

FERNANDO FERRAZ DE TOLEDO MACHADO
FREDERICO GUSTAVO DE SOUZA LIMA CAMPOS DE OLIVEIRA

PROJETO DE SISTEMA PARA TELECOMANDO E MONITORAÇÃO REMOTA



Trabalho de formatura apresentado
à Escola Politécnica da Universidade
de São Paulo para a obtenção dos
títulos de Graduado em Engenharia.

Orientador:
Prof. Dr. Newton Maruyama

São Paulo
2000

Machado, Fernando F. de T.; Oliveira, Frederico G. de S.
L. C. de

Projeto de Sistema para Telecomando e
Monitoração Remota.

São Paulo, 2000.

50p. + anexos.

Trabalho de Formatura – Escola Politécnica da
Universidade de São Paulo. Departamento de Engenharia
Mecânica.

1. Acesso remoto a experimentos do Laboratório
de Automação e Controle. 2. LabView. I. Universidade
de São Paulo. Escola Politécnica. Departamento de
Engenharia Mecânica. II. t

DEDALUS - Acervo - EPMN



31600005958

AGRADECIMENTOS

Ao nosso estimado orientador Prof. Dr. Newton Maruyama que, com seu acompanhamento, orientação e incentivo sempre presentes, tornou possível a execução deste trabalho.

Ao colega Gunther Mittermayer que nos ajudou com seus conhecimentos sobre comunicação cliente-servidor.

Ao colega Fábio Tozzeto Ramos pela colaboração e esclarecimentos oportunos.

Este trabalho é dedicado a Jacob Hounsgaard que foi o irmão mais velho de todos durante o curso.

A todos que direta ou indiretamente colaboraram na realização deste trabalho.

LISTA DE FIGURAS

Figura 1 –	Arquitetura do sistema de telecomando e monitoração remota.....	4
Figura 2 –	Fluxo de dados na tecnologia ASP.....	15
Figura 3 –	Interface gráfica do LabView.....	18
Figura 4 –	Painel do software remoto (cliente).....	19
Figura 5 –	Montagem simulada pelo programa para gerador de função - osciloscópio	20
Figura 6 –	Montagem simulada pelo programa para gerador de função - integrador – osciloscópio.....	21
Figura 7 –	Montagem simulada pelo programa para gerador de função - placa de aquisição - osciloscópio.....	23
Figura 8 –	Montagem simulada pelo programa para gerador de função - planta – osciloscópio.....	24

SUMÁRIO

LISTA DE FIGURAS

RESUMO

1. INTRODUÇÃO E OBJETIVOS.....	1
2. ABORDAGEM DO PROBLEMA.....	3
3. CRONOGRAMA.....	7
4. FORMULAÇÃO DO PROJETO.....	8
4.1. COMUNICAÇÃO ENTRE OS COMPUTADORES.....	8
4.2. SOFTWARE DE CONTROLE.....	9
4.3. SOFTWARE DO COMPUTADOR REMOTO.....	11
5. SOLUÇÕES ABORDADAS.....	13
5.1. CLIENTE-SERVIDOR.....	13
5.2. INTERFACE GRÁFICA.....	17
6. EXPERIMENTOS VIRTUAIS DISPONÍVEIS.....	19
6.1. RECURSOS DISPONÍVEIS.....	19
6.2. GERADOR DE FUNÇÃO – OSCILOSCÓPIO.....	20
6.3. GERADOR DE FUNÇÃO – INTEGRADOR – OSCILOSCÓPIO.....	21
6.4. GERADOR DE FUNÇÃO – PLACA DE AQUISIÇÃO DE DADOS – OSCILOSCÓPIO.....	22
6.5. GERADOR DE FUNÇÃO – PLANTA – OSCILOSCÓPIO.....	23
7. POSSÍVEIS DESENVOLVIMENTOS.....	26
8. CONCLUSÕES.....	27
9. BIBLIOGRAFIA.....	28
ANEXOS.....	29

RESUMO

Este projeto objetivou o estudo de alternativas para a criação de um sistema para telecomando e monitoração remota, viabilizando a criação de um laboratório virtual.

Numa primeira fase, dentre muitas foi selecionada uma alternativa baseada em webserver. Nessa solução, o servidor gerencia, junto com um aplicativo controlador da aquisição, os pedidos de conexão e o envio e recebimento de dados. Essa solução tornou-se inviável pois os aplicativos webserver analisados ou tinham como pré requisito um custo adicional de compra, ou exigiam uma interface a mais de comunicação com o software controlador da aquisição. Numa fase posterior, foi identificado como forte candidato a solução implementada através do LabVIEW, contemplando uma comunicação via protocolos TCP/IP e uma aquisição simulada de dados, a qual teria como evolução natural uma aquisição de dados via placa de aquisição. Foram assim analisados os parâmetros críticos dessa solução como velocidade e confiabilidade da comunicação, tendo com resultado um desempenho satisfatório.

Na fase de projeto, foram definidas as entradas e saídas de cada macro função utilizada pelo sistema. Depois de desenvolvido, o aplicativo foi testado em todas as funcionalidades disponíveis, determinando os reparos necessários para o seu perfeito funcionamento.

No final, evidenciou-se a viabilidade da idéia com o funcionamento completo do sistema, tendo-se observado como uma futura oportunidade a sua aplicação nos laboratórios da escola.

1. INTRODUÇÃO E OBJETIVOS:

O projeto propõe o desenvolvimento de uma infra-estrutura para dar suporte a atividades de ensino experimental à distância dentro do tema de Automação e Controle. Dentro desse contexto, equipamentos de um Laboratório Didático de Automação poderão ser controlados e monitorados remotamente através da rede Internet, possibilitando aos alunos uma forma alternativa para a realização de experimentos.

Um Laboratório Virtual (L.V.) proporciona aos alunos acesso contínuo a seus experimentos hipotéticos ou reais e está cada vez mais se tornando viável graças ao desenvolvimento da rede Internet. Com o Laboratório Virtual, os alunos não precisarão ir até a Universidade para realizarem o experimento e não será preciso que professores fiquem disponíveis no laboratório durante a execução dos mesmos.

Esse tipo de contato dos alunos com os experimentos pode incentivar o interesse dos mesmos não só no que tange as atividades do laboratório mas como o projeto e a implementação de um controlador remoto e suas muitas aplicações.

Outra característica importante do L. V. é que os experimentos podem estar disponíveis para não-alunos devidamente autorizados que tenham acesso à Internet.

Podem ser desenvolvidos experimentos reais, ou seja, que vão existir fisicamente dentro de um espaço na Universidade, e experimentos virtuais, que podem ser simulados pelo computador servidor. Esta possibilidade aumenta

indefinidamente a quantidade de experimentos que podem estar disponíveis para os alunos sem adicionar custo algum ao desenvolvimento do laboratório.

O fato do aluno estar realizando experimentos virtuais gera, porém, um problema de falta de contato com a realidade física dos mesmo, portanto alguns experimentos reais devem estar disponíveis para, ao menos, minimizar este problema.

2. ABORDAGEM DO PROBLEMA

Como arquitetura geral do processo de controle remoto, mostrada na Figura 1, o usuário remoto adquire via Internet um programa escrito em linguagem G para ser executado no LabView assim como um arquivo texto de instruções. Este programa tem interface definida e um conjunto de referências de entradas para operar cada experiência onde usuário remoto pode alterar as variáveis de entrada conforme interesse para verificar os efeitos de tais alterações na saída do sistema.

O usuário deve abrir o arquivo no LabView e acertar os parâmetros e o tipo de simulação conforme instruções. O experimento pode se executado durante um ciclo ou continuamente e o servidor transmite a resposta do sistema ao cliente assim que ela é gerada. O envio simultâneo da saída do sistema ao computador remoto pode ser feito de duas maneiras, utilizando o protocolo de comunicação UDP (User Datagram Protocol) ou o TCP (Transmission Control Protocol). TCP é uma conexão orientada que garante fluxo bidirecional de dados. No caso de perda de dados, TCP requer uma retransmissão e ainda pode descartar dados duplicados ou reorganizar dados fora de ordem. UDP oferece menor tempo de transmissão porém implementa uma conexão onde dados podem ser perdidos, duplicados ou chegarem fora de sequência. Além disso os dados recebidos em tempo real estariam sujeitos a atrasos de transmissão de dados da rede. O protocolo utilizado foi o TCP.

O usuário pode alterar os parâmetros e selecionar outro tipo de experimento conforme desejado e algumas simulações geram arquivos com a saída do sistema para que o usuário possa utilizar outras ferramentas como o MatLab ou o Excel para análise matemática.

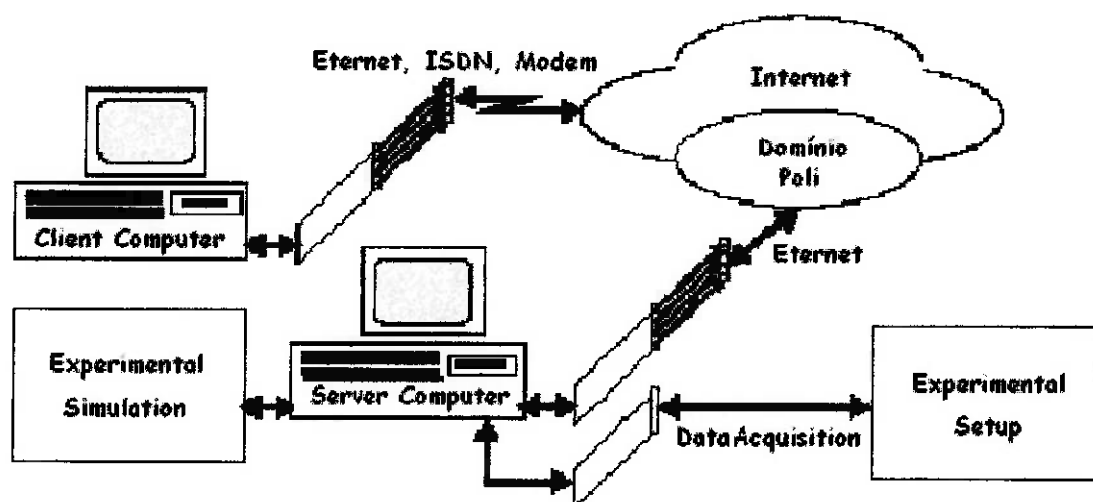


Figura 1: Arquitetura do sistema de telecomando e monitoração remota

Na Figura 1, percebe-se os dois tipos possíveis de experimento, os simulados e os reais. Em ambos o cliente envia os parâmetros via internet para o servidor que simula o experimento ou envia os parâmetros para o conversor digital-analógico para que possa ser recebido pela planta. A resposta é então adquirida ou gerada no computador remoto e enviada ao cliente.

Essa abordagem permite ao usuário opções de controle do experimento, o que garante maior responsabilidade e autonomia ao cliente. O usuário implementa remotamente o controle e as vantagens disso são:

- A possibilidade de utilizar clientes “poderosos” para a implementação de controladores complexos. Esta é a filosofia de um processo de controle centralizado onde um computador central (cliente) monitora um grupo de sensores ou atuadores e os alimenta com seus sinais de comando. Ao mesmo tempo, isso favorece a utilização de computadores servidores menos poderosos e mais baratos já que suas funções são encaminhar os sinais de controle aos atuadores, amostrar dos sensores e encaminhar estes sinais de volta para o cliente. O custo da implementação de um Laboratório Virtual baseado nessa filosofia de cliente(controlador)/servidor é consideravelmente mais barata que a filosofia tradicional de um computador dedicado para cada experimento.
- Habilidade de implementar controles avançados sem depender de possíveis limitações no lado do servidor como limitação de tamanho ou consumo de energia.
- Privacidade do código de controle implementado, já que o controle é implementado e executado remotamente.
- Flexibilidade em projetar o processo de controle já que o usuário está livre para implementar qualquer tipo de controle no servidor.
- Melhor segurança contra “downloads” de vírus desde que o servidor não compila programas fornecidos pelo cliente.

Neste trabalho foi implementado somente o Laboratório Virtual com experimentos simulados. Todos os experimentos virtuais são adaptáveis a

experimentos reais bastando para isso substituir o sinal gerado pelo servidor por um sinal adquirido por uma placa de aquisição de dados. O programa LabView possui drivers prontos para diversos tipos de placas de aquisição de dados, o que torna a adaptação simples.

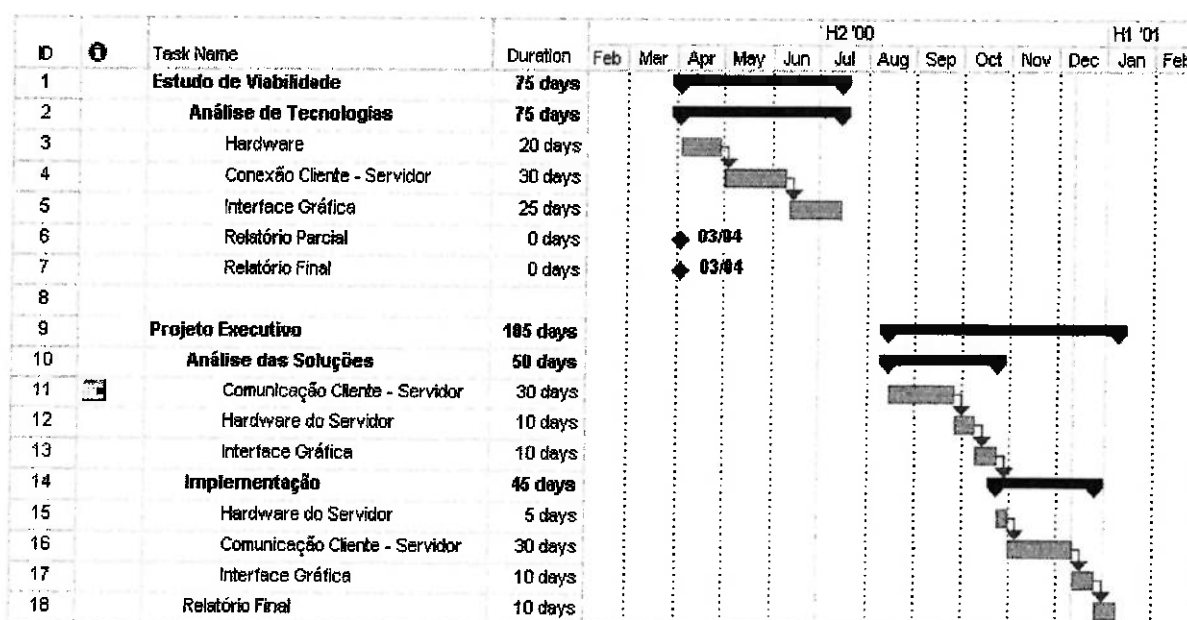
Na fase de projeto foram as seguintes definidas tecnologias para posterior implementação:

- Comunicação entre os computadores (remoto e servidor),
- Definição do software do computador servidor que controla os experimentos,
- Definição do software que atua no computador remoto que recebe dados e envia comandos ao computador servidor,
- Interface gráfica do computador remoto para display dos dados recebidos.

3. CRONOGRAMA

As atividades e o cronograma do trabalho foram alteradas em virtude do escopo de estudo de viabilidade da primeira etapa e problemas na implementação.

Dentro dessas mudanças os prazos foram atingidos com sucesso, viabilizando as atividades que foram cumpridas no projeto executivo.



4. FORMULAÇÃO DO PROJETO

4.1. COMUNICAÇÃO ENTRE COMPUTADORES

4.1.1. Especificações Funcionais

- **Desempenho:** A comunicação entre os computadores deve satisfazer critérios de desempenho com relação a velocidade, confiabilidade e segurança, pré definidos para atender uma transmissão eficiente de dados. Essa conexão cliente servidor também deve ser independente da plataforma do computador remoto, já que o objetivo do sistema é possibilitar o acesso dos alunos de maneira mais simples possível, ou seja, sem depender da máquina e sistema operacional que o aluno estiver usando.

- **Conforto:** A comunicação deve ser feita de maneira que o computador remoto não precise de requisitos mínimos sofisticados, tornando a comunicação complexa e custosa. Um modem e um software de comunicação devem satisfazer os requisitos mínimos para o cliente se conectar ao computador servidor.

- **Segurança:** A comunicação entre os computadores deve prover segurança suficiente para garantir que nenhum usuário sem acesso permitido possa se conectar ao computador servidor.

4.1.2. Especificações Operacionais

- **Confiabilidade:** A confiabilidade da comunicação exige pré requisitos como uma comunicação onde as informações que estão sendo recebidas são

as que foram enviadas, um tempo estável de conexão mínimo suficiente para a aquisição e transmissão dos dados.

4.1.3. Especificações Construtivas

- **Capacidade:** A comunicação entre os computadores deve ser capaz de transmitir dados confiáveis bidirecionalmente e em velocidade suficiente para que o usuário adquise os dados sem que o atraso da transmissão dos dados prejudique o comando.

4.1.4. Entradas e Saídas

- **Desejáveis:** Dados que possam ser transmitidos e compreendidos pelos computadores (servidor e remoto).
- **Indesejáveis:** Dados que não possam ser transmitidos ou compreendidos pelos computadores (servidor e remoto).

4.2. SOFTWARE DE CONTROLE

4.2.1. Especificações Funcionais

- **Desempenho:** o software de controle deve ser capaz de receber informações sobre o tipo de operação desejada e, com base nisso, determinar qual a ordem de sinais que devem ser enviados para a execução da tarefa. A linguagem deve permitir a comunicação com outros softwares locais ou remotos.

- **Segurança:** o software deve fazer o monitoramento da operação e estar pronto para cancelá-la se receber algum comando remotamente do usuário ou alguma informação de mau andamento do processo.

4.2.2. Especificações Operacionais

- **Confiabilidade:** o software deve estar livre de “bugs” e a linguagem de programação deve ser robusta o suficiente para executar o programa.

- **Controlabilidade:** o software deve permitir entradas durante sua operação e utilizá-las para executar os novos comandos.

4.2.3. Especificações Construtivas

- **Capacidade:** o programa deve minimizar o tempo entre o recebimento e o envio de dados para que não se tenha atrasos significativos nos resultados.

- **Updating:** o programa deve ter a possibilidade de ser substituído por versões mais eficientes se melhores soluções forem encontradas.

4.2.4. Entradas

- **Desejáveis:** deve-se entrar com dados corretos dentro de limites e opções pré-estabelecidas e sinais de monitoramento compreensíveis pelo programa.

- **Indesejáveis:** qualquer dado que não faça sentido ou gere uma saída impossível de implementar.

4.2.5. Saídas

- **Desejáveis:** sinais de comando corretos e dados corretos para o computador remoto.

- **Indesejáveis:** sinais de comando incorretos ou dados incorretos para o computador remoto.

4.3. SOFTWARE DO COMPUTADOR REMOTO

4.3.1. Especificações Funcionais

- **Desempenho:** o software deve ser capaz de receber as informações provenientes da rede e enviar sinais de comando como escolha do tipo de operação e cancelamento do processo. A linguagem deve ser capaz de realizar esse tipo de troca de informações. Além disso, deve ter uma interface gráfica de modo que os resultados possam ser vistos na forma de gráficos, planilhas, etc.

- **Conforto:** o software deve ser escrito em uma linguagem que possa rodar em qualquer tipo de máquina ou plataforma já que estará atuando no computador remoto sem que o usuário necessite de modificações ou instalações em seu computador pessoal.

4.3.2. Especificações Operacionais

- **Confiabilidade:** o programa deve estar livre de "bugs" e atenção especial deverá ser dada aos dados que serão enviados ao computador servidor.

- **Controlabilidade:** O software deve ter uma interface simples com comandos bem definidos e acessíveis.

4.3.3. Especificações Construtivas

Capacidade: o programa deve minimizar o tempo entre o recebimento e o envio de dados para que não se tenha atrasos significativos nos resultados.

Updating: o programa deve ter a possibilidade de ser substituído por versões mais eficientes se melhores soluções forem encontradas.

4.3.4. Entradas

- **Desejáveis:** Receber dados coerentes do computador servidor e comandos corretos do usuário do computador remoto.

- **Indesejáveis:** Receber dados que não possam ser interpretados do computador servidor e comandos do usuário do computador remoto que não possam ser entendidos.

4.3.5. Saídas

- **Desejáveis:** O usuário deve receber resultados coerentes da experiência. Deve ter também uma interface gráfica para uma maior interação do usuário com o processo em andamento.

- **Indesejáveis:** O usuário não deve receber resultados distorcidos e o computador remoto também não deve enviar ao servidor sinais de comando incoerentes.

5. SOLUÇÕES ABORDADAS

5.1. CLIENTE-SERVIDOR

5.1.1. TCP/IP: Transmission Control Protocol e Internet Protocol

Overview

Através de protocolos de comunicação pode-se implementar uma conexão cliente-servidor que satisfaça os pré requisitos do projeto. O protocolo TCP/IP, por ser uma combinação de protocolos de comunicação que viabiliza uma transmissão de dados confiável e poderosa (internet), atende as funcionalidades requeridas do projeto.

Esse tipo de protocolo já vem instalado na maioria dos PCs pois ele é a base de comunicação da rede mundial (internet). Ele é caracterizado como confiável pois ele verifica se os dados foram realmente transmitidos e confirma se foram transmitidos corretamente. Devido a essa verificação, esse protocolo tem como ponto fraco a baixa velocidade de comunicação com relação a outros protocolos, o UDP por exemplo.

Implementação,

A implementação desses protocolos é feita em baixo nível o que dificulta a sua utilização. No entanto, alguns métodos já implementados por alguns objetos do LabVIEW permitiram um nível de programação mais alto, facilitando a implementação de uma conexão através de funções como "abre conexão", "escreve dados" e etc.

5.1.2. ASP - Active Server Pages

Overview

Através de páginas ASP pode-se implementar uma comunicação entre um servidor e um computador remoto via internet. O conceito de páginas ASP consiste na divisão de processos entre estações clientes e servidores, com a finalidade de buscar melhor performance, menor tempo de resposta e maior facilidade de manutenção.

Active Server Pages são páginas web que possuem conteúdo dinâmico. Tais páginas consistem em arquivos de extensão .asp que contêm combinações de Server-Side scripts e tags HTML. Todo o código de programação existente em páginas Asp é executado no servidor, e este retorna ao cliente somente respostas em HTML padrão – o que faz com que aplicações Asp possam ser acessadas por qualquer browser existente no mercado. Uma aplicação feita em Asp pode ainda conter linhas de Client-Side script, que serão executados na estação cliente.

Implementação

Por essas páginas estarem hospedadas num servidor Microsoft Information Server, essa solução possui uma limitação muito forte, já que é desenvolvida pela Microsoft e assim, existiria um custo adicional pela sua implementação.

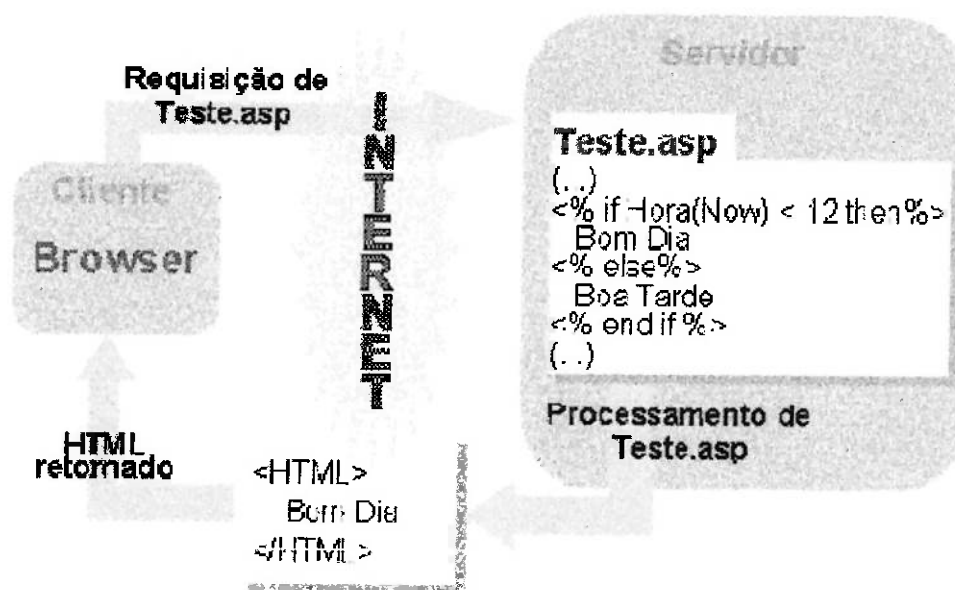


Figura 2: Fluxo de dados na tecnologia ASP

Plataformas (compatibilidade).

Páginas asp necessitam ser hospedadas no servidor Web da Microsoft: o Internet Information Server (IIS) . Este servidor deve ser instalado numa máquina NT Server 4.

Assim a sua compatibilidade com sistemas heterogêneos também fica limitada, pois a única plataforma na qual seria possível a sua implementação seria Windows.

5.1.3. JSP (Java Server Pages)

Overview

Java Server Pages são páginas com conteúdo dinâmico desenvolvidas em java. Esse tipo de ferramenta possibilita a comunicação cliente-servidor pois o código dessas páginas dinâmicas fica armazenado no servidor e é

acionado quando o usuário chama a página no browser do computador remoto. Dessa maneira, o software de escrita de parâmetros e leitura de dados pode ser desenvolvido em java, e o request ser gerado via browser no cliente.

Páginas JSP têm como grande vantagem serem desenvolvidas em Java o que permite uma independência de plataforma, o que é uma característica poderosa para aplicativos de internet.

No entanto, analisando a sua integração com a ferramenta LabVIEW, que será responsável pela aquisição de dados no servidor, o JSP apresenta pontos críticos de integração. O LabVIEW permite o acesso de seus objetos via activeX, o que acaba dificultando a sua comunicação com o JSP.

5.1.4. Datasocket Server

Overview

Datasocket server é uma ferramenta que permite que o computador dedicado a aquisição armazene os dados aquisitados em um servidor de dados que permitirá a leitura remota de outros computadores. Dessa maneira, os dados aquisitados podem ser acessados por qualquer computador conectado a internet, o que é um pré requisito do projeto. Essa solução pode ser implementada através de objetos do LabVIEW e o ComponentWorks do Visual Basic.

Existe uma pré condição que limita essa solução. O software de aquisição de dados já deve estar rodando no servidor com parâmetros pré determinados, pois o cliente não consegue acionar o servidor via

ComponentWorks. Dessa maneira, o experimento ficaria limitado a uma única condição.

Implementação

O datasocket server pode ser facilmente implementado através de objetos do LabVIEW e de componentes do ComponentWorks (add on do Visual Basic).

5.2.INTERFACE GRÁFICA

5.2.1. Excel

O Excel é uma das soluções para a interface gráfica. Através do desenvolvimento de macros pode-se gerar uma interface via planilha aonde o usuário entra com os parâmetros em algumas células e gera gráficos através de alguns botões associados a essas macros.

Essa interface está associada ao uso do visual basic como base de desenvolvimento do software de comunicação. Dessa maneira, o cliente não consegue acionar o servidor remotamente para fazer a aquisição dos dados.

5.2.2. LabVIEW

A interface desenvolvida através do LabVIEW proporciona um ambiente bastante interativo e didático (figura abaixo), possibilitando a visualização e controle da aquisição dos dados em tempo de execução.

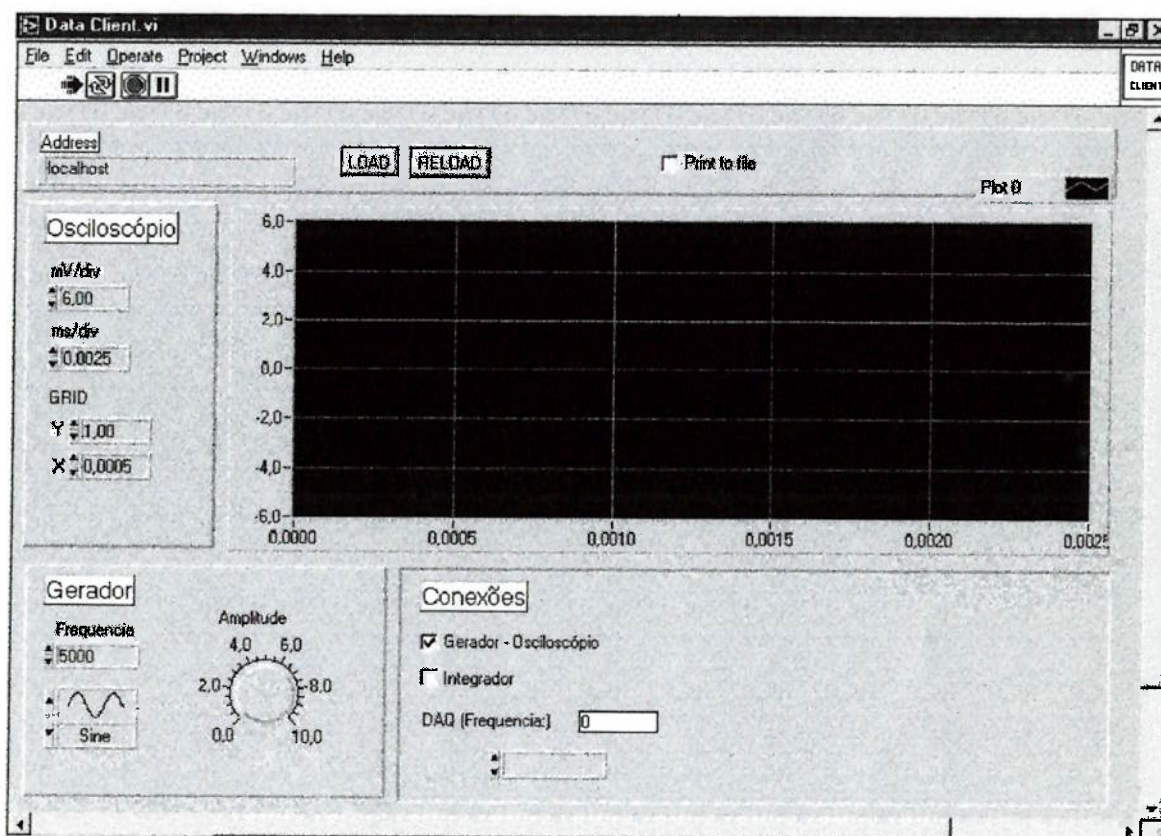


Figura 3: Interface gráfica do LabView

Existe uma restrição forte dessa interface gráfica, ela deve ser rodada dentro do LabVIEW. Uma possível alternativa seria gerar um arquivo executável através de uma versão mais nova do LabVIEW, que permite a compilação do programa gerado dentro desse ambiente. Assim, o executável gerado fica independente do LabVIEW.

6. EXPERIMENTOS VIRTUAIS DISPONÍVEIS

6.1. RECURSOS DISPONÍVEIS

O software que fica disponível para o cliente apresenta a interface gráfica representada na figura 4. É possível selecionar quatro tipos de montagem para a simulação:

- Gerador de Função - Osciloscópio
- Gerador de Função - Integrador - Osciloscópio
- Gerador de Função - Placa de Aquisição de Dados - Osciloscópio
- Gerador de Função - Planta – Osciloscópio

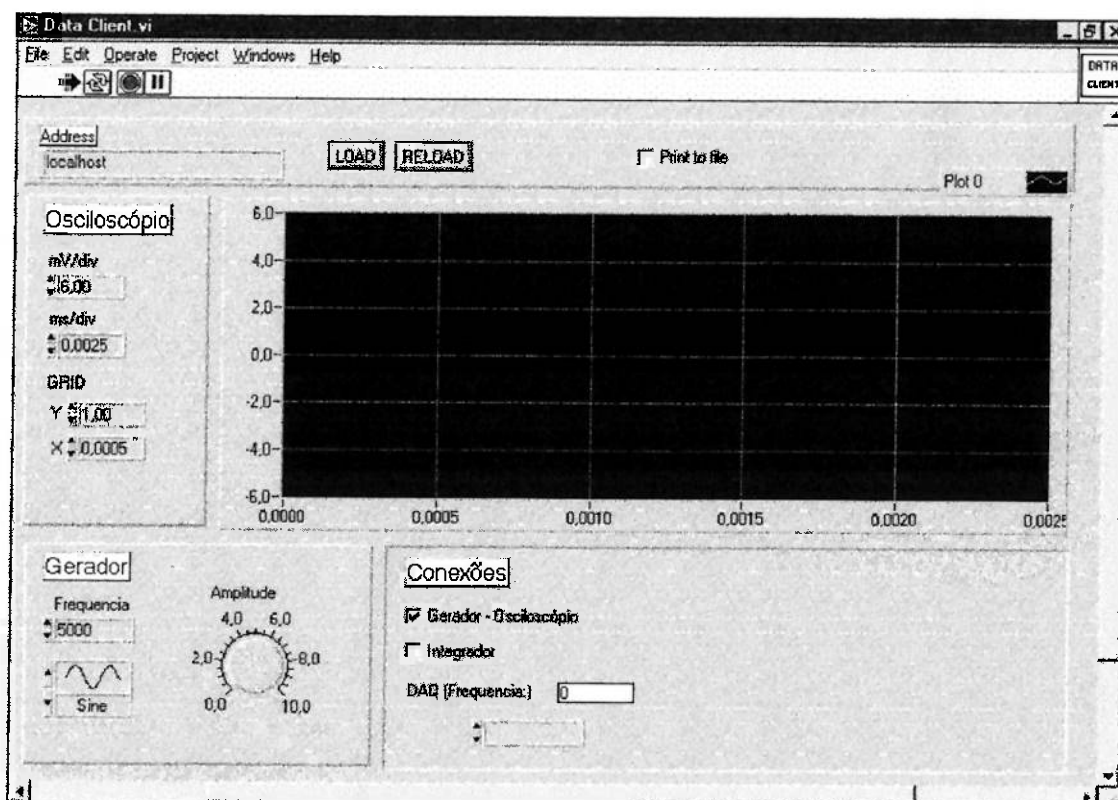


Figura 4: Painel do software remoto (cliente)

6.2. GERADOR DE FUNÇÃO - OSCILOSCÓPIO

6.2.1. Objetivo

Familiarização do usuário com o programa e sua interface gráfica.

6.2.2. Descrição

A saída do simulador de gerador de função é ligada diretamente ao osciloscópio cuja tela fica disponível no monitor do cliente para que o mesmo possa observar o sinal gerado e comparar se a frequência verificada no osciloscópio é a mesma frequência selecionada no gerador de função.

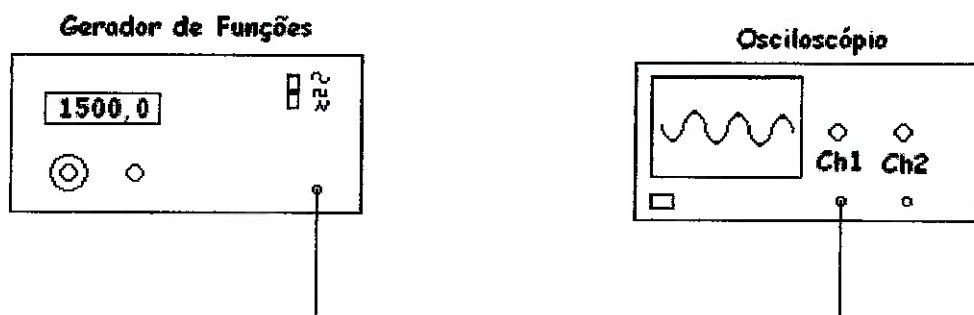


Figura 5: Montagem simulada pelo programa para gerador de função - osciloscópio

6.2.3. Instruções de Funcionamento

O usuário deve selecionar no painel a opção *Gerador - Osciloscópio* no quadro *Conexões*, deve escolher a frequência, amplitude e tipo de onda no quadro *Gerador* e ajustar as opções *mV/div*, *ms/div* e *GRID* para melhor visualização do sinal de saída.

6.3. GERADOR DE FUNÇÃO - INTEGRADOR - OSCILOSCÓPIO

6.3.1. Objetivo

Familiarização do usuário com o programa e sua interface gráfica e utilização dos dados de saída para construção do diagrama de Bode do integrador.

6.3.2. Descrição

A saída do simulador de gerador de função é ligada ao integrador e a saída deste é ligada ao osciloscópio cuja tela fica disponível no monitor do cliente para que o mesmo possa observar o sinal de saída do integrador e comparar sua frequência, amplitude (ganho) e fase com os mesmos dados do sinal do gerador de função. O programa gera um arquivo contendo o sinal de saída do integrador para análise matemático do mesmo pelo usuário no MatLab ou Excel.

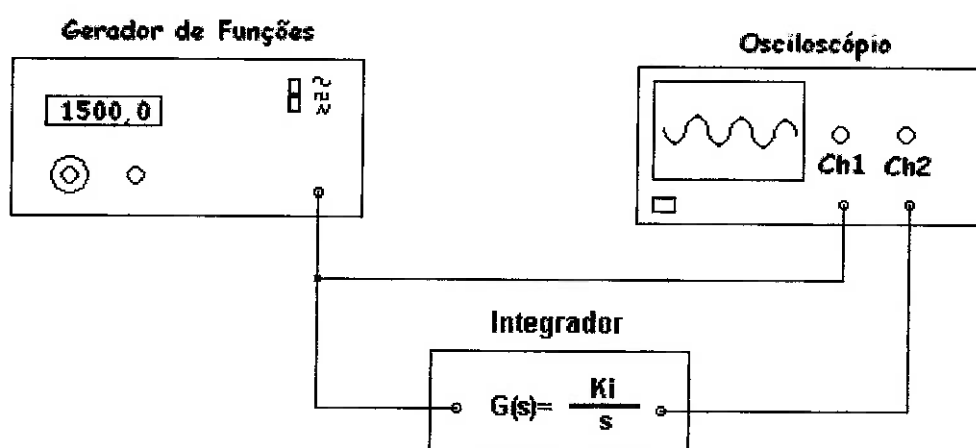


Figura 6: Montagem simulada pelo programa para gerador de função - integrador – osciloscópio

6.3.3. Instruções de Funcionamento

O usuário deve selecionar no painel a opção *Integrador* no quadro *Conexões*, deve escolher a frequência, amplitude e tipo de onda senoidal no quadro *Gerador* e ajustar as opções *mV/div*, *ms/div* e *GRID* para melhor visualização do sinal de saída.

6.4. GERADOR DE FUNÇÃO - PLACA DE AQUISIÇÃO - OSCILOSCÓPIO

6.4.1. Objetivo

Observação da amostragem de sinais escolha da frequência de amostragem correta e verificação das possíveis distorções causadas no processo.

6.4.2. Descrição

A saída do simulador de gerador de função é ligada à placa de aquisição de dados e a saída deste é ligada ao osciloscópio cuja tela fica disponível no monitor do cliente para que o mesmo possa observar o sinal adquirido e comparar sua frequência, amplitude (ganho) e fase com os mesmos dados do sinal do gerador de função. O programa gera um arquivo contendo o sinal de saída do integrador para análise matemático do mesmo pelo usuário no MatLab ou Excel.

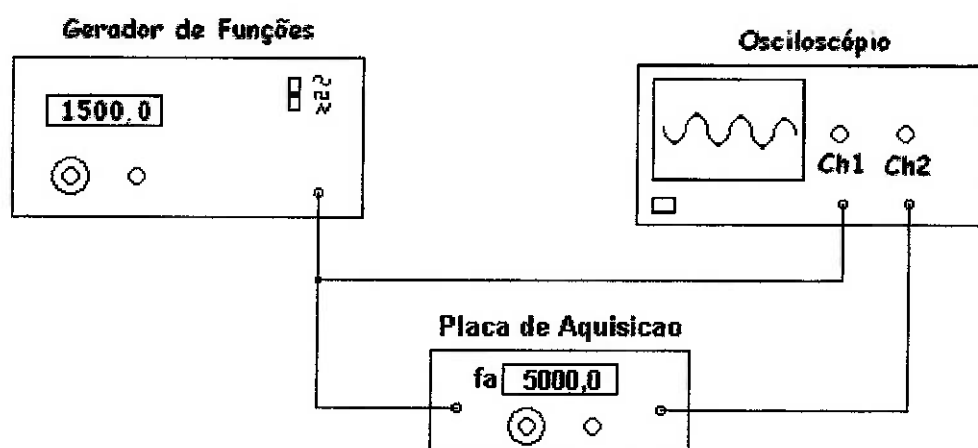


Figura 7: Montagem simulada pelo programa para gerador de função - placa de aquisição - osciloscópio

6.4.3. Instruções de Funcionamento

O usuário deve selecionar no painel a opção *Gerador-Osciloscópio* no quadro *Conexões* e escolher a frequência de amostragem, deve escolher a frequência, amplitude e tipo de onda no quadro *Gerador* e ajustar as opções *mV/div*, *ms/div* e *GRID* para melhor visualização do sinal de saída.

6.5. GERADOR DE FUNÇÃO - PLANTA - OSCILOSCÓPIO

6.5.1. Objetivo

Análise dos sinais de entrada e saída da planta para determinação dos parâmetros necessários para o projeto do controlador.

6.5.2. Descrição

A saída do simulador de gerador de função é ligada à placa de aquisição de dados. O sinal adquirido vai para a planta e a saída desta é novamente adquirida pela placa e finalmente vai para o osciloscópio, cuja tela fica disponível no monitor do cliente para que o mesmo possa observar o sinal de saída e comparar sua frequência, amplitude (ganho) e fase com os mesmos dados do sinal do gerador de função. O programa gera um arquivo contendo o sinal de saída do integrador para análise matemática do mesmo pelo usuário no MatLab ou Excel.

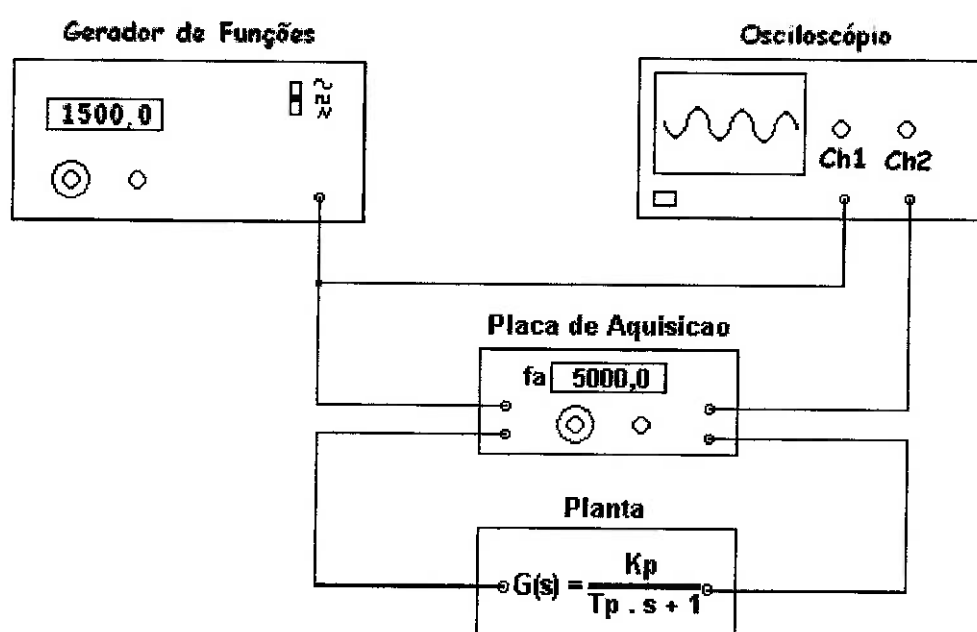


Figura 8: Montagem simulada pelo programa para gerador de função - planta - osciloscópio

6.5.3. Instruções de Funcionamento

O usuário deve selecionar no painel a opção *Gerador-Osciloscópio* no quadro *Conexões* e escolher a frequência de amostragem, deve escolher a frequência, amplitude e tipo de onda no quadro *Gerador* e ajustar as opções *mV/div*, *ms/div* e *GRID* para melhor visualização do sinal de saída.

7. POSSÍVEIS DESENVOLVIMENTOS

O modo como foi implementada a solução permite a utilização do programa para o gerenciamento não só de outros experimentos virtuais como também para o de experimentos reais.

Para a utilização de experimentos reais deve-se instalar uma placa de aquisição de dados para coletar sinais provenientes de um gerador de função ou planta integrando facilmente com o programa de comunicação cliente-servidor já desenvolvido.

Novos experimentos virtuais podem ser gerados e executados no servidor utilizando a mesma interface gráfica que foi desenvolvida neste trabalho.

Pode-se ainda, em versão mais atualizada do LabVIEW, compilar os programas desenvolvidos (VIs) permitindo que a conexão cliente-servidor seja independente do LabVIEW no computador remoto.

8. CONCLUSÕES

A detalhada análise das possíveis soluções permitiu selecionar a proposta mais adequada para os requisitos de projeto. Um dos pontos-chaves da seleção foi a comunicação cliente-servidor o qual acabou determinando como pré-requisito o software LabVIEW tanto no cliente como no servidor. O LabVIEW como base de desenvolvimento do projeto permitiu que o cliente determinasse os parâmetros de aquisição dinamicamente, tornando a aplicação didática e dinâmica. Outro fator que decidiu a seleção foi o fato de versões mais completas do LabVIEW permitirem a criação de executáveis o que viabiliza a futura independência do LabVIEW.

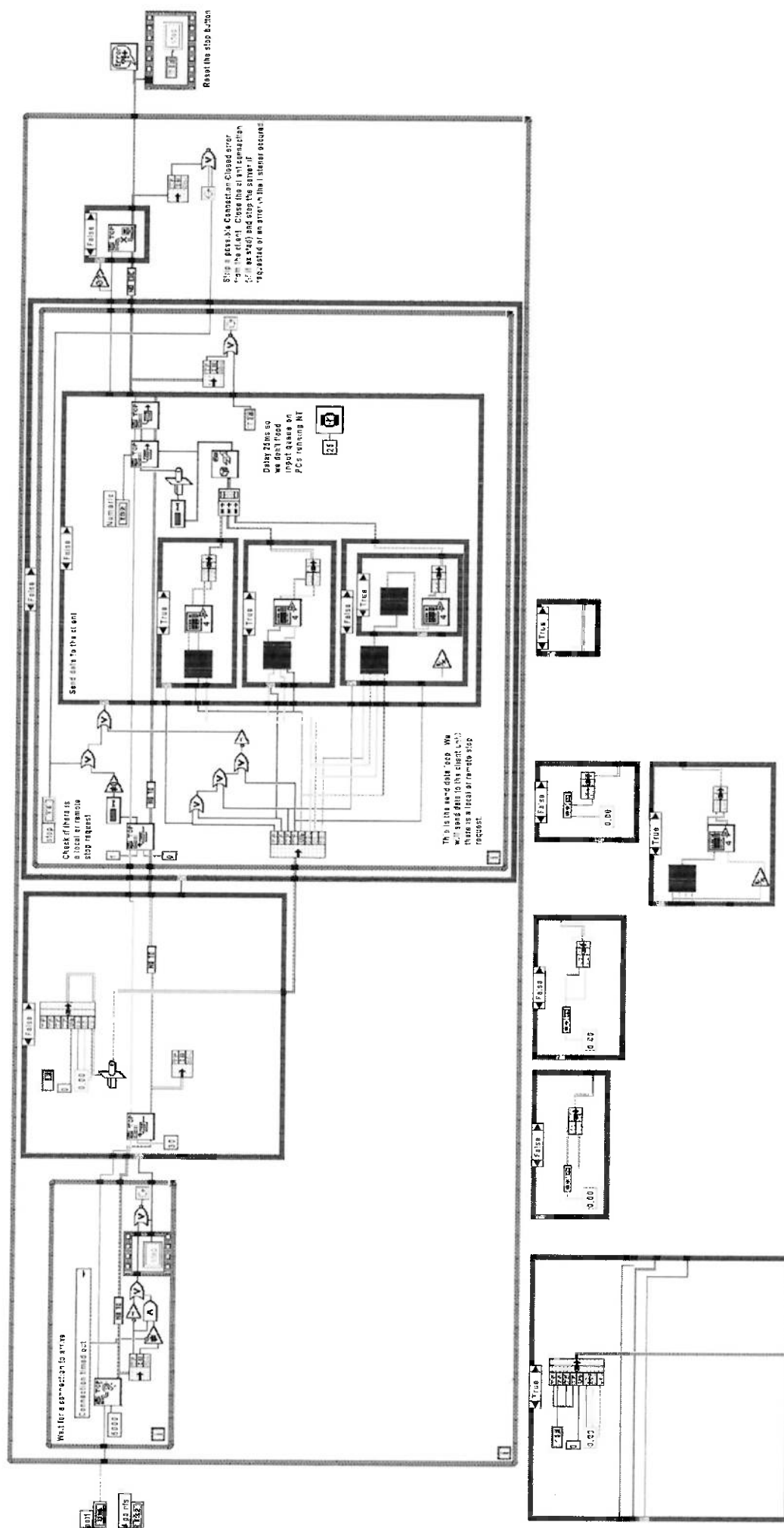
O projeto permite a implementação de outros experimentos virtuais que podem ser facilmente adaptados ao modelo desenvolvido assim como o gerenciamento de experimentos reais já existentes no Laboratório de Automação e Controle. Este fator é de extrema importância pois permite que alunos tenham acesso a uma grande variedade de sistemas sem gerar gasto para o departamento.

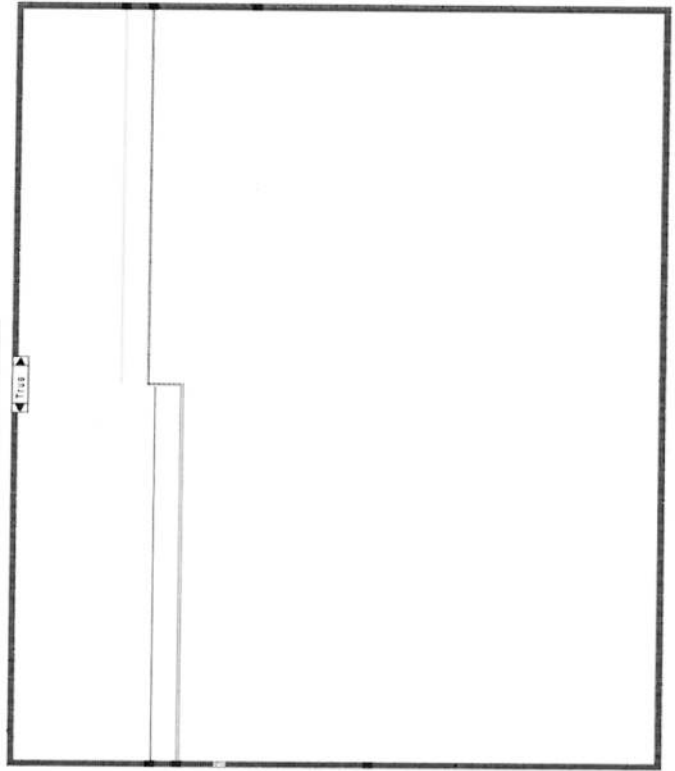
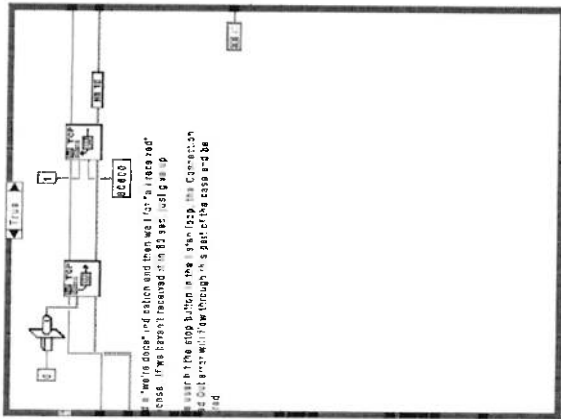
9. BIBLIOGRAFIA

- [01] Jamahl W. Overstreet e Anthony Tzes, "An Internet-Based Real-Time Control Engineering Laboratory", Polytechnic University, Outubro 99.
- [02] Eduardo L. L. Cabral e Antônio Carlos de Lima, "Familiarização com os Equipamentos do Laboratório e Amostragem Digital de Sinais Contínuos - Parte 1", Escola Politécnica da USP.
- [03] Eduardo L. L. Cabral, "Projeto de Controlador no Domínio da Frequência e Implementação Digital", Escola Politécnica da USP.
- [04] <ftp.unicamp.com.br>
- [05] www.ni.com
- [06] www.geocities.com/SiliconValley/Network/7460/corba.html
- [07] www.net.unicamp.br
- [08] "Estudo de Viabilidade", Escola Politécnica da USP
- [09] "Projeto Executivo", Escola Politécnica da USP

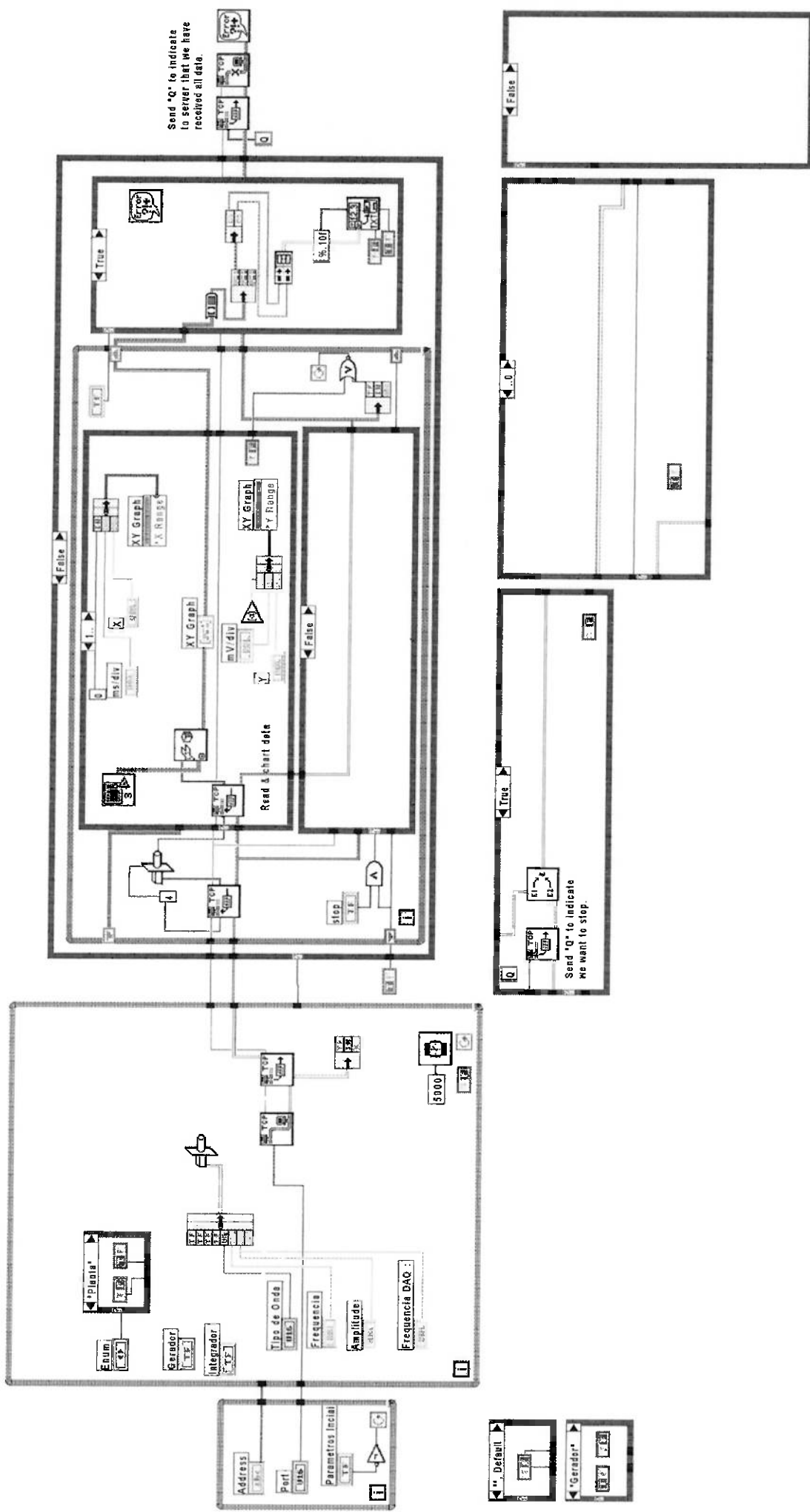
ANEXOS

Block Diagram



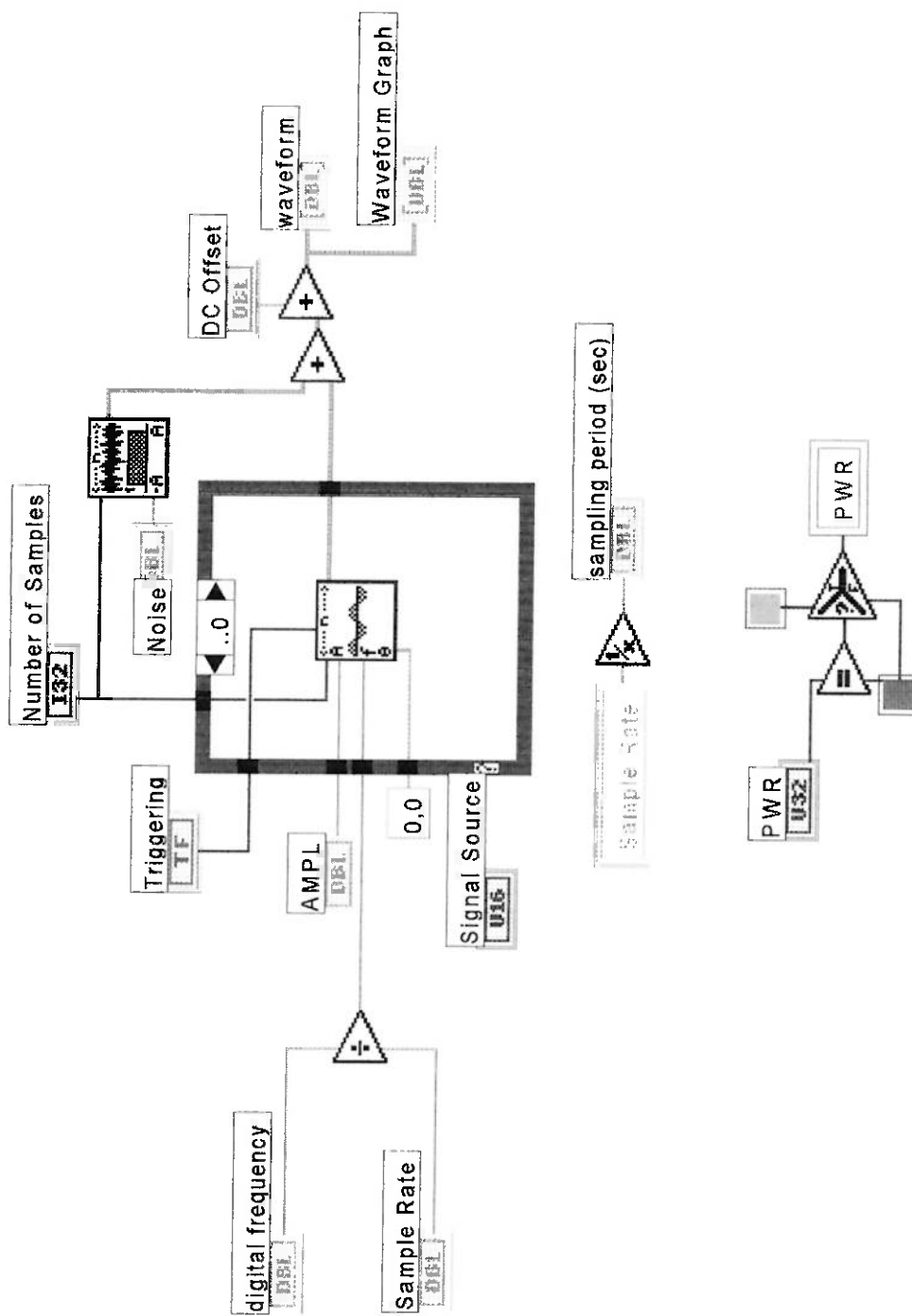


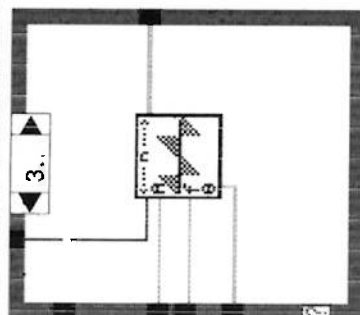
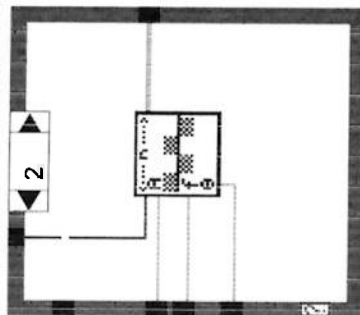
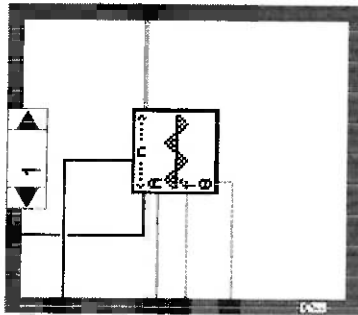
Block Diagram



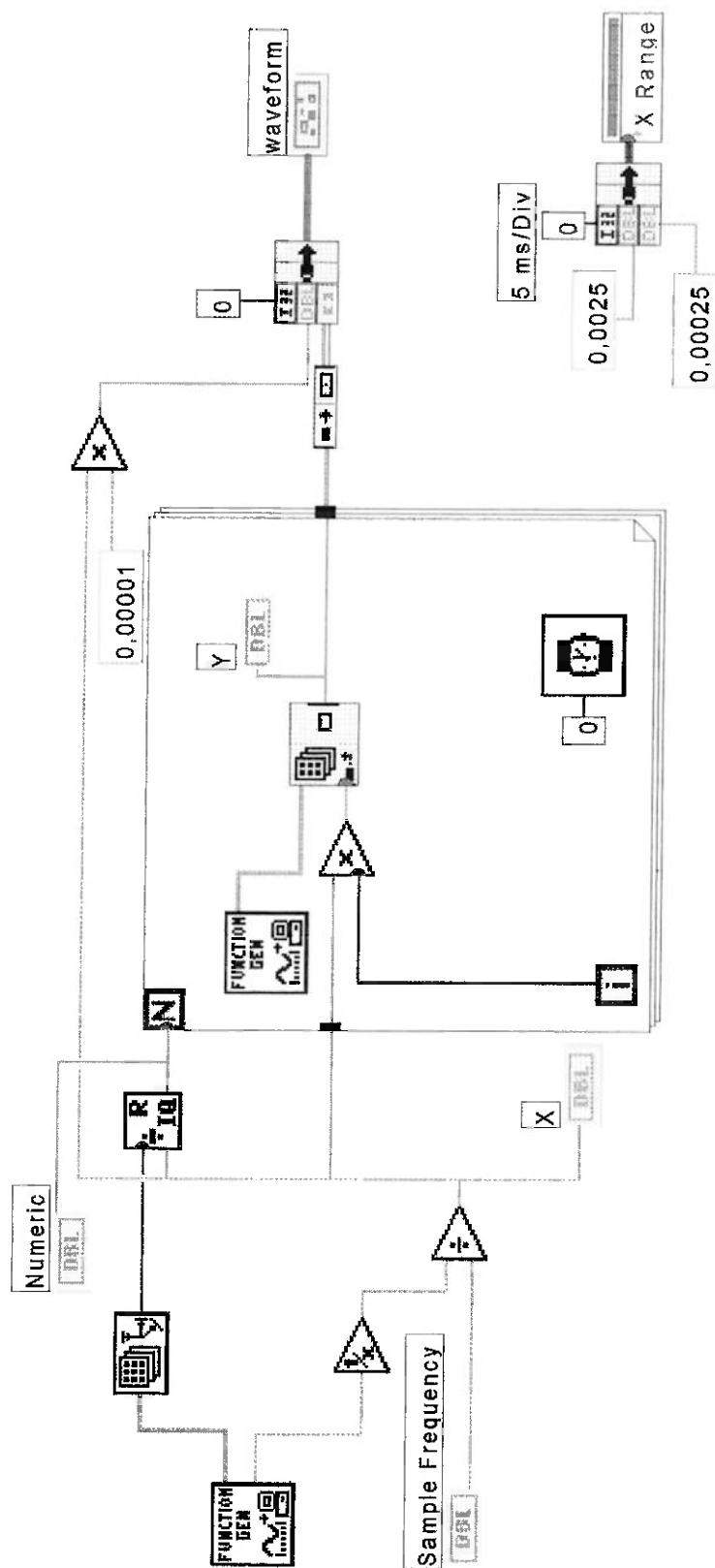
True	

Block Diagram

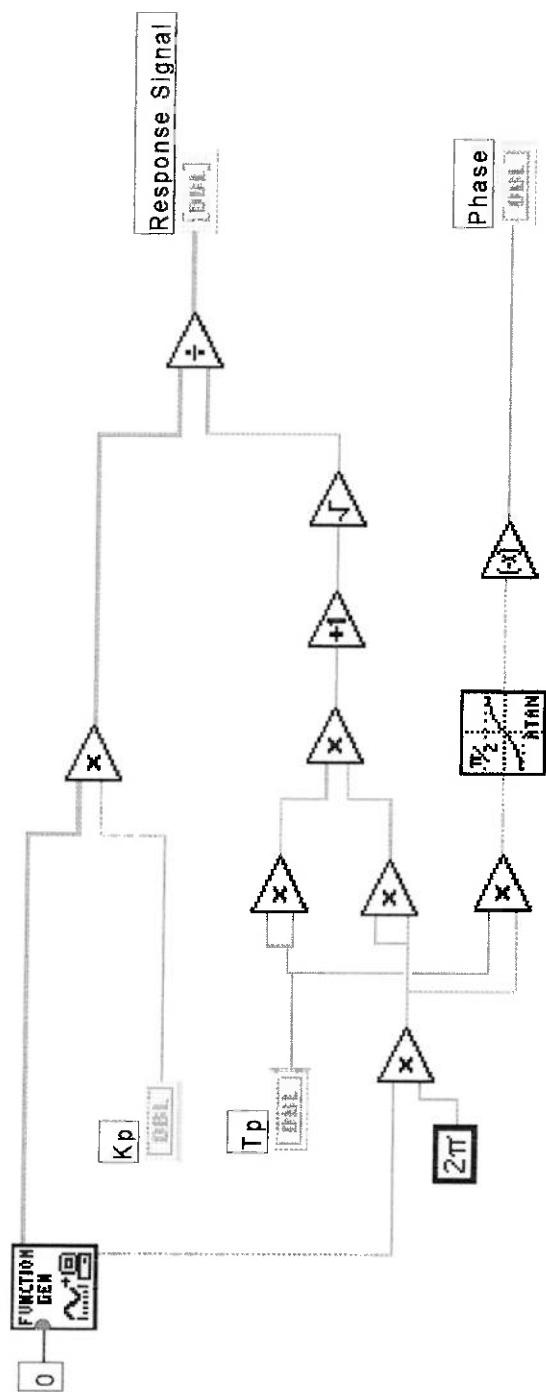




Block Diagram



Block Diagram



Block Diagram

