIMENTO* – Nome do Experimento do LEARN.

A tabela de execução guarda os dados de cada execução de um experimento por um usuário, sendo que um usuário pode executar um mesmo experimento quantas vezes desejar. Armazena as seguintes informações:

²⁵ Os usuários não serão excluídos fisicamente do banco de dados, mas somente sua situação será alterada para Inativa e assim este ficará sem acesso a áreas restritas do site. A não exclusão é justificada para que se possa saber o que o usuário executou no LEARN, caso de ex-alunos de Controle Dinâmico.

- *COD_EXPERIMENTO* – Código interno do experimento executado, é uma chave estrangeira que vem da tabela de cadastro de experimentos.
- *COD_USUARIO* – Código interno do usuário responsável pela execução, é uma chave estrangeira que vem da tabela de cadastro de usuários.
- *DT_EXECUCAO* – Data e hora do início da execução.
- *DT_CONCLUSAO* – Data e hora de conclusão da execução.
- *DADOS* – Dados da execução com informações suficientes para refazer e analisar a execução posteriormente.
- *DESC_IP* – Endereço IP de onde o usuário executou o experimento.
- *IND_STATUS* - Situação da execução, o cadastro das possíveis situações está na tabela de DOMINIO. Os valores possíveis estão descritos na Tabela 5-2.

Valor	Significado
1	Concluído, o usuário pegou os dados
2	Concluído, o usuário abandonou os dados
NULL	Experimento não teve início, ou os dados são de um período em que os dados não eram armazenados ²⁶

Tabela 5-2 Conjunto de Valores de Situação de uma Execução

A tabela domínio armazena os valores possíveis para colunas com conjunto limitado e de valores conhecidos, como *ind_status* do cadastro de usuário. Com essa tabela única para armazenar estes domínios, o entendimento do modelo é facilitado, não é necessária a criação de várias tabelas para cada conjunto limitado e conhecido de valores.

²⁶ Os resultados de experimentos passaram a serem armazenados de forma completa no início do 2º semestre de 2003.

5.2.2. As Visões²⁷ do Banco de Dados do LEARN

As visões de banco de dados proporcionam uma maneira mais clara de analisar informações do que as tabelas físicas. Os dados das tabelas físicas são tratados e agrupados de maneira lógica proporcionando uma nova visão.

A visão *V_EXECUCAO_EXPERIMENTO* trata os dados de execuções e utiliza informações como o nome completo do usuário para mostrar as execuções de determinado experimento.

A visão *V_EXECUCOES_DIA_SITUACAO* agrupa e conta o número de execuções por dia e por situação. A partir desta, é possível com uma simples consulta SQL determinar quantas execuções foram bem sucedidas e quantas tentativas ocorreram em um determinado dia.

5.3. A expansão projetada para o Banco de Dados

Os dados dos experimentos realizadas no 2º semestre 2003 revelam que o banco de dados LEARN cresce, como mostra a equação (7), em função do número de alunos e da quantidade de experimentos executados no semestre. Estes dados utilizam as seguintes formulações:

- O arquivo de resultados dos experimentos é da ordem de 150Kb.
- O número mínimo de execução para cada experimento é de uma execução.

$$\text{tam} = n \cdot \text{exp} \cdot 20000 \text{ [bytes/semestre]} \quad (7),$$

onde:

- tam – Tamanho de incremento do arquivo de banco de dados em *bytes*.
- n – Número de alunos que utilizaram o LEARN.
- exp – Número de experimentos realizado no semestre.

²⁷ Normalmente referenciadas pelo termo em inglês Views.

O tamanho do arquivo pode sofrer alterações caso o número de experimentos de testes seja significativo em relação ao número de execuções por aluno.

Com base na projeção de crescimento, o banco de dados está configurado para armazenar informações em 10 arquivos de disco, aumentando assim a velocidade de acesso. Este número deve ser incrementado com o crescimento do banco. Adicionando-se novo arquivo a cada ano, pode-se garantir a rapidez no acesso aos dados.

O limite de tamanho do banco de dados é imposto pelo sistema de arquivos em uso, NTFS²⁸. Mudando para LINUX, com sistema de arquivos iNodo, recomenda-se a inclusão de um novo arquivo de dados a cada semestre.

5.4. Resumo

O banco de dados do LEARN possibilita o armazenamento e a fácil recuperação das informações de execuções de experimentos pelos usuários. Fornecendo assim, uma maneira de análise e validação dos dados por um professor ou monitor. A partir dos dados, também, pode-se extrair várias informações estatísticas como: uso por dia, grupo e tentativas de acesso.

Como o banco de dados e a arquitetura do servidor, ver capítulo 4, não estão restritos ao Sistema de Controle de Nível de Líquidos facilmente pode-se incluir novos experimentos no LEARN.

O Conjunto formado pela arquitetura do capítulo 4 e o banco de dados apresentado neste capítulo utiliza softwares gratuitos e de código aberto, com alta portabilidade entre sistemas operacionais, aumentando ainda mais a qualidade do LEARN e criando um ambiente barato e de qualidade para a experimentação remota.

²⁸ *NetWork File System*

6. Controle no Espaço de Estados

6.1. Introdução

A Engenharia de Controle possui varias técnicas de controle de processos[3][4][5][6][1], como controle PID, controle de Avanço e Atraso, Controle por Linearização Exata, Controle no Espaço de Estados e outros. A implementação de Controle no Espaço de Estados é foco deste capítulo. A Figura 6-1 mostra a estrutura de controle no Espaço de Estados.

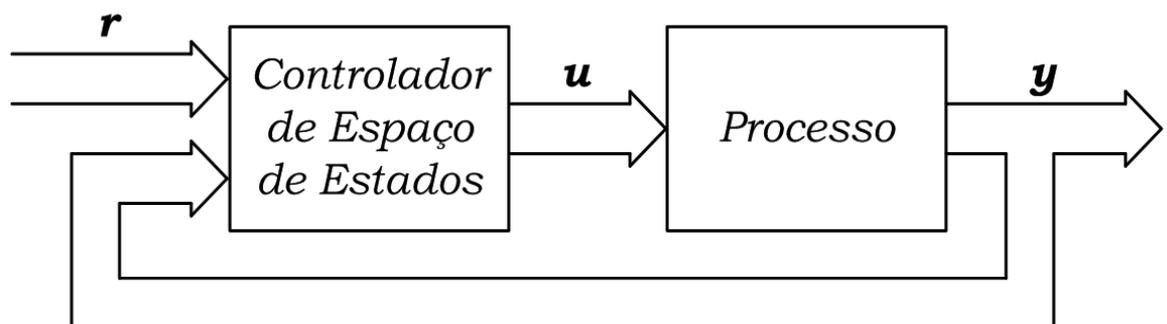


Figura 6-1 Diagrama de Blocos de Controle no Espaço de Estados[1]

O controle no Espaço de Estados embora bastante robusto e base para o Controle Ótimo ainda é menos utilizado que métodos de controle por transformada, como controle PID. Isso ocorre porque no mercado são encontrados vários kits prontos para controle PID, bastando ajustar os parâmetros do controlador.

O controle no Espaço de Estados requer um número maior de sensores para a realimentação de estados, ou o uso de estimadores de estado. Já nas técnicas de controle por transformada é necessária somente a realimentação do sinal de saída. A utilização de estados intermediários, caso do controle no Espaço de Estados, fornece uma resposta mais eficiente.

O item 6.2 fará uma análise das vantagens e desvantagens deste tipo de controle comparando com os métodos de transformadas.

O item 6.3 resume o modelo matemático do Sistema de Controle de Nível de Líquidos no Espaço de Estados.

O item 6.4 apresenta o controle sendo aplicado ao Sistema de Controle de Nível de Líquidos com controle de uma entrada e uma saída.

O item 6.5 trata o controle sendo aplicado ao Sistema de Controle de Nível de Líquidos com controle de mais de uma entrada e mais de uma saída.

O item 6.6 mostra a implementação dos controladores no Espaço de Estados.

6.2. Espaço de Estados e Métodos de Transformadas

A Tabela 6-1[3] apresenta uma comparação entre as técnicas de controle por transformadas[3][5][6][4][3] e o controle no Espaço de Estados.

Métodos por Transformadas	Controle no Espaço de Estados
Somente se aplica a sistema de uma entrada e uma saída	Aplica-se a sistema com uma ou várias entradas e uma ou várias saídas.
Em sua maioria não levam em conta estados intermediários do sistema.	Utiliza estados intermediários do sistema.
Aumentam a complexidade do sistema através da inclusão de pólos e zeros.	Não necessita incluir pólos e zeros.
Dificulta o posicionamento de pólos e zeros desejados em malhas fechadas.	Facilidade de posicionamento de pólos e zeros desejados em malhas fechadas.
São limitados a sistemas lineares invariantes no tempo.	Podem ser aplicados a sistemas não lineares e variantes no tempo.
A rejeição de entradas modificantes e interferentes é realizada após ser observadas na saída.	Pode-se projetar estimadores de entradas modificantes e interferentes de efeito contrário antes que afetem o sistema físico.
Somente necessita da realimentação da	Necessita de realimentação de todos os

saída.	estados, ou do uso de estimadores de estado.
--------	----------------------------------------------

Tabela 6-1 Comparativo entre Métodos por Transformadas e por Espaço de Estados

6.3. Modelagem Matemática no Espaço-de-Estados

O Sistema de Controle de Nível de Líquidos é um sistema que possui várias não linearidades. Algumas delas, como histerese e problemas de resolução, foram desprezadas porque, apesar de interferirem, não alteram o resultado consideravelmente.

A dinâmica dos atuadores e do sistema é muito lenta quando comparada com a dinâmica do controle, já que esta possui uma frequência entre 200-215Hz, o que permite a utilização de técnicas de controle contínuo, mesmo este sendo discreto.

As não linearidades modeladas tratam das relações entre vazões e altura de nível de líquidos. Não linearidades dos sensores e atuadores estão incluídas nas calibrações realizadas na seção 3.3.

Considerando o processo da Figura 3-2, as seguintes variáveis podem ser definidas:

- q_{i1}, q_{i2} , vazão de entrada [cm^3/s] nos tanques 1 e 2, fornecidas pelas motobombas;
- q_{13}, q_{32} , vazão de entrada [cm^3/s] entre tanques 1-3 e 3-2;
- q_{o1}, q_{o2} , vazão de saída [cm^3/s] dos reservatórios 1 e 2;
- h_1, h_2, h_3 , altura [cm] do nível de líquido nos reservatórios 1, 2 e 3.

6.3.1. Modelo Não Linear do Sistema

Aplicando o balanço de massa a cada um dos reservatórios tem-se:

$$A \frac{dh_1}{dt} = q_{i1} + \text{signal}(h_3 - h_1)q_{13} - q_{o1}, \quad (8),$$

$$A \frac{dh_2}{dt} = q_{i2} + \text{signal}(h_3 - h_2)q_{32} - q_{o2}, \quad (9),$$

$$A \frac{dh_3}{dt} = -\text{signal}(h_3 - h_1)q_{13} - \text{signal}(h_3 - h_2)q_{32} \quad (10),$$

Nas equações (8) a (10), A é a seção transversal dos reservatórios e $\text{signal}(\cdot)$ é uma função que retorna -1 , se o sinal do argumento é negativo; 0 , se for igual a zero; e 1 , se for positivo.

As vazões nos tanques são dadas por:

$$q_{o1} = k_1 \sqrt{h_1}, \quad (11),$$

$$q_{o2} = k_2 \sqrt{h_2}, \quad (12),$$

$$q_{13} = k_{13} \text{signal}(h_3 - h_1) \sqrt{|h_3 - h_1|}, \quad (13),$$

$$q_{32} = k_{32} \text{signal}(h_3 - h_2) \sqrt{|h_3 - h_2|}. \quad (14),$$

Substituindo a vazão de saída do tanque 1 - equação (11) - e a vazão entre o tanque 3 e o tanque 1 - equação(13) - na equação (8). Da mesma forma, a vazão de saída do tanque 2 - equação (12) - e a vazão entre o tanque 3 e o tanque 2 - equação (14) - na equação (9), e a vazão entre o tanque 1 e o tanque 3 - equação (13) - e a vazão entre o tanque 2 e o tanque 3 - equação (14) - na equação (10), obtem-se a seguinte representação não linear do sistema de tanques acoplados:

$$A \frac{dh_1}{dt} = q_{i1} + \text{signal}(h_3 - h_1)k_{13} \sqrt{|h_3 - h_1|} - k_1 \sqrt{h_1} \quad (15),$$

$$A \frac{dh_2}{dt} = q_{i2} + \text{signal}(h_3 - h_2)k_{32} \sqrt{|h_3 - h_2|} - k_2 \sqrt{h_2} \quad (16),$$

$$A \frac{dh_3}{dt} = -\text{signal}(h_3 - h_1)k_{13} \sqrt{|h_3 - h_1|} - \text{signal}(h_3 - h_2)k_{32} \sqrt{|h_3 - h_2|} \quad (17),$$

Considerando h_1, h_2, h_3 como variáveis de estado x_1, x_2, x_3 , o modelo não linear no Espaço de Estados será:

$$A \dot{x}_1 = q_{i1} + \text{signal}(x_3 - x_1)k_{13}\sqrt{|x_3 - x_1|} - k_1\sqrt{x_1} \quad (18),$$

$$A \dot{x}_2 = q_{i2} + \text{signal}(x_3 - x_2)k_{32}\sqrt{|x_3 - x_2|} - k_2\sqrt{x_2} \quad (19)$$

$$A \dot{x}_3 = -\text{signal}(x_3 - x_1)k_{13}\sqrt{|x_3 - x_1|} - \text{signal}(x_3 - x_2)k_{32}\sqrt{|x_3 - x_2|} \quad (20)$$

Onde q_{i1} e q_{i2} são as entradas do sistema e as variáveis de estado x_1, x_2 são as saídas do sistema, logo:

$$\begin{aligned} y_1 &= x_1 \\ y_2 &= x_2 \end{aligned} \quad (21)$$

Este modelo pode ser implementado computacionalmente. Porém, para projeto de controlador poderá ser utilizado um modelo linearizado, ver 6.3.2.

6.3.2. Modelo Linearizado no Ponto de Operação

O modelo descrito neste item será linearizado em torno do ponto de operação $(\bar{H}_1, \bar{H}_2, \bar{H}_3, \bar{Q}_{i1}, \bar{Q}_{i2})$, onde

Fazendo uma analogia com um sistema elétrico, relacionando os parâmetros do sistema hidráulico com os parâmetros “R” e “C” de um circuito elétrico[4][5]. A resistência (R) é calculada como a diferença de nível para gerar uma vazão unitária, dada por:

$$R = \frac{(\text{alteracao_no_nivel_de_liquido} - [\text{cm}])}{(\text{alteracao_na_vazão} - [\text{cm}^3/\text{s}])}$$

$$R = \frac{dH}{dQ_o} \quad (22)$$

A capacitância (C) pode ser definida como sendo a variação na quantidade de líquido necessária para causar uma variação unitária no potencial (altura no nível)[1], logo é calculada como:

$$C = \frac{\text{alteracao_no_liquido_armazenado} - [\text{cm}^3]}{(\text{alteracao_no_nivel_de_liquido} - [\text{cm}])}$$

$$C = \frac{dV}{dH} \quad (23)$$

A capacitância (C) representa a área do tanque e a resistência (R) a perda de carga que o fluido sofre até chegar ao reservatório. Este valor de resistência é também usado para determinação da influência de um tanque sobre o outro, através de seus respectivos dutos de comunicação.

Os parâmetros “R” do sistema são determinados experimentalmente, enquanto que o valor da capacitância dos tanques é determinado pela área da seção transversal do tanque.

A partir das equações (11), (12), (13) e (14) pode-se determinar que:

$$R_1 = \frac{2\sqrt{\bar{H}_1}}{k_1}, \quad (24)$$

$$R_2 = \frac{2\sqrt{\bar{H}_2}}{k_2}, \quad (25)$$

$$R_{13} = \frac{\text{signal}(h_3 - h_1) 2\sqrt{\left| \bar{H}_3 - \bar{H}_1 \right|}}{k_{13}}, \quad (26)$$

$$R_{23} = \frac{\text{signal}(h_3 - h_2) 2\sqrt{\left| \bar{H}_3 - \bar{H}_2 \right|}}{k_{23}}, \quad (27)$$

Definindo as variáveis de estados como:

$$\begin{aligned} x_1 &= h_1 - H_1, \\ x_2 &= h_2 - H_2, e, \\ x_3 &= h_3 - H_3 \end{aligned} \quad (28)$$

as entradas do sistema como:

$$\begin{aligned} u_1 &= q_{i1} - Q_{i1}, \\ u_2 &= q_{i2} - Q_{i2} \end{aligned} \quad (29)$$

e as vazões podem ser descritas como:

$$\frac{x_1 - x_3}{R_{13}} = q_{13} \quad (30)$$

$$\frac{x_2 - x_3}{R_{32}} = q_{32} \quad (31)$$

$$\frac{x_1}{R_1} = q_{o1} \quad (32)$$

$$\frac{x_2}{R_2} = q_{o2} \quad (33)$$

Substituindo as equações (24) a (33) no Modelo Não Linear no Espaço de Estados, equações (18), (19) e (20) e fazendo $C=A$, tem-se:

$$\begin{cases} C \frac{dx_1}{dt} = u_1 + \frac{x_3 - x_1}{R_{13}} - \frac{x_1}{R_1} = u_1 + \frac{x_3}{R_{13}} - \frac{x_1}{R_{13}} - \frac{x_1}{R_1} \\ C \frac{dx_2}{dt} = u_2 + \frac{x_3 - x_2}{R_{32}} - \frac{x_2}{R_2} = u_2 + \frac{x_3}{R_{32}} - \frac{x_2}{R_{32}} - \frac{x_2}{R_2} \\ C \frac{dx_3}{dt} = \frac{x_1 - x_3}{R_{13}} - \frac{x_3 - x_2}{R_{32}} = \frac{x_1}{R_{13}} - \frac{x_3}{R_{13}} - \frac{x_3}{R_{32}} + \frac{x_2}{R_{32}} \end{cases} \quad (34)$$

logo representando na forma matricial tem-se que o modelo linearizado no Espaço de Estados na forma matricial é:

$$\begin{cases} \dot{x}_1 \\ x_1 \\ \dot{x}_2 \\ x_2 \\ \dot{x}_3 \\ x_3 \end{cases} = \begin{bmatrix} -\frac{1}{C} \frac{R_1 + R_{13}}{R_1 R_{13}} & 0 & \frac{1}{CR_{13}} \\ 0 & -\frac{1}{C} \frac{R_2 + R_{32}}{R_2 R_{32}} & \frac{1}{CR_{32}} \\ -\frac{1}{CR_{13}} & -\frac{1}{CR_{32}} & -\frac{1}{C} \frac{R_{13} + R_{32}}{R_{13} R_{32}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} \frac{1}{C} & 0 \\ 0 & \frac{1}{C} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (35)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

6.4. Sistema de Controle de Nível de Líquidos SISO

O Controle no Espaço de Estados de Sistema com uma entrada e uma saída- SISO - faz parte do estudo de disciplinas curriculares de Engenharia Mecatrônica e Elétrica. O LEARN proporciona este tipo de controle visando aplicação prática dos estudos teóricos dos alunos.

A grande vantagem do controle SISO no Espaço de Estados em relação a métodos de transformada, ver Tabela 6-1, é que aquele utiliza os estados intermediários para determinar o sinal do atuador.

O Sistema de Controle de Nível de Líquidos pode ser modelado como sistema de segunda ordem com diversas configurações de entrada/saída. Considerando um sistema de segunda ordem onde a altura do tanque 3 - h_3 - é a variável de saída, obtém-se o modelo linearizado SISO de 2ª ordem - equação (36) - partir do modelo da equação matricial (35):

$$\begin{cases} \dot{x}_2 \\ \dot{x}_3 \end{cases} = \begin{bmatrix} -\frac{1}{C} \frac{R_2 + R_{32}}{R_2 R_{32}} & \frac{1}{C R_{32}} \\ -\frac{1}{C R_{32}} & -\frac{1}{C} \frac{R_{13} + R_{32}}{R_{13} R_{32}} \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1/C \\ 0 \end{bmatrix} u_2 \quad (36)$$

$$\{y = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \end{bmatrix}$$

Baseado no Modelo Não Linear no Espaço de Estados, equações (18), (19) e (20), o modelo SISO não linear de 2ª ordem, é:

$$\dot{x}_2 = q_{i2} + \text{signal}(x_3 - x_2) k_{32} \sqrt{|x_3 - x_2|} - k_2 \sqrt{x_2} \quad (37)$$

$$\dot{x}_3 = -\text{signal}(x_3 - x_1) k_{13} \sqrt{|x_3 - x_1|} - \text{signal}(x_3 - x_2) k_{32} \sqrt{|x_3 - x_2|} \quad (38)$$

Os modelos Linearizados - equação (35)- e Não Linear - equações(36) e (38)- consideram que a válvula entre o tanque 3 e o tanque 1 não está fechada. Para aplicação na disciplina de Controle Dinâmico do Departamento de Engenharia Elétrica da UnB, utiliza-se a válvula entre tais tanques fechada, sendo assim os modelos são:

$$\begin{cases} \dot{x}_2 \\ \dot{x}_3 \end{cases} = \begin{bmatrix} -\frac{1}{C} \frac{R_2 + R_{32}}{R_2 R_{32}} & \frac{1}{CR_{32}} \\ -\frac{1}{CR_{32}} & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1/C \\ 0 \end{bmatrix} u_2 \quad (39)$$

$$\{y = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \end{bmatrix}$$

e o modelo não-linear é:

$$\dot{x}_2 = q_{i2} + \text{signal}(x_3 - x_2)k_{32}\sqrt{|x_3 - x_2|} - k_2\sqrt{x_2} \quad (40)$$

$$\dot{x}_3 = -\text{signal}(x_3 - x_2)k_{32}\sqrt{|x_3 - x_2|} \quad (41)$$

6.5. Sistema de Controle de Nível de Líquidos MIMO²⁹

No Sistema de Controle de Nível de Líquidos as válvulas podem ser configuradas de maneira a obter mais de uma saída, essas saídas serão as alturas do tanque 1 e tanque 2, sendo estas também variáveis de estado. As entradas serão as vazões provenientes das moto-bombas. O nível do terceiro tanque será utilizado como variável de estado e fará parte do modelo matemático.

As equações (18), (19) e (20) e o modelo matricial (35) modelam o sistema com duas entradas e duas saídas: a primeira, de forma mais genérica, considerando não linearidades; e a segunda linearizando o sistema no ponto de operação. No sistema real as linearizações não existem, porém estas são importantes para encontrar a matriz de realimentação de estados do sistema em malha fechada, tal equação pode ser obtida pelo algoritmo de LQR, que determina o valor para menor esforço dos atuadores.

²⁹Sistema onde há mais de uma entrada e mais de uma saída, do inglês Multiple Input Multiple Output.

6.6. Implementação do Controlador no Espaço de Estados

A utilização de Controladores no Espaço de Estados permite alterar a dinâmica do sistema de forma a levar os pólos desejados para qualquer posição. Em alguns casos tais dinâmicas não podem ser estabelecidas por limitações dos atuadores, porém se consegue atingir dinâmicas melhores que utilizando técnicas de transformadas[3][4].

A lei de controle é da forma:

$$u[t] = -Kx[t] \quad (42)$$

Para usar o sistema de maneira que este siga sinais de referência, é necessária a inclusão desses sinais na lei de controle de maneira que gerem um vetor com o mesmo número de estados do sistema. A lei de controle é alterada para a equação (43), esta equação pode ser mostrada como no diagrama de blocos da Figura 6-2

$$u[t] = -Kx[t] + \bar{N}r[t] \quad (43)$$

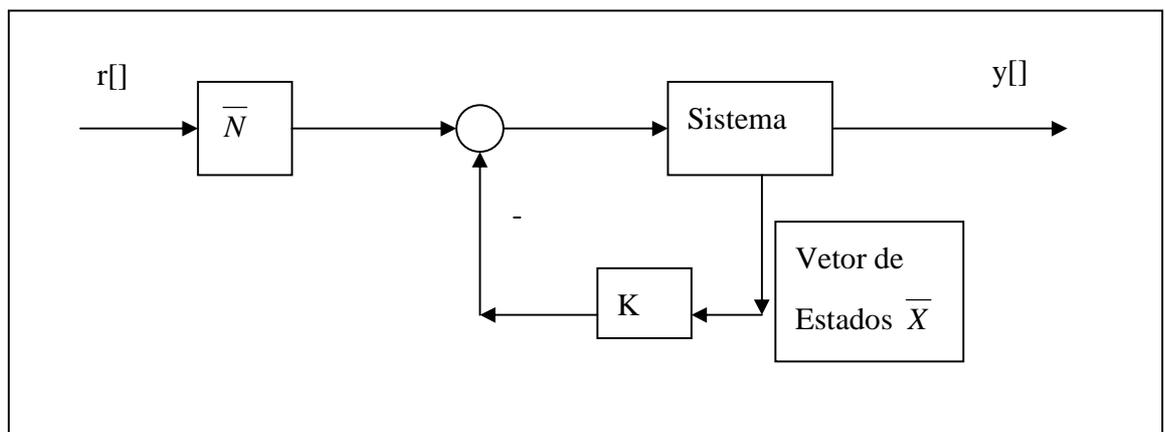


Figura 6-2 Diagrama de Blocos do Controlador

O objetivo da matriz \bar{N} é permitir que a saída siga a referência, eliminando erros de regime permanente. Porém, em alguns casos, tal fato é difícil de se obter em sistemas variantes no tempo que não possuem integrador no seu modelo, como o de nível de líquidos. Nestes casos, a inclusão do ganho integral melhora a resposta. Veja o diagrama de blocos da Figura 6-3.

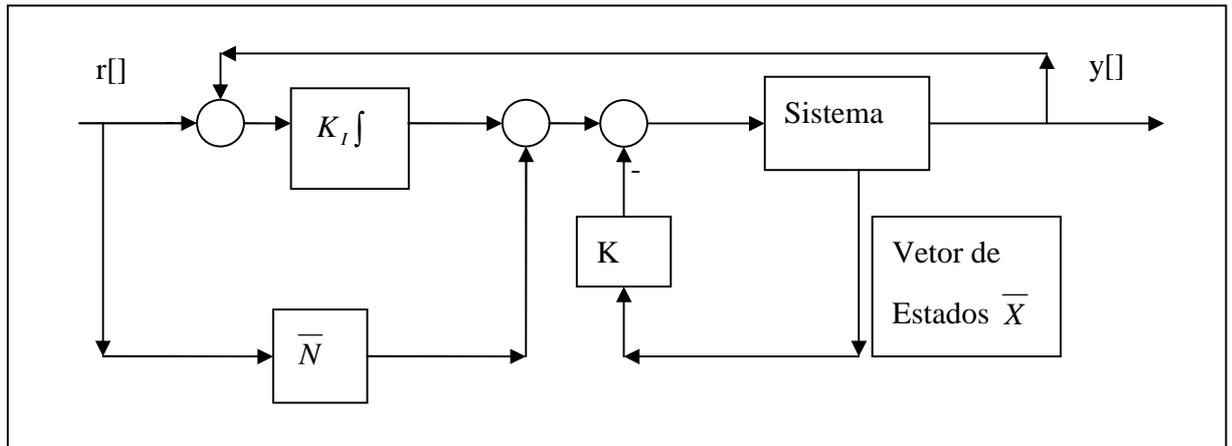


Figura 6-3 Diagrama de Blocos do Controlador com Integrador

A utilização do integrador adiciona tantas variáveis de estados, bem como de pólos ao sistema, quanto for o número de variáveis de saídas. Com isso, a equação de controle mantém a forma da Lei de Controle Com Entrada da Referência (43). Porém, o vetor de estados é ampliado e o número de linhas da matriz de realimentação de estados também. Com o canal integral do controlador, as rejeições de falha de modelo e até mesmo de projeto da matriz de realimentação de estados serão minimizadas.

6.7. Resumo do Capítulo

O controle de Sistema no Espaço de Estados é mais abrangente e versátil que projetos por métodos de transformadas. No caso do experimento nível de líquidos, já existia sensores que fornecem as variáveis de estados facilitando assim a implementação do controle.

O controle de SISO do Sistema de Controle de Nível de Líquidos é bastante propício para uso no LEARN na disciplina de Controle Dinâmico. Já o controle MIMO pode ser utilizado para aplicações de técnicas de Controle Não Linear e Controle Ótimo.

7. Utilização do Laboratório de Controle Dinâmico

O LEARN disponibilizou durante o segundo semestre de 2003 experimentos para todas as turmas de laboratório de Controle de Dinâmico da Universidade de Brasília.

Os dados deste capítulo são relativos ao período de dois experimentos do LEARN, 29 de outubro a 13 de dezembro de 2003. Somente as execuções bem sucedidas foram considerados para análise, não sendo levado em conta as tentativas de acesso. As informações utilizadas para gerar os gráficos foram extraídas das visões do banco de dados do LEARN, ver item 5.2.2.

7.1. Execuções por Aluno

Com as informações do banco de dados do laboratório, ver capítulo 5, foi gerado o gráfico da Figura 7-1 de número de execuções por aluno. O número de alunos está localizado no eixo Y do gráfico, e o número de execuções realizadas, no eixo X.

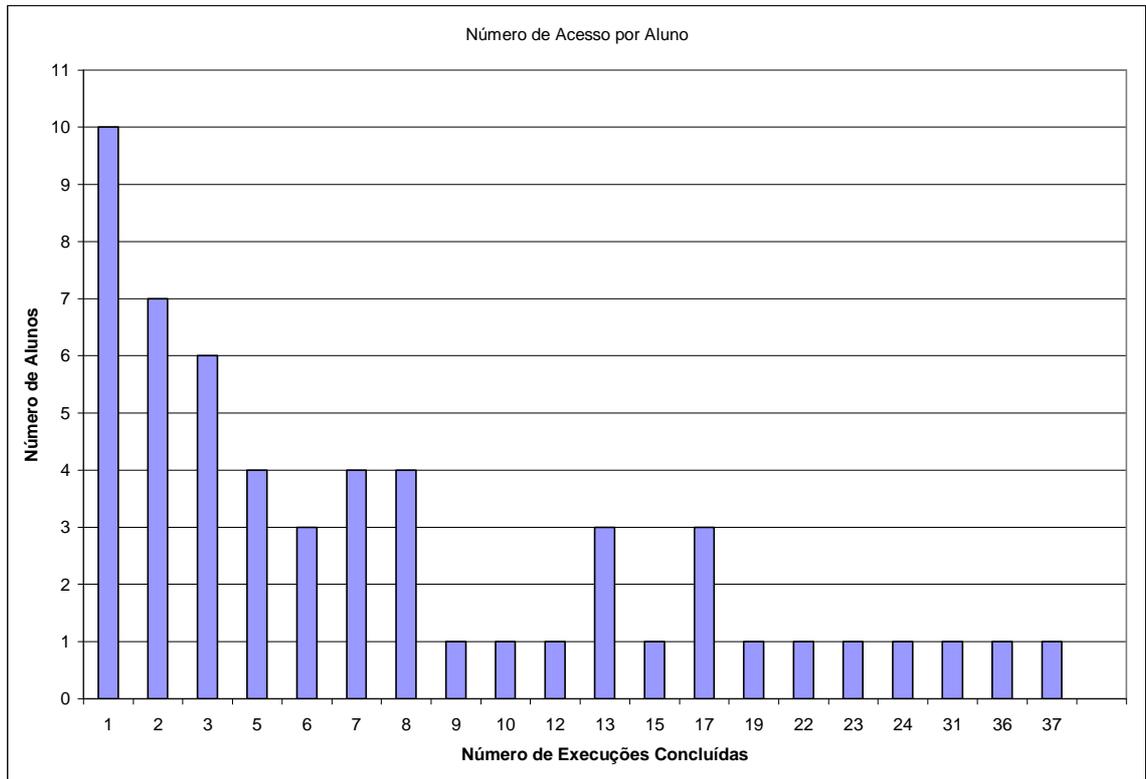


Figura 7-1 Número de Execuções por Aluno

A análise dos dados da Figura 7-1 revelam que:

- 10 alunos realizaram apenas uma execução, não sendo suficiente para produzir os dois relatórios do período analisado.
- Metades dos alunos executaram o Sistema de Nível de Líquidos menos de cinco vezes.
- Alguns alunos realizam experimentos em excesso, sete alunos realizaram mais de 20 vezes os experimentos.

Uma alternativa para evitar essas distorções no número de execuções por alunos, como as mostradas na Figura 7-1, seria a exigência de um pré-relatório para execução dos experimentos remotos.

7.2. Execuções por Grupo

Analisando as execuções por grupos obteve-se o gráfico da Figura 7-2.. Este gráfico mostra de forma percentual o uso do LEARN por alunos, professores e membros do laboratório.

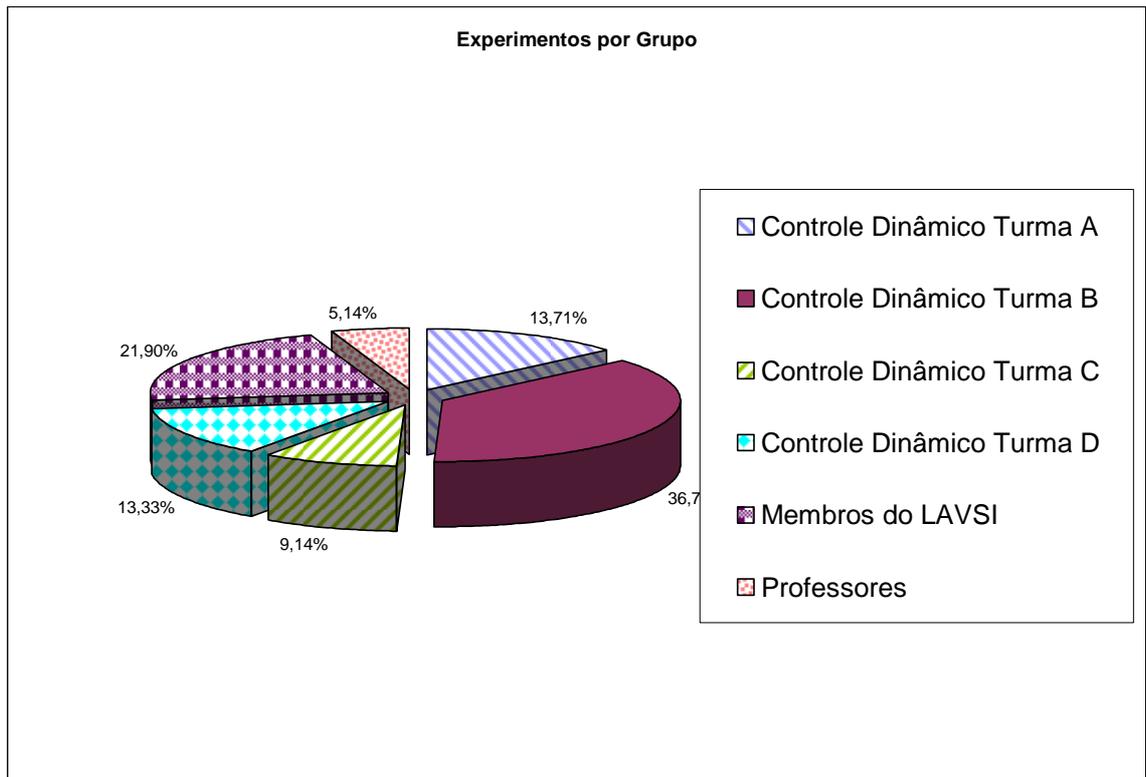


Figura 7-2 Percentual de Uso por Grupo

Estes dados revelam que:

- Os alunos da Turma B realizaram quatro vezes mais experimentos que os alunos da Turma C de Controle Dinâmico.
- Os alunos das turmas A e D realizaram quase o mesmo número de experimentos.

7.3. Execuções por Dia

O gráfico da Figura 7-3 apresenta a atividade no laboratório diariamente. No eixo X, temos a data e no eixo Y, o número de experimentos realizados.

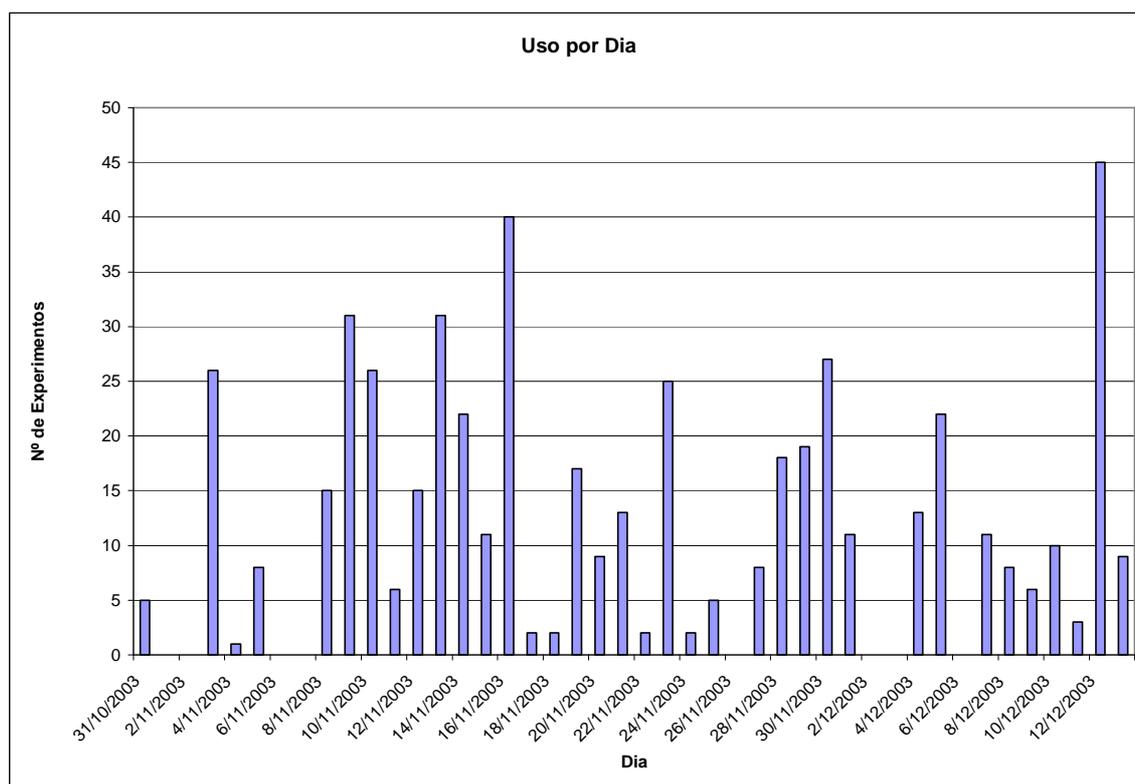


Figura 7-3 Número de Execuções por Dia

Os dados de execução por dia mostram grande volume de acesso em vésperas de entrega de relatórios e pouca movimentação em vésperas de provas, como provas de Controle Dinâmico. Com a finalidade de distribuir de forma homogênea o acesso por dia, as seguintes medidas são sugeridas:

- Limitação de execução de experimentos por dia a cinco para cada aluno;
- Datas de entrega de relatórios diferentes para cada turma.

8. Conclusão

O Sistema de Nível de Líquidos antes deste trabalho possuía as seguintes características³⁰:

- Um supervisor dependente de Controle PID,
- Somente o Controle PID estava disponível como método do Controle do Sistema de Nível de Líquidos;
- Utilização de plataforma e aplicativos proprietários de alto custo, como o IIS da Microsoft;
- Não existia controle de execução de experimentos;
- Os resultados de execuções de experimentos não eram armazenados;
- Quando o aplicativo de controle não estava em uso, o sistema ligava os atuadores em potência máxima;
- Os sensores não utilizavam a máxima tensão permitida pela placa A/D;
- Problemas de variação de tensão de alimentação, com uma única fonte para todo o sistema; e
- A vazão máxima dos atuadores era de $50 \frac{cm^3}{s}$.

O método e controle no Espaço de Estados, implementado neste trabalho, aumentou as possibilidades de controle do Sistema de Nível de Líquidos, permitindo aos alunos a aplicação conhecimento teórico sobre Espaço de Estados em sistemas reais e não somente simulações por software. Possibilitando, ainda, a criação de novos experimentos para o Laboratório de Controle Dinâmico e de Controle Digital.

O supervisor desenvolvido neste trabalho é independente do tipo de controle utilizado e restringe-se a apresentar os resultados de experimentos em tempo real. Esta

³⁰ Somente estão listadas as que sofreram modificação neste trabalho.

característica, além da independência do controle, possibilita a utilização de *multicasting*, onde outros usuários podem ter acesso aos resultados do experimento em execução.

O servidor *web* e os demais softwares utilizados, como o gerenciador de banco de dados e a tecnologia de confecção de páginas dinâmicas, são gratuitos e de fonte aberto, características importantes em ambientes de pesquisa.

Os aplicativos desenvolvidos seguem padrões e tendências mundiais em desenvolvimento de softwares, utilizando técnicas de orientação a objeto. Isso, além de melhorar a qualidade do laboratório é um fato que motiva pesquisadores e estudantes a trabalharem com estas tecnologias.

O banco de dados implementado armazena todas as informações de execuções, bem como realiza o controle destas, facilitando o acompanhamento dos experimentos e criando um ambiente de utilização em educação à distância. Além de possibilitar a geração de relatórios de uso dos experimentos e resultados obtidos.

As alterações na parte eletrônica do Sistema de Nível de Líquidos realizadas resolveram os seguintes problemas: queda de tensão em fonte; baixa potência utilizada dos atuadores; e travamento de softwares que habilitavam as moto-bombas em tensão máxima.

Com estas alterações a vazão máxima das moto-bombas aumentou para $130 \frac{cm^3}{s}$ em 10V de tensão e um circuito onde está tensão poderá atingir até 17V.

Com essas implementações, atualmente, o Laboratório de Ensino de Automação Remoto - LEARN é uma realidade. Possui um ambiente completo para experimentação remota e uma arquitetura que facilita a inclusão de novos experimentos como o Sistema de Automação Predial do Laboratório de Automação, Visão e Sistemas Inteligentes – LAVSI. Vale ressaltar, também, que ocorreram melhorias na documentação do Sistema de Nível de Líquidos.

**“O impossível nós fazemos rápido,
milagres são mais difíceis!”**

Adolfo Bauchspiess, com adaptações.

A era da informação exige mudança na estrutura educacional e na formação de novos Engenheiros. Com este intuito, o LEARN disponibiliza experimentos remotos e os aplica na disciplina Controle Dinâmico dos cursos de Engenharia Elétrica e Mecatrônica da Universidade de Brasília.

A arquitetura *web* do LEARN é moderna, segue padrões mundiais, criando um ambiente robusto e confiável para utilização em laboratórios remotos. Este ambiente utiliza somente softwares livres e testados, diminuindo os custos e melhorando a performance.

Em laboratórios presenciais o professor verifica a presença e ajuda os alunos. Com esse mesmo objetivo, o LEARN utiliza um banco de dados onde além de armazenar informações de execuções de experimentos armazena também os resultados de tais experiências. Estas informações possibilitam uma análise pós-experimentação e uma verificação mais eficiente pelo professor, pois este pode verificar a veracidade dos dados contidos nos relatórios.

Um laboratório com infra-estrutura eficiente também deve aumentar o número de experimentos disponíveis. para isso o Controle no Espaço de Estados pode ser utilizado pelos alunos do LEARN. O LEARN possibilita o uso de controladores no Espaço de Estados no Sistema de Nível de Líquidos.

9. Bibliografia

- [1] Gosmann, Hugo Leonardo. Um Sistema Multivariável de Tanques Acoplados para Avaliação de Técnicas de Controle. Dissertação de Mestrado, Publicação ENE.DM-001^a, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 76p.
- [2] Vicinio, A.: Automatic Control Telelab. [on-line] URL: <http://www.dii.unisi.it/~control/act/home.html>. Imagem capturada 2 de julho de 2003
- [3] Bauchspiess, A. Notas de Aula: Controle Digital. Departamento de Engenharia Elétrica, Universidade de Brasília, 2003.
- [4] Franklin, Gene F., J. David Powell, e Mich L. Workman. Digital Control of Dinamic Systems. Addison Wesley Publishing Company
- [5] Feedback Controle
- [6] Ogata Engenharia de Controle Moderno
- [7] Model View Controle Designer Patern, url: <http://java.sun.com>.
- [8] Apache Jakarta Struts FrameWork, <http://jakarta.apache.org/struts>.
- [9] Apache Jakarta TomCat, Web Server e Container JSP/Servlet, <http://jakarta.apache.org/tomcat>
- [10] Comunidade WEB FireBrid.Br, <http://www.firebird.com.br>
- [11] IbPhoenix, Comunidade de Consultoria e Produtos para FireBird, <http://www.ibphoenix.com>.
- [12] Fundação para Organização e Desenvolvimento do FireBird, <http://www.firebirdfoundation.org>
- [13] Apache Group WEB site, <http://www.apache.org>
- [14] Tiles FrameWork, <http://jakarta.apche.org>
- [15] Livro de banco de Dados
- [16] PEARL (Practical Experimentation by Accessible Remote Learning). [on-line] URL: <http://www.kmi.open.ac.uk/projects/pearl/>
- [17] Universidade de Erlangen (Friedrich-Alexander University Erlangen-Nuremberg) [on-line] URL: <http://www.uni-erlangen.org/>
- [18] Universidade virtual de Hagen – (Lernraum Virtuelle Universität – Hagen) [on-line] URL: <https://vu.fernuni-hagen.de/>
- [19] Universidade de Berlin (Freie Universität Berlin) [on-line] URL: <http://www.fu-berlin.de/>
- [20] Universidade da Florida (Florida University) [on-line] URL <http://www.ufl.edu/>
- [21] Laboratório Remoto da Universidade de Pisa [on-line] URL: <http://www.piaggio.cci.unipi.it/robotics/telelab-eng.html>

- [22] Laboratório Remoto da Universidade do Tennessee – UTC [on-line] URL: imagem capturada em 2 de julho de 2003
- [23] Laboratório Remota da Universidade de Cingapura [on-line] URL: Imagem capturada em 2 de julho de 2003
- [24] Laboratório Remoto da Universidade de Siena : Automatic Control Telelab. [on-line] URL: <http://www.dii.unisi.it/~control/act/home.html>.
- [25] LEARn – Pagina
- [26] Livro de TCP/IP
- [27] Trabalho do Rafael e do Luis
- [28] Trabalho do Bruno e Alexandre.
- [29] Trabalho do Bruno
- [30] Trabalho Alexandre.
- [31] Artigo Bruno Adolfo e Alexandre
- [32] Java sum.com
- [33] Jakarta Commons