



UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
COORDENAÇÃO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

RELATÓRIO DE ESTÁGIO SUPERVISIONADO

Marcus André Ramos Berger

Campina Grande – Paraíba – Brasil
Junho de 2005



Biblioteca Setorial do CDSA. Fevereiro de 2021.

Sumé - PB

RELATÓRIO DE ESTÁGIO SUPERVISIONADO

Relatório apresentado à coordenação de estágios de Engenharia Elétrica da UFCG, como parte dos requisitos à obtenção de título de engenheiro eletricista.

Aluno:



Marcus André Ramos Berger

Orientador:



Péricles Rezende Barros

Local do Estágio: Laboratório de Instrumentação Eletrônica e Controle.

Período de Estágio: 28/02/2005 a 01/06/2005.

Campina Grande – Paraíba – Brasil

Junho de 2005

AGRADECIMENTOS

Em primeiro lugar a Deus pela força nos momentos mais difíceis, em segundo aos meus pais pelo amor e apoio irrestrito durante toda a minha vida.

Aos amigos que tanto nos ajudaram durante toda a graduação.

E também aos professores pela paciência, orientação e compreensão não só durante o período de estágio, mas durante toda a graduação. Em especial ao Professor Péricles Rezende Barros.

APRESENTAÇÃO

O presente relatório refere-se ao estágio supervisionado realizado pelo aluno do curso de graduação em Engenharia Elétrica da Universidade Federal de Campina Grande (UFCG), Marcus André Ramos Berger, no Laboratório de Instrumentação Eletrônica e Controle (LIEC), desta mesma universidade. O estágio teve vigência durante o período de 28/02/05 a 01/06/05 sob orientação e supervisão Professor Péricles Rezende Barros.

O estágio supervisionado teve carga horária de 120 horas, e é uma disciplina integrante da grade curricular do curso de Engenharia Elétrica, indispensável para obtenção do diploma de Engenheiro Eletricista.

SUMÁRIO

| | |
|---|----|
| 1. Introdução..... | 7 |
| 2. O Laboratório de Instrumentação Eletrônica e Controle..... | 8 |
| 3. Atividades Desenvolvidas..... | 9 |
| 3.1. Melhoria do Sistema Supervisório | 9 |
| 3.1.1. O Módulo Termoeletrico..... | 10 |
| 3.1.2. PLC5/15 e Programação Ladder..... | 11 |
| 3.1.3. Interface Homem Máquina..... | 15 |
| 3.1.4. Estudo e Configuração da Comunicação OPC..... | 17 |
| 3.2. Sintonia de Malha | 20 |
| 3.2.1. Técnicas de Sintonia em Malha Aberta..... | 21 |
| 3.2.2. Técnicas de Sintonia em Malha Fechada..... | 22 |
| 3.2.3. Software ProcessTune..... | 23 |
| 3.2.4. Software IdentAn..... | 25 |
| 3.2.5. Experimentos Realizados..... | 26 |
| 3.3. Testes com o Software SSP-Laplace..... | 29 |
| 3.4. Estudo da Especificação XML-OPC DA..... | 31 |
| 4. Conclusões..... | 32 |
| Referências Bibliográficas..... | 33 |

FIGURAS

| | |
|--|----|
| Figura 1. LIEC | 9 |
| Figura 2. Sistema | 10 |
| Figura 3. Modulo Termoeletrico. | 11 |
| Figura 4. CLP 5/15 à esquerda e módulo analógico à direita..... | 12 |
| Figura 5. Software RSLogix utilizado para programar em Ladder..... | 13 |
| Figura 6. Visualização dos valores nas posições de memória N10..... | 14 |
| Figura 7. Interface para Experimento de Margem de Fase..... | 16 |
| Figura 8. Interface para Experimento de Margem de Ganho. | 16 |
| Figura 9. Software RSLinx em operação comunicando-se através de OPC com o software OPCLink. | 17 |
| Figura 10. Experimento Relé Padrão, onde SP é o SetPoint ou referência, MV é a entrada processo e PV e saída da malha. | 21 |
| Figura 11. Experimento do Relé Padrão utilizado para obter a Margem de Ganho do sistema em malha fechada..... | 22 |
| Figura 12. Experimento de Ganho de Malha através do qual se obtêm a Margem de Fase do sistema em malha fechada..... | 22 |
| Figura 13. Software ProcessTune..... | 24 |
| Figura 14. Software IdentAn. | 25 |
| Figura 15. Interface mostrando resposta ao degrau em malha aberta. | 27 |
| Figura 16. Interface mostrando resposta ao degrau em malha fechada. | 27 |
| Figura 17. Interface durante um Experimento de Margem de Ganho..... | 28 |
| Figura 18. Interface durante um Experimento de Margem de Fase. | 28 |
| Figura 19. Software SSP-Laplace. | 30 |
| Figura 20. Exemplo de mensagem SOAP em OPC-XML. | 32 |

1. INTRODUÇÃO

Nos últimos dois séculos ocorreram profundas transformações nos modos de produção que levaram as conseqüentes mudanças no trabalho e mercado de trabalho. Estas transformações não representaram apenas mudanças na forma como produtos eram fabricados pelo homem, mas repercutiram em toda sociedade trazendo novos paradigmas como eficiência, qualidade e agilidade. O processo de globalização no século passado trouxe a competitividade e influência cultural a nível mundial para todas as empresas e nações.

Esta competitividade, agregada aos conceitos vigentes no trabalho, faz com que a exigência no que diz respeito à qualidade dos profissionais esperados pelo mercado seja cada vez maior. Ainda, exige-se que o profissional possua diversas outras aptidões além do conhecimento teórico obtido em sala de aula. Entre estas a habilidades está a capacidade de resolver problemas aplicando os conhecimentos na prática.

Dentro deste contexto, o estágio serve como uma oportunidade do aluno conhecer a prática, adquirindo maior segurança, aprendendo a conviver em uma ambiente de trabalho com outros profissionais e ainda se auto-avaliando no que diz respeito às suas aptidões e vocação além de ser uma oportunidade ímpar para o aprendiz.

Neste relatório serão descritas as atividades realizadas durante o período do estágio, a começar pelas atividades realizadas com o sistema SCADA acoplado à planta do módulo termoeletrico incluindo o desenvolvimento de novas interfaces, a configuração e utilização dos programas InTouch, OPCLink, RSLinx e RSLogix além do estudo do Padrão OPC. Depois serão apresentados os testes de sintonia de malha realizados e as técnicas e softwares utilizados. Por fim, serão descritos os estudos conduzidos sobre o padrão XML-OPC e outras tecnologias relacionadas que levaram a apresentação de alguns seminários durante o período de estágio.

2. O Laboratório de Instrumentação Eletrônica e Controle.

O Laboratório de Instrumentação Eletrônica e Controle é um dos laboratórios de pesquisa do Departamento de Engenharia Elétrica da Universidade Federal de Campina Grande. Por este já passaram centenas de pesquisadores e alunos de pós-graduação. O laboratório atua em diversas linhas de pesquisa entre elas: instrumentação eletrônica, automação industrial, identificação de sistemas e controle de processos. Este conta com modernos equipamentos como microcomputadores, osciloscópios, Controladores Lógicos Programáveis Siemens e Allen-Bradley, redes industriais como GPIB, Profibus e DeviceNet, além de alguns processos em escala laboratorial construídos pelo pessoal do próprio laboratório como, por exemplo, Módulos Termoelétricos, Tanques em Batelada e Esteiras industriais que conectados aos CLP's ou aos microcontroladores são utilizados para atividades de pesquisa na área de controle.

Ainda, o laboratório dispõe de um corpo docente altamente qualificado composto por doutores que trabalham tanto em projetos de pesquisa como em projetos de desenvolvimento junto com alunos de mestrado e doutorado. Diversos projetos já foram conduzidos em parcerias com empresas na área de instrumentação eletrônica e controle. Vale a pena salientar os projetos em parceria com Petrobrás e a parceria para o desenvolvimento de equipamentos de telemetria para o Centro de Lançamento de Alcântara, por exemplo.



Figura 1. LIEC

3. ATIVIDADES REALIZADAS

Durante o período de estágio foram realizadas diversas atividades inseridas dentro do contexto dos projetos de pesquisa e desenvolvimento realizados no laboratório.

3.1. Melhoria do Sistema SCADA.

No Laboratório de Instrumentação Eletrônica e Controle existem diversas plantas em escala laboratorial utilizadas para pesquisa e testes. Uma delas é composta por um módulo termoeletrônico que é controlado por um Controlador Lógico Programável (CLP) Allen Bradley modelo 5/15. Este CLP está conectado a um dos microcomputadores do laboratório onde está instalado o sistema SCADA (*Supervisory Control and Data Acquisition* – Controle Supervisório e Aquisição de Dados) responsável pela supervisão e controle desse sistema e que opera em sistema Windows. Foram desenvolvidas diversas atividades nesta plataforma como a construção de novas interfaces para a realização de experimentos para Sintonia de Controladores PID, configuração de comunicação OPC (*OLE for Process Control*, onde *OLE* significa *Object Linking and Embedding*) entre o software supervisório e o software de

comunicação com o CLP e o desenvolvimento de novos programas para o CLP utilizando linguagem Ladder. Estas atividades serão descritas a seguir.

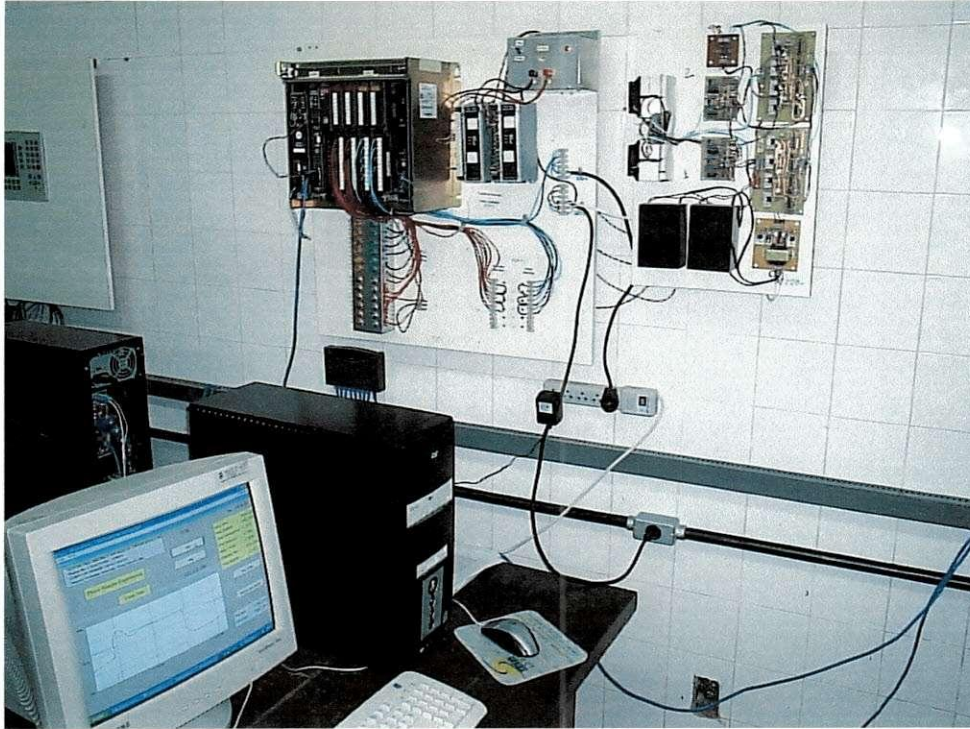


Figura 2. Sistema

3.1.1. O Módulo Termoelétrico

Primeiramente, se faz necessária uma breve descrição do módulo Termoelétrico existente no laboratório. Este módulo é composto por duas placas semicondutoras que agem como bombas de calor nas duas secções de uma placa metálica que por sua vez age como carga de calor. Nas placas semicondutoras ocorre o Efeito Peltier, ou seja, com o módulo tem-se um processo térmico de duas entradas e duas saídas (TITO) com alto acoplamento onde através do controle da corrente que atravessa cada placa semicondutora é possível controlar a temperatura desta. A temperatura na placa pode variar entre 10°C e 70°C quando operando em temperatura ambiente de 24°C.

O módulo termoelétrico possui alguma instrumentação embarcada. Na saída do sistema onde é obtida a medição de temperatura realizada por sensores LM35 existem apenas amplificadores de sinal sem nenhum filtro. Como a planta serve para teste das técnicas de

sintonia, a presença de ruído serve para validar a robustez das técnicas. Para aplicação de corrente no módulo é utilizada modulação PWM (Modulação por Largura de Pulso). A largura dos pulsos é controlada pela tensão presente no resistor conectado a saída do módulo analógico do CLP já que a saída deste é uma saída de corrente entre 0-20mA. No caso, a variação de tensão para o controle da largura de pulso do circuito PWM está entre 0 e 10V.

Ainda, o módulo termoeletrico resulta em um processo complexo com grandes não-linearidades o que o torna interessante para teste de técnicas de sintonia de controladores.

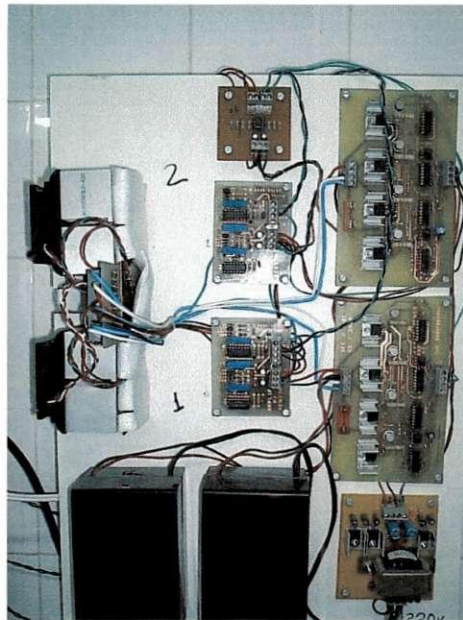


Figura 3. Modulo Termoeletrico.

3.1.2. CLP 5/15 e Programação Ladder

O módulo termoeletrico tem suas duas entradas e duas saídas conectadas ao CLP através de seu módulo analógico 1791-NDC. Assim, através do CLP é possível obter os valores de temperatura nas extremidades das placas como também controlar a corrente elétrica que é aplicada. Este módulo opera com corrente elétrica entre 0 e 20mA na saída e é alimentado com 24Vdc. Este possui 2 saídas e 4 entradas. No sistema existente no laboratório, atualmente estão sendo utilizadas apenas duas entradas e duas saídas. Para conversão dos sinais são utilizados 12 bits.

O CLP 5/15 é um controlador muito versátil e muito popular na indústria. Sua memória central de leitura e escrita CMOS/RAM armazena instruções do programa de aplicação em arquivos de programa, e valores numéricos e estado dos dispositivos de entrada e saída em arquivos de dados. O programa de aplicação é um conjunto de instruções em uma determinada ordem que descreve as operações a serem executadas e as condições de operação destas instruções. Durante a execução do programa de aplicação, o controlador monitora continuamente o estado dos dispositivos de entrada e, baseado nas instruções desse programa, energiza ou desenergiza os dispositivos de saída. O CLP ocupa a ranhura da extrema esquerda do chassi 1771 de entrada e saída. O 5/15 possui 6k palavras de memória que pode ser expandida para 13k e este pode endereçar 1024 pontos de entrada e saída. Este controlador possui um vasto conjunto de instruções que incluem controle PID, conversão de dados, diagnóstico, escrita/leitura direta de módulos analógicos e transferência em bloco de dados.

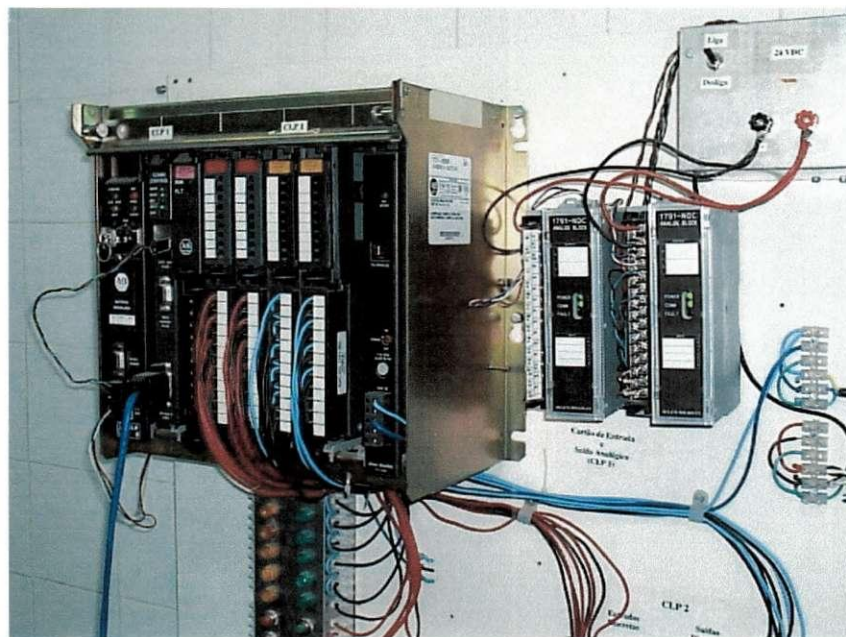


Figura 4. CLP 5/15 à esquerda e módulo analógico à direita.

O CLP comunica-se com o microcomputador através de protocolo RS-232. No computador existe um software específico do fabricante do CLP Rockwell, o RSLinx, que gerencia a comunicação com o CLP (dos drivers) e opera como um servidor para qualquer outro programa que comunique-se através de DDE ou OPC. Todo programa que acesse os dados do

CLP deve utilizar esse servidor, inclusive o software onde é realizada a programação do CLP, o RSLogix, também da Rockwell.

A programação do CLP é realizada no RSLogix utilizando a Linguagem Ladder que é baseada em diagramas de comandos elétricos.

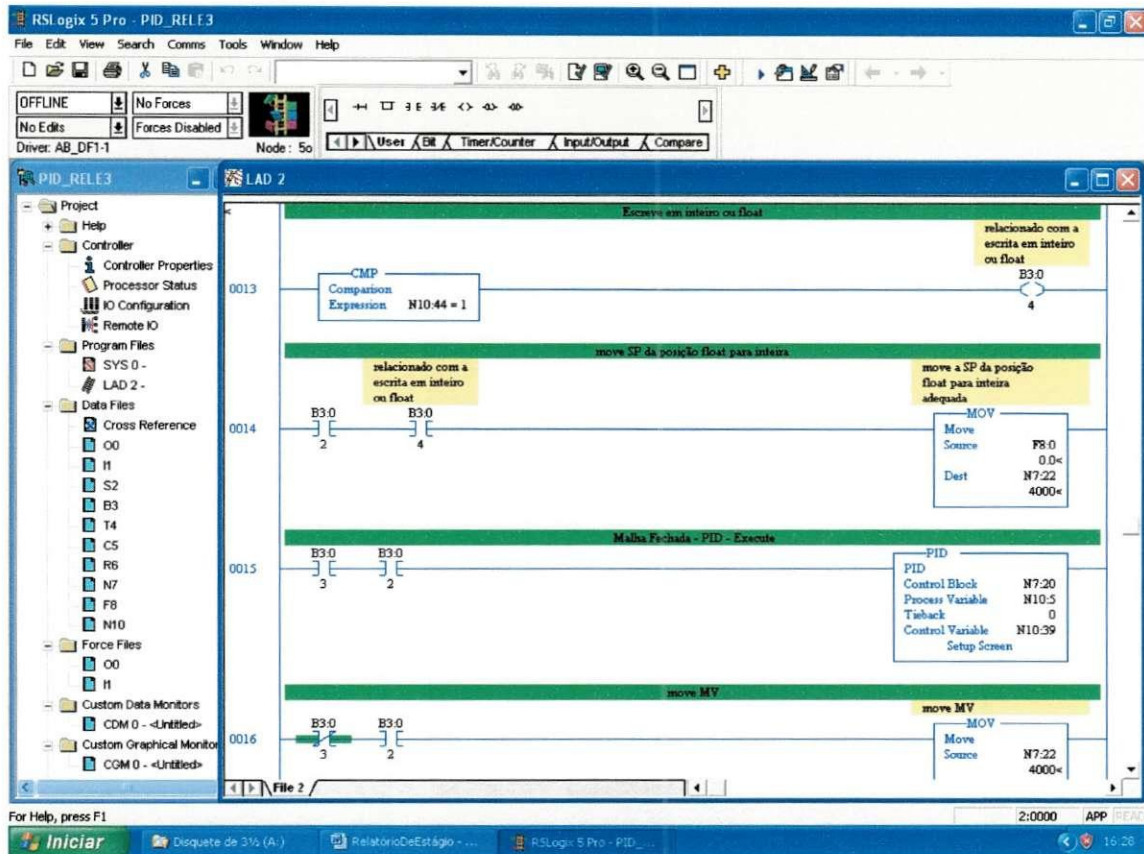


Figura 5. Software RSLogix utilizado para programar em Ladder.

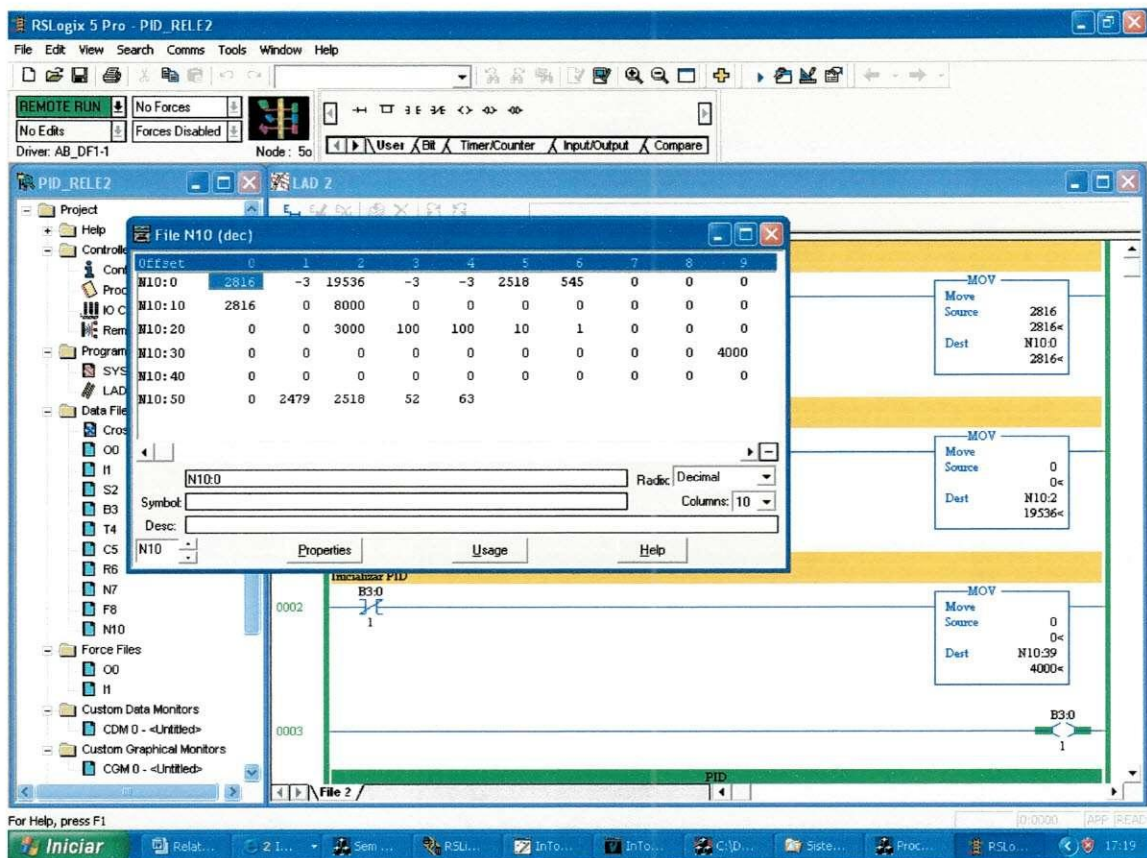


Figura 6. Visualização dos valores nas posições de memória N10.

Uma das atividades desenvolvidas durante o estágio foi a modificação do software que operava no CLP para que esta pudesse agregar mais algumas facilidades. O novo programa desenvolvido, pode abrir e fechar a malha de controle para operação com o PID e ainda possibilita a leitura dos dados relativos as variáveis do processo tanto em posições de memória de palavras inteiras (posições de memória N) como de palavras de ponto flutuante (posições de memória F) necessárias para comunicação com o software SSP-Laplace como será explicado mais adiante.

3.1.3. Interface Homem Máquina

Para supervisão do sistema existe no microcomputador um software SCADA que é responsável pela monitoração do sistema. Através desse software, os dados presentes em qualquer posição de memória do CLP podem ser acessados e alterados. No LIEC utiliza-se o InTouch como software SCADA devido a sua facilidade de operação e versatilidade. Este software é largamente utilizado em diversas indústrias como, por exemplo, na Petrobrás.

O InTouch comunica-se através de DDE ou OPC (neste caso utilizando outro programa chamado OPCLink) tanto com o servidor RSLinx do CLP como também com outros programas servidores. No caso deste sistema, o InTouch comunica-se utilizando OPC como o RSLinx e DDE com o software ProcessTune.

As atividades realizadas no laboratório compreenderam além da configuração da comunicação do SCADA utilizando OPC com o servidor do CLP, o desenvolvimento de novas interfaces homem-máquina específicas para realização de experimentos à relé. No caso, os experimentos de Margem de Fase e Margem de Ganho. Estas telas são apresentadas nas figuras 7 e 8 e estes experimentos são discutidos nas secções 3.2.1 e 3.2.2.

A construção destas novas interfaces teve como objetivo criar interfaces adequadas para os experimentos que deveriam ser realizados, com uma maior usabilidade e facilidade de operação já que nas telas estariam apenas as informações estritamente necessárias para realização de cada um dos dois experimentos de relé. Ainda estas telas serviram como base para o engenheiro da Petrobrás responsável pela construção da interface em outro software SCADA, o SSP-Laplace, como será explicado na secção 3.3.

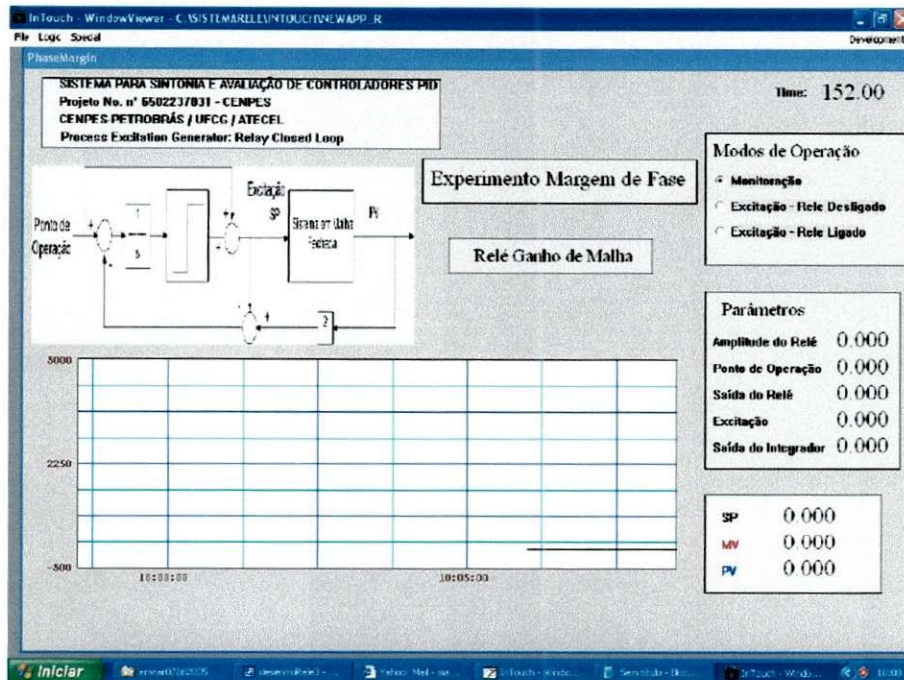


Figura 7. Interface para Experimento de Margem de Fase.

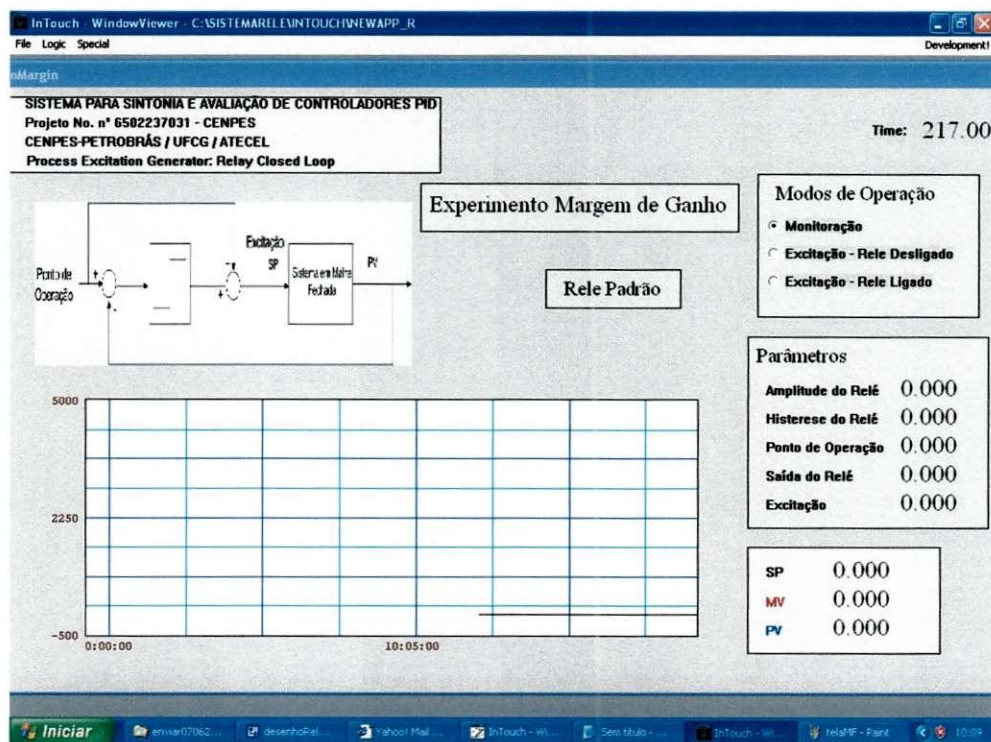


Figura 8. Interface para Experimento de Margem de Ganho.

3.1.4. Estudo e Configuração da Comunicação OPC

Outra tarefa realizada no laboratório foi a configuração da comunicação OPC entre os softwares que antes era realizada utilizando DDE. Além disso, foi realizado um estudo sobre esta tecnologia que é relativamente nova e de outras tecnologias relacionadas.

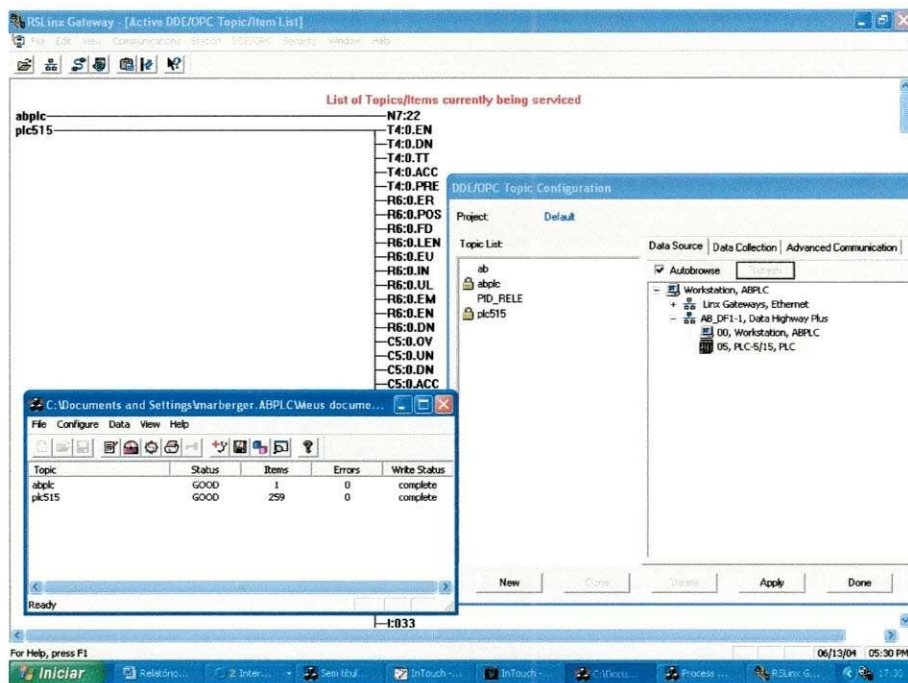


Figura 9. Software RSLinx em operação comunicando-se através de OPC com o software OPCLink.

O padrão OPC atualmente é uma tendência na comunicação entre processos (programas de computador) em aplicações de automação industrial, pois apresenta inúmeras vantagens em relação às outras tecnologias utilizadas anteriormente, além de, por ser um padrão que define um conjunto de especificações, facilita a interoperabilidade entre diferentes sistemas de software.

Este padrão foi definido e adotado por um consórcio de várias empresas vinculadas à organização OPC Foundation. A priori é um protocolo de comunicação interprocessos operando no sistema Windows da Microsoft que atende às especificações das aplicações de automação

industrial. Este estabelece algumas interfaces comuns que programas devem implementar para permitir a comunicação padronizada com outros programas.

A comunicação de dados entre sistemas de controle tradicionais é normalmente baseada nos diferentes protocolos utilizados por cada fabricante específico de equipamentos de automação dependendo da rede de comunicação utilizada, por exemplo, Profibus, ControlNet, Foundation Fieldbus, entre outros. Assim, é necessário um driver (um programa que implementa os protocolos de comunicação necessários para acessar o dispositivo) para realizar a comunicação com os dispositivos de um determinado fabricante. Por não haver um padrão de comunicação entre as diversas aplicações, cada sistema que necessitasse acessar um driver específico de um fabricante deveria implementar um programa especificamente construído para comunicar-se com aquele driver já que não existia um padrão para comunicação.

O padrão OPC surgiu como uma forma de estabelecer regras de comunicação entre aplicações. Portanto, não elimina a necessidade dos drivers de comunicação que estão relacionados a cada fabricante específico dos dispositivos de campo. O padrão define o tipo de métodos e tipos de dados que poderão ser utilizados entre a aplicação cliente e a aplicação servidor, ou seja, a interface de comunicação entre os aplicativos.

A tecnologia é baseada em OLE (*Object Linking and Embedding*) que por sua vez é baseada em COM (*Component Object Model*, ou Modelo de Objeto Componente). O objeto COM é a implementação das diversas interfaces existentes para um objeto OPC. Este modelo define uma arquitetura cliente-servidor onde cada fonte de informação que pode corresponder a um ou mais dispositivos ou aplicativos é representada por servidor. Nesta arquitetura, o objeto COM será um servidor e a aplicação que se comunica com este, o cliente. Esta arquitetura provê uma maior flexibilidade aos sistemas de software já que a programação baseada em componentes de software facilita a reutilização de código. Assim, OPC permite que um servidor se comunique com vários clientes e vice-versa.

Existem dois tipos de interfaces: a *custom* e a *automation*. A primeira é utilizada por aplicações clientes desenvolvidas em linguagens que suportam chamadas de funções por ponteiros como C, C++ e Delphi. A segunda por linguagens que não possuem este recurso como Visual Basic, por exemplo. Neste tipo de interface, os métodos são chamados por nomes e não por ponteiros. As diversas interfaces são definidas por especificações, as principais são *OPC Common Definitions and Interfaces* (que define as funcionalidades básicas para as outras especificações), *OPC Data Access* (que define a interface para leitura e escrita de dados em tempo real) e *OPC Alarms and Events* (que define uma interface para monitoração de eventos).

O padrão define uma abstração dos dados existentes no servidor na forma de uma hierarquia de objetos de software. O primeiro objeto é o *OPCServer* que representa o próprio servidor OPC; depois existem os *OPCGroup* que além de definir como os dados são organizados dentro do servidor pelo cliente, define importantes parâmetros para transferência de dados que são utilizados como, por exemplo, taxa de atualização, estado ativo/inativo, banda morta e leitura assíncrona/síncrona; o último nível é composto pelos objetos *OPCItems* que são a representação dos dados existentes no servidor. Por exemplo, um dado pode ser representado por mais de um item e cada item desse pode estar em um grupo diferente como uma representação diferente do mesmo dado para cada cliente diferente. Isto proporciona uma grande versatilidade já que a abstração dos dados pode ser realizada da maneira mais conveniente para o cliente.

Assim, o padrão OPC proporciona algumas vantagens em relação à tecnologia anterior:

- Padronização das interfaces de comunicação entre os servidores e clientes de dados em tempo real (OPC DA), facilitando a integração e manutenção de sistemas;
- Eliminação da necessidade de drivers de comunicação específicos (proprietários);
- Melhoria do desempenho e otimização da comunicação entre dispositivos de automação;
- Interoperabilidade entre sistemas de diversos fabricantes;
- Integração entre sistemas ERP (Planejamento de Recursos do Empreendimento), MES (Sistemas de Execução de Manufatura) e aplicativos Windows;
- Redução de custos e tempo para desenvolvimento de interfaces e drivers de comunicação, com conseqüente redução do custo de integração de sistemas;
- Facilidade de desenvolvimento e manutenção de sistemas e produtos para comunicação em tempo real;
- Facilidade de treinamento.

O trabalho realizado no LIEC constitui após o estudo inicial dos conceitos principais do Padrão OPC, o estudo dos manuais dos softwares utilizados no laboratório e a configuração destes para operarem utilizando OPC.

3.2. Sintonia de Malha

O controlador Proporcional-Integral-Derivativo (PID) é certamente o algoritmo de controle mais encontrado na indústria, em especial na química e petroquímica. Pesquisas revelam que em mais de 90% das malhas de controle da indústria são utilizados estes controladores. Geralmente, na maioria das indústrias a sintonia destes controladores não é automatizada e requer dedicação de técnicos durante um tempo considerável. Ainda, como a determinação manual de parâmetros de controle PID é uma tarefa árdua e dispendiosa pela forte interação existente entre estes parâmetros, muitas malhas não são sintonizadas para um desempenho ótimo, prejudicando a eficiência dos processos que controlam. Processos operando sob controladores bem sintonizados resultam em produtos com melhor qualidade e menor custo, pois ocorrerá menor variação do produto em relação às especificações de produção e menor consumo de energia e de insumos, levando à menores perdas econômicas.

As técnicas utilizando realimentação à relé para sintonia de controladores têm grande aceitação na indústria, pois utilizam experimentos em malha fechada, os quais mantêm o processo sob controle e operando próximo o ponto de operação (setpoint), o que é preferível em relação a testes em malha aberta como, por exemplo, testes de resposta ao degrau os quais ainda são largamente utilizados na indústria. Ainda, o emprego da realimentação a relé traz outras vantagens, em processos lineares estáveis geralmente garante uma oscilação estacionária sustentável (ciclo limite) e, além disso, o uso destas técnicas não requer nenhuma informação anterior sobre as constantes do processo para garantir uma boa escolha do período de amostragem.

Diante destas vantagens o uso de experimentos utilizando relé em técnicas de sintonia de controladores PID é algo valioso para indústria. Durante o período de estágio foram desenvolvidas atividades neste contexto, em especial, realizando testes com os softwares IdentAn e ProcessTune desenvolvidos em projeto conduzido pelo professor Péricles Rezende Barros no LIEC em convênio com a Petrobrás. Ainda foram estudadas algumas das técnicas utilizadas para sintonia utilizando relé, inclusive técnicas que foram resultado de tese de doutorado desenvolvida no laboratório. O resultado deste trabalho foi a publicação de um artigo que contou ainda com a participação do estagiário George Acioli Jr submetido ao IECON 2005.

3.2.1. Técnicas de Sintonia em Malha Aberta.

Aqui são denominadas técnicas em malha aberta as técnicas utilizadas para o projeto inicial do controlador, ou seja, existe um processo e se quer projetar um controlador PID que garanta algumas especificações em relação à resposta do sistema em malha fechada, overshoot e amortecimento, por exemplo. Estas técnicas utilizam realimentação à rele e por isso os testes são conduzidos em malha fechada. São chamadas técnicas de malha fechada, as técnicas onde já existe um controlador e se quer sintonizá-lo para novas especificações em especial no que se refere a novas margens de ganho e de fase.

Foram estudadas e testadas as seguintes técnicas de sintonia:

- Método da Resposta em Frequência de Ziegler-Nichols;
- Método de Clair;
- Método de Tyreus-Luyben.

Todos os métodos são baseados em experimentos utilizando a estrutura do Relé Padrão como mostrado na figura 6.

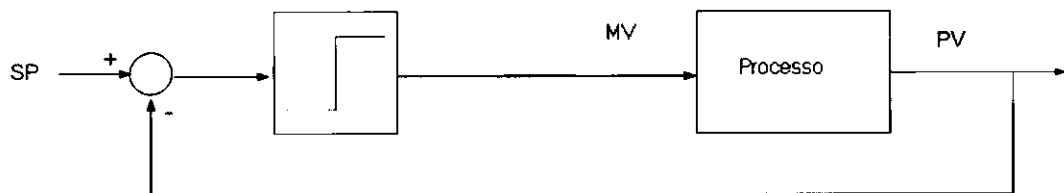


Figura 10. Experimento Relé Padrão, onde SP é o SetPoint ou referência, MV é a entrada do processo e PV e saída da malha.

Para aplicação destas técnicas eram obtidos a frequência crítica e o ganho crítico do processo através dos testes utilizando o Relé Padrão e a partir destes parâmetros as técnicas de sintonia sugeriam os parâmetros do controlador PID. Para sintonia utilizando estas técnicas foram utilizados os softwares ProcessTune e IdentAn como será explicado nas secções 3.2.3 e 3.2.4.

3.2.2. Técnicas de Sintonia em Malha Fechada

São denominadas técnicas de sintonia em malha fechada as técnicas aplicadas ao sistema em malha fechada onde já existe um controlador em operação, ou seja, já existe um controlador e deseja-se analisar sua condição de operação e alterar seus parâmetros para alcançar uma sintonia mais apropriada em especial no que diz respeito a novas margens de fase e de ganho.

A técnica estudada e testada foi desenvolvida durante tese de doutorado do aluno Gustavo Henrique Machado de Arruda orientado pelo Professor Péricles Rezende Barros no LIEC. Esta é uma técnica interativa que utiliza os resultados obtidos em experimentos de Relé Padrão e experimentos de Ganho de Malha para iterativamente obter novos parâmetros do controlador PID de forma a satisfazer novas especificações de Margem de Fase e Margem de Ganho para o controlador. A Margem de Fase e a Margem de Ganho servem como uma indicação da estabilidade relativa do sistema.

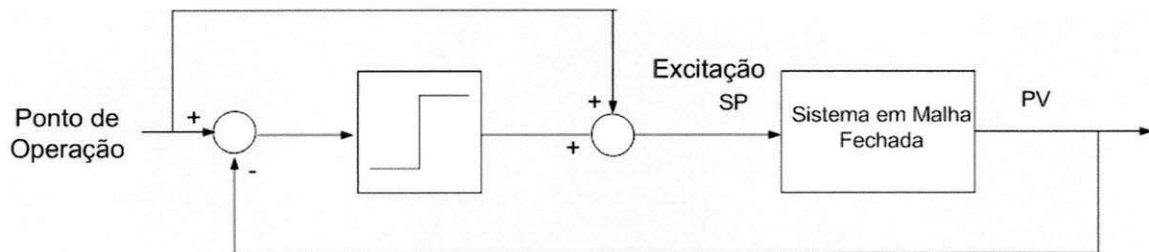


Figura 11. Experimento do Relé Padrão utilizado para obter a Margem de Ganho do sistema em malha fechada.

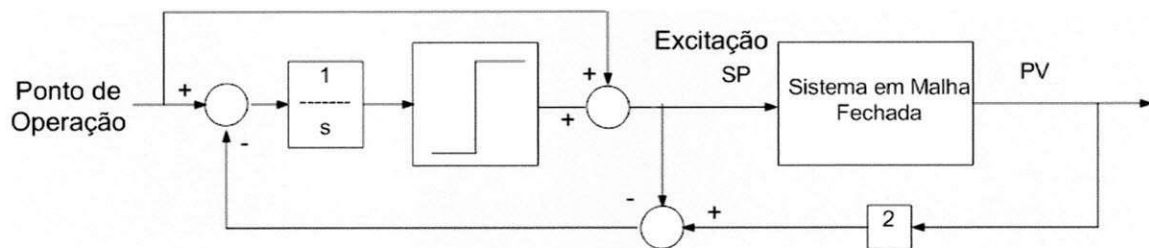


Figura 12. Experimento de Ganho de Malha através do qual se obtêm a Margem de Fase do sistema em malha fechada.

O procedimento para utilização da técnica consiste em inicialmente realizar um experimento com Relé Padrão ou de Ganho de Malha e analisar respectivamente a Margem de Ganho ou a Margem de Fase do sistema em malha fechada fornecida. Daí, estes experimentos serem chamados também de Experimento de Margem de Ganho e Experimento de Margem de Fase respectivamente para facilitar a identificação. Depois de realizado um destes experimentos obtêm-se uma das margens e então são determinados novos parâmetros do PID para uma nova especificação de margem se a margem de fase ou de ganho do sistema não é satisfatória. Uma nova margem de ganho acarreta mudança no ganho do controlador e uma nova margem de fase mudança no termo integrativo. O novo controlador é então sintonizado para os novos parâmetros e novamente avaliado, se a sua margem de ganho ou de fase ainda não é adequada este é novamente sintonizado.

As margens de ganho e da fase adequadas dependem dos processos e da resposta que se deseja da malha fechada, uma resposta mais agressiva ou mais conservativa, por exemplo. Geralmente, na indústria procura-se uma margem de ganho entre 2.5 e 3 e uma margem de fase por volta de 60° para garantir um desempenho satisfatório o que é um projeto razoavelmente conservativo. Os testes utilizando esta técnica foram realizados utilizando os softwares ProcessTune e IdentAn.

3.2.3. Software ProcessTune

Este software foi desenvolvido em projeto em conjunto com a Petrobrás e durante o estágio foi possível realizar alguns testes com este na planta existente no laboratório. Ainda foi possível acompanhar a visita do engenheiro Rodrigo Cesar d'Albriex de Carvalho do Cenpes/Petrobrás com intuito de migrar o algoritmo do ProcessTune para o software SSP-Laplace.

São implementados no ProcessTune algoritmos de diversas técnicas de relé: Relé Padrão, Relé Padrão com Integrador, Relé Generalizado e Ganho de Malha. No entanto, o trabalho realizado concentrou-se nas técnicas de Relé Padrão e Ganho de Malha já que estas obtiveram melhores resultados quando submetidas a ruído e proporcionavam as informações para efetuar a sintonia em malha fechada utilizando a Margem de Ganho e de Fase obtidas.

Este software trabalha on-line junto com o SCADA de forma a receber dados do processo vindos do CLP através deste, executar os algoritmos e enviar os resultados de volta

para enfim serem aplicados ao processo através da atuação do CLP. Entre outras funcionalidades, o ProcessTune pode armazenar os dados processados através de arquivos para posteriormente serem utilizados por outros programas como o IdentAn.

A operação do sistema ocorre da seguinte maneira: o ProcessTune comunica-se com o InTouch através de DDE e há a transferência das variáveis necessárias para execução dos algoritmos de relé como a variável do processo, a amplitude do relé especificado pelo usuário no InTouch, a histerese do relé e o ponto de operação do sistema. A configuração de parâmetros pelo usuário é realizada apenas através do InTouch. O ProcessTune retorna a saída do relé que é o setpoint do sistema em malha fechada além do valor do integrador se a técnica possui um integrador. Este valor é importante em termos da monitoração da dinâmica do sistema. Os parâmetros do InTouch, então, são enviados para o programa servidor que os enviará para o CLP.

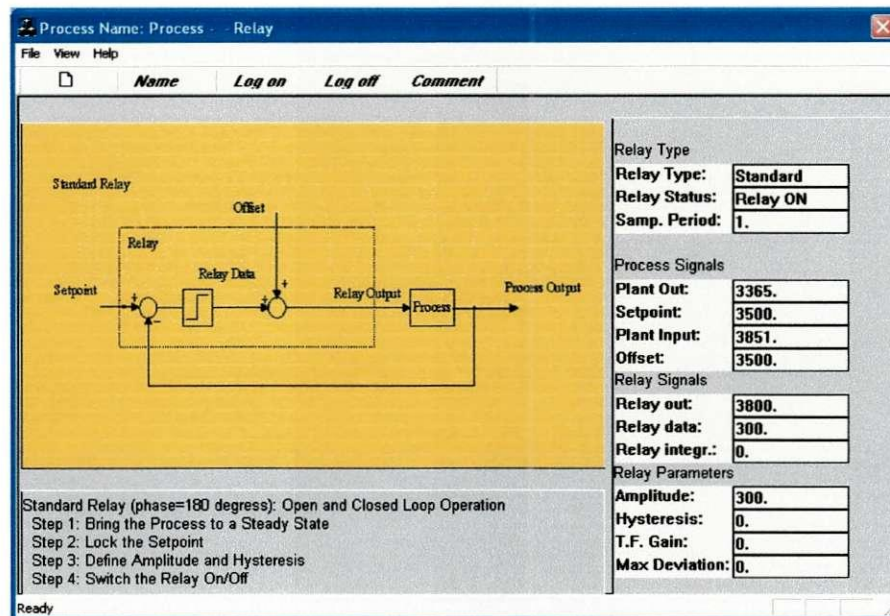


Figura 13. Software ProcessTune.

3.2.4. O Software IdentAn.

Este software foi desenvolvido juntamente com o ProcessTune, entretanto para trabalhar off-line, com objetivo de avaliar, identificar e sintonizar malhas de controle. Os dados obtidos através dos testes utilizando o software ProcessTune são processados por este programa que possui uma grande interatividade com o usuário. Através do IdentAn pode-se então identificar diversos modelos inclusive de primeira ordem com atraso (FOPDT), avaliar diversas característica dos sinais obtidos do processo como dados estatísticos, por exemplo, projetar controladores, sintonizar controladores já operacionais nas malhas para novas especificações de margem de fase e ganho, por exemplo, entre outras funcionalidades.

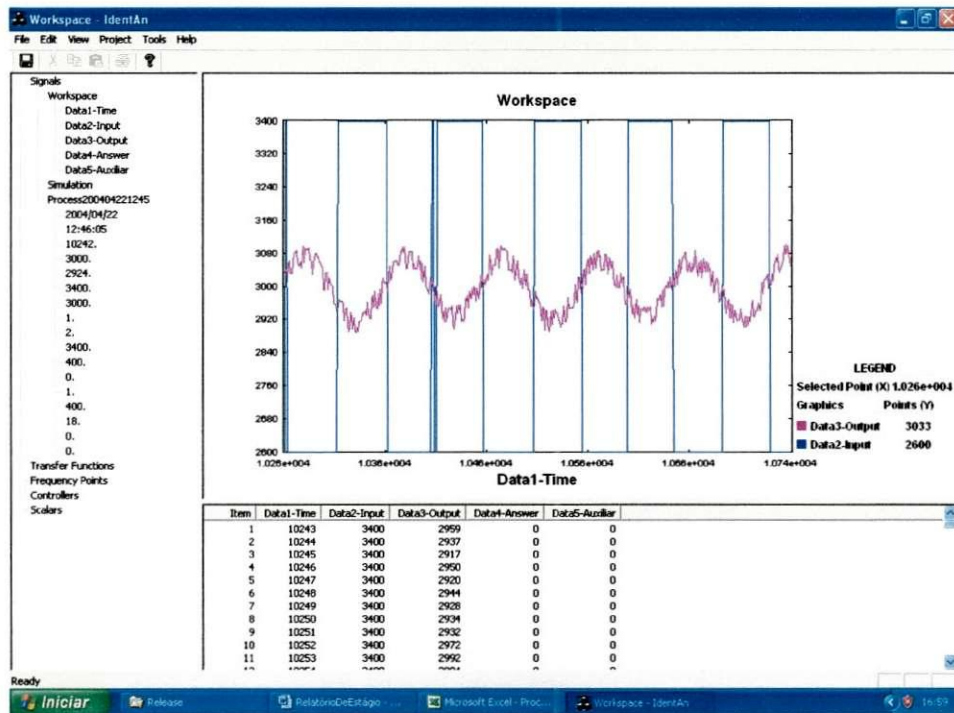


Figura 14. Software IdentAn.

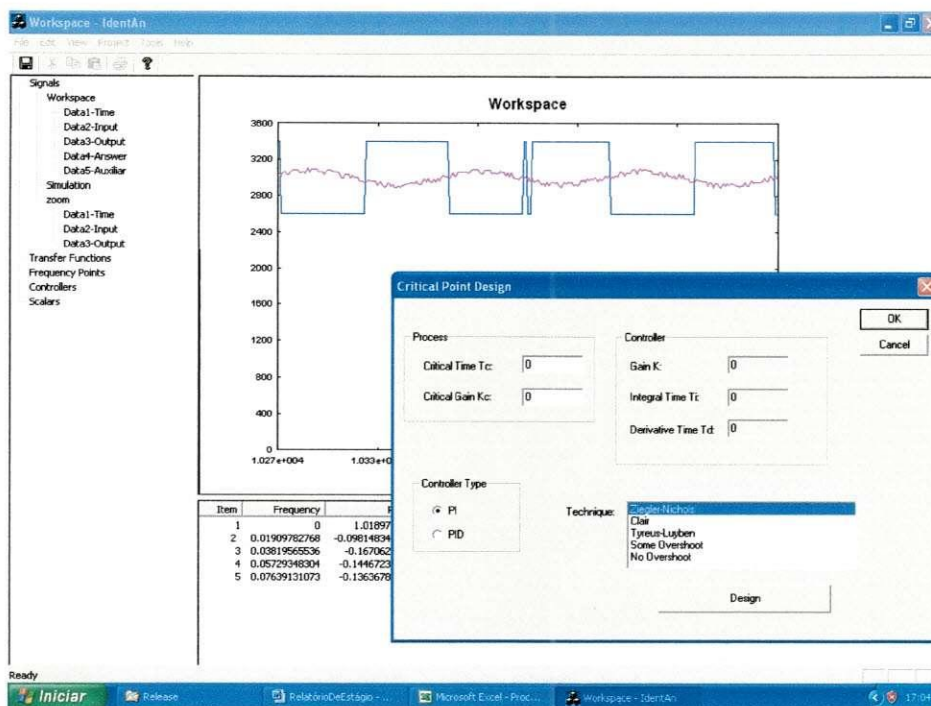


Figura 9. Sintonia de Controlador utilizando o Software IdentAn.

3.2.5. Experimentos Realizados

Diversos experimentos utilizando os softwares ProcessTune e IdentAn foram realizados com dois objetivos. O primeiro, testar o funcionamento dos softwares à procura de “bugs”, avaliando sua usabilidade e suas características de utilização, de forma a sugerir melhorias e correções futuras além de sua documentação de ajuda ao usuário. Em segundo lugar avaliar as técnicas de sintonia em um processo em escala laboratorial, em um ambiente controlado. Os experimentos realizados resultaram em um artigo que foi enviado para o Congresso do IECON2005. As figuras a seguir mostram alguns exemplos de resultados obtidos durante a realização dos testes.

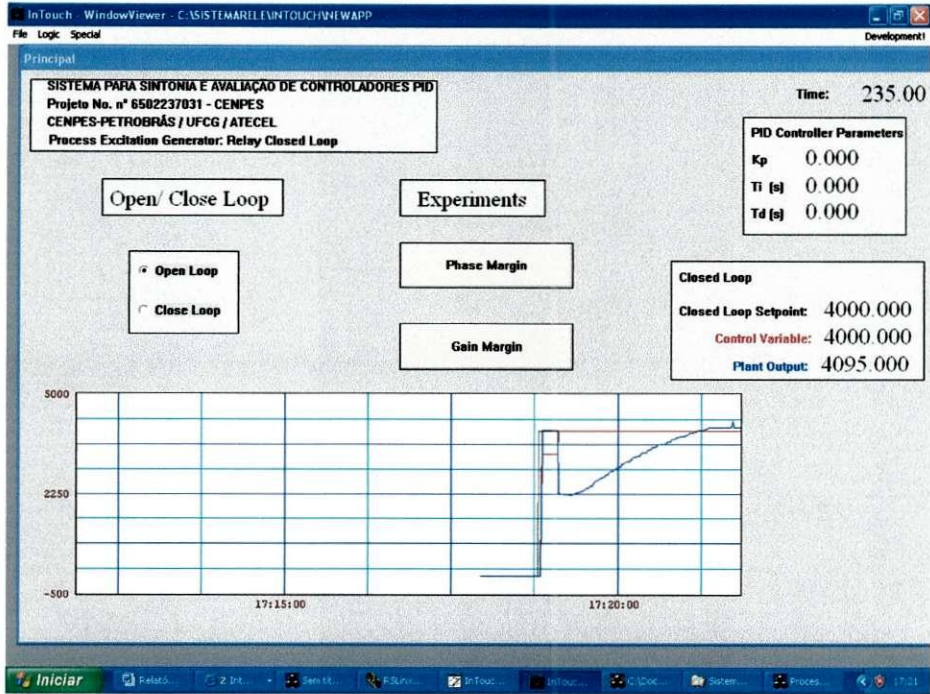


Figura 15. Interface mostrando resposta ao degrau em malha aberta.

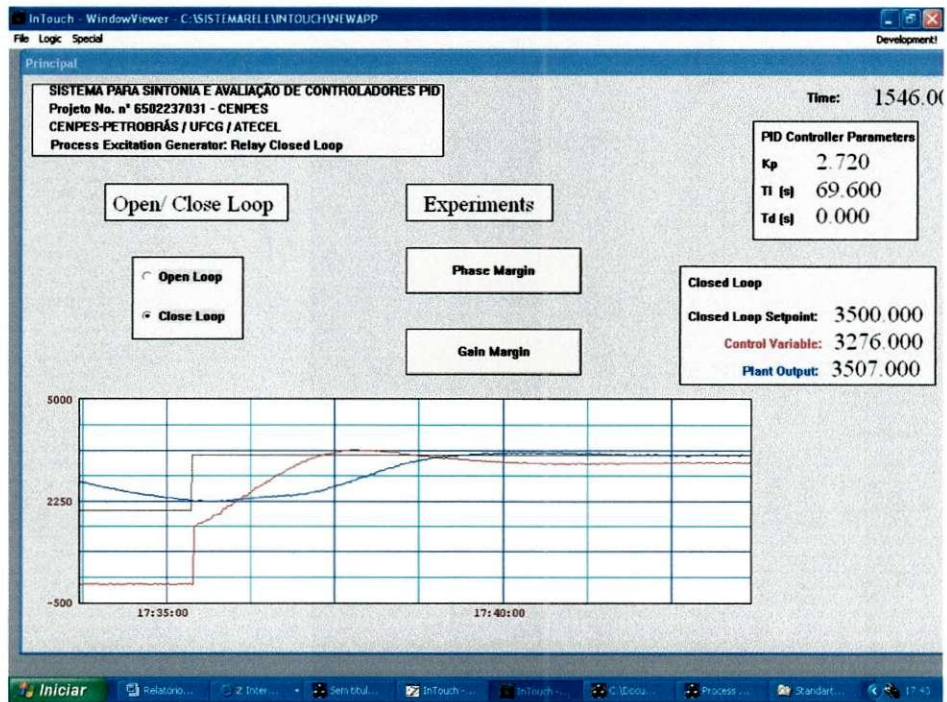


Figura 16. Interface mostrando resposta ao degrau em malha fechada.

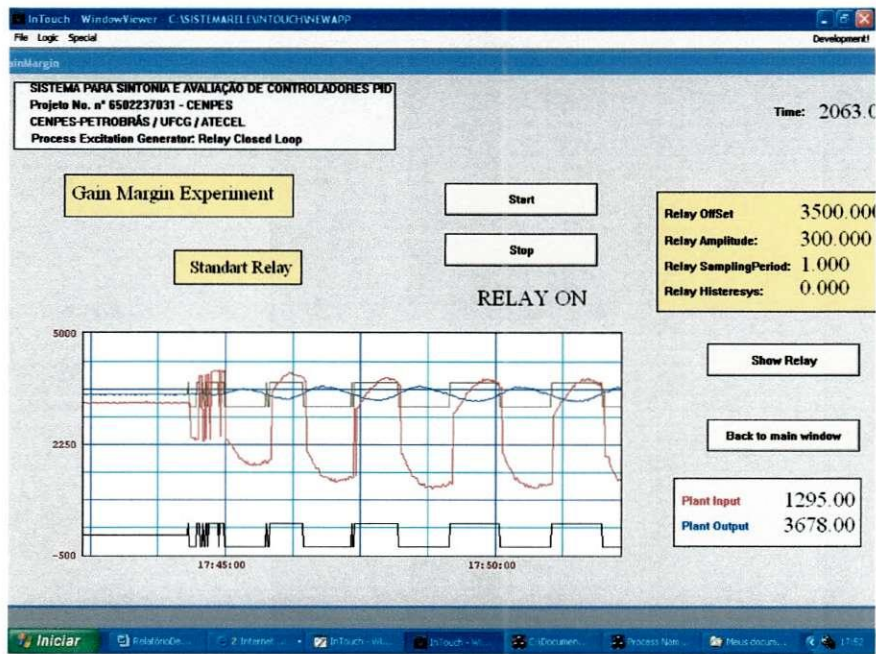


Figura 17. Interface durante um Experimento de Margem de Ganho.

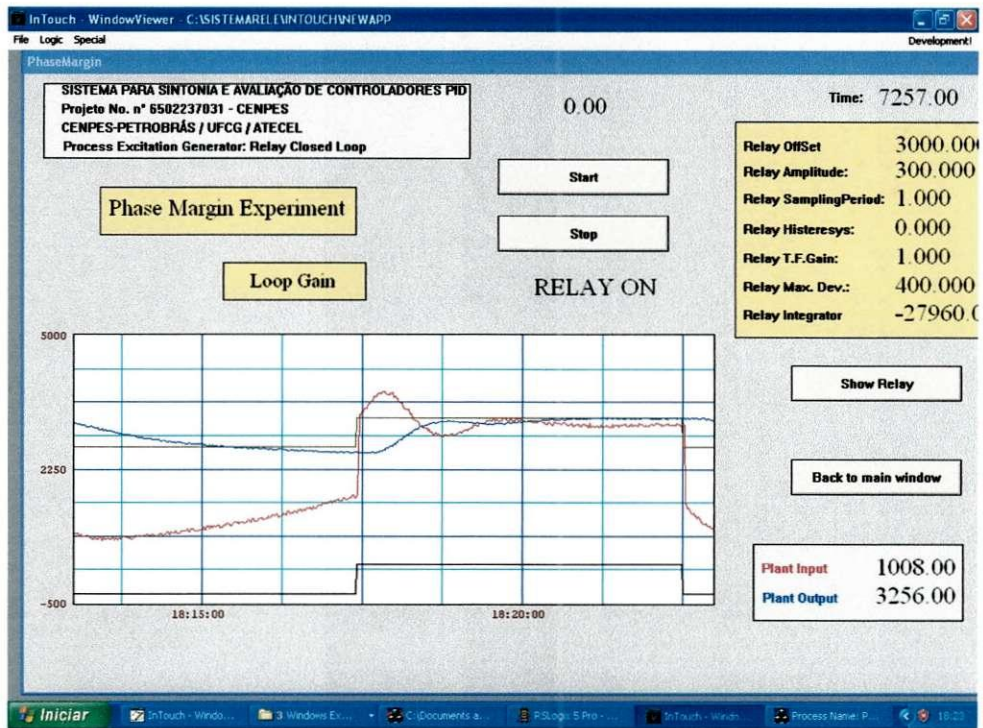


Figura 18. Interface durante um Experimento de Margem de Fase.

Nas figuras 15, 16, 17 e 18 é possível observar a execução de alguns testes com o sistema. Na primeira, um teste da resposta ao degrau em malha aberta, na segunda um teste de resposta ao degrau em malha fechada, na terceira um teste de margem de ganho e na quarta um experimento de margem de fase.

Estas telas apresentadas também foram construídas durante as atividades de estágio como versão intermediária, as versões finais são apresentadas nas figuras 5 e 6.

Vale a pena realizar um comentário sobre a figura 17 onde é possível observar a forte não-linearidade do processo onde o sinal de entrada do processo é muito assimétrico se comparado ao sinal de saída.

Outros resultados dos testes podem ser vistos nos artigos enviados para o IECON2005 que se encontra em anexo.

3.3. Testes com o Software SSP-Laplace

Durante visita do engenheiro da Petrobrás foi possível entrar em contato e realizar alguns testes com o software SSP-Laplace que está sendo desenvolvido pela empresa InduSoft em parceria com o Cenpes/Petrobrás e deverá ser o novo sistema SCADA a equipar diversas unidades da empresa. Este software tem como função principal facilitar o desenvolvimento de aplicações de controle e de engenharia objetivando melhorar o acompanhamento e controle das unidades de processo. Ainda, o software incorpora os principais algoritmos disponíveis na Petrobrás nas áreas de inferências e de controle de forma que o usuário do programa necessita apenas configurar a aplicação escolhida sem a necessidade de programação, o que facilita a integração de novas aplicações na planta.

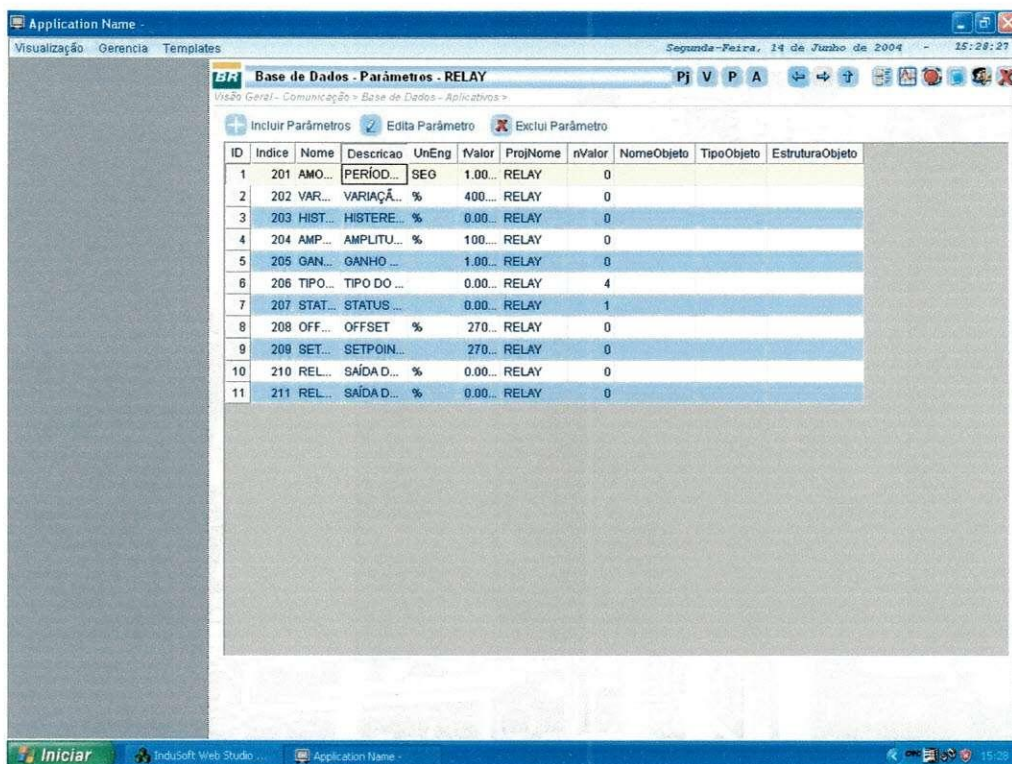


Figura 19. Software SSP-Laplace.

Nossas atividades durante a visita compreenderam além de uma breve explanação sobre o software por parte do visitante, o suporte às suas atividades durante sua permanência no laboratório que teve como objetivo a migração do algoritmo existente no software ProcessTune para o SSP-Laplace. Depois da migração do algoritmo foram conduzidos alguns testes no sistema existente no laboratório conectado ao módulo termoeletrico. Para realização dos testes foi configurada a comunicação OPC entre o SSP-Laplace e o RSLinx. Ainda, foram necessárias modificações no programa que operava no CLP já que o SSP-Laplace transferia apenas dado em ponto flutuante.

3.4. Estudo da Especificação XML-OPC DA.

Dando continuidade aos estudos realizados sobre o padrão OPC, foi realizado um estudo sobre a especificação XML-OPC inclusive com a apresentação de seminários sobre o assunto e outros temas relacionados como, por exemplo, a plataforma.NET. Este trabalho serviu como fundamentação e preparação para um projeto que deverá ter início em breve no laboratório. Neste projeto, em parceria com a Petrobrás, propõem-se um estudo comparativo das tecnologias relacionadas e a criação de soluções para automação de procedimentos no meio industrial utilizando sistemas móveis sem fio como SmartPhones e PDA's. Foram estudadas as principais características da especificação XML-OPC e como esta poderia ser adequada para prover a comunicação entre os dispositivos móveis no campo e o resto do sistema.

A especificação XML-OPC define como pode ser compatibilizada a comunicação utilizando o padrão OPC através de arquivos XML. Assim, OPC pode ser facilmente integrado com aplicações WEB através da utilização do padrão XML. XML (*eXtensible Markup Language*) é uma tecnologia para descrição de dados portátil, amplamente suportável e não proprietária. Foi criada em 1996 pelo consórcio W3C (*World Wide Web Consortium*) e desde então virou um padrão para armazenamento de dados que são transferidos entre aplicações. Qualquer tipo de dado pode ser descrito utilizando XML. Já OPC é baseado na tecnologia OLE/COM o que o torna altamente dependente da plataforma Windows. Existem opções de portar DCOM (*Distributed Component Object Model*, ou Modelo de Objetos Componentes Distribuídos) para outros sistemas operacionais como Linux. Porém, a especificação XML-OPC é uma solução mais versátil principalmente pela portabilidade de XML para qualquer sistema operacional, enquanto que utilizando DCOM este deveria ser especificamente transportado para cada sistema.

XML-OPC utiliza conceitos do protocolo SOAP (Protocolo de Acesso a Objetos Simples) de forma que informação estruturada possa ser transmitida em uma mensagem SOAP. SOAP já é amplamente utilizado para transferência de dados na WEB em especial entre aplicações business-to-business. Este é um protocolo independente de plataforma que utiliza XML para chamadas remota de métodos em HTTP (*Hypertext Transfer Protocol*). Cada pedido e cada resposta é encapsulado em uma mensagem SOAP que contém toda a informação necessária em um arquivo XML para processar seu conteúdo. HTTP é o protocolo padrão para envio de informação na Internet, seu uso junto aos arquivos XML permite a troca de informação entre diferentes sistemas operacionais através das mensagens SOAP.


```

<soap:Body>
  <WriteResponse xmlns="http://opcfoundation.org/webservices/XMLDA/1.0/">
    <WriteResult
      RcvTime="2003-05-27T05:19:26.3687500-07:00"
      ReplyTime="2003-05-27T05:19:26.4687500-07:00"
      ServerState="running" />
    <RItemList>
      <Items ItemName="Simple Types/UInt" />
      <Items ItemName="Simple Types/Int" />
      <Items ItemName="Simple Types/Float" />
    </RItemList>
  </WriteResponse>
</soap:Body>

```

Figura 20. Exemplo de mensagem SOAP em OPC-XML.

Na especificação estão todas as informações necessárias para o desenvolvimento de aplicações utilizando XML-OPC. A especificação define o procedimento para conexões entre cliente e servidor que neste caso são *subscriptions* (assinaturas) já que a conexão não é mantida, define também os tipos de dados que podem ser trocados, os tipos de faltas e erros que podem ocorrer durante o envio de dados ou a operação do servidor, e os tipos de chamadas e respostas que podem ocorrer. Ainda, especificação XML-OPC define os *schemas* que são arquivos para validação da formatação dos arquivos de mensagens utilizados pelos programas conhecidos como *parsers* responsáveis pelo tratamento dos arquivos XML.

4. CONCLUSÕES

O estágio foi uma experiência enriquecedora visto que neste período foi possível acompanhar a rotina de um laboratório de pesquisa e trabalhar com as mais recentes tecnologias na área de automação industrial em um meio composto por profissionais de alto valor técnico.

O conhecimento teórico adquirido durante todo o curso em especial na área de automação industrial e controle foi posto em prática e ampliado durante a permanência no laboratório e em especial através da participação no projeto desenvolvido para a Petrobrás.

Apesar do estágio ter sido realizado dentro do ambiente acadêmico foi possível entrar em contato com pessoal da indústria através do suporte dado e troca de experiências durante a visita do engenheiro da Petrobrás como também utilizar e se familiarizar com tecnologias que

estão começando a ser aplicadas na indústria como OPC e XML-OPC. Ainda, foi possível adquirir um bom conhecimento sobre a utilização de equipamentos industriais na prática através do trabalho desenvolvido com CLP5/15 da Allen Bradley.

REFERÊNCIAS BIBLIOGRÁFICAS

Manual de Programação e Operação CLP-5, Allen Bradley.

Fonseca, M de O; Seixas Filho, Constantino; Las Casas, F, **Padrão OPC: Aplicação e Implementação**, ISA, 2005.

Souza, L.C.A., **Desenvolvimento de um Driver Padrão OPC para Controlador Lógico Programável em Ambiente Windows NT**. Dissertação de Mestrado, Departamento de Engenharia Elétrica, UFMG, Belo Horizonte, MG, Março de 1999.

OPC-XML DA Specification, OPC Foundation, July 12, 2003.

De Arruda, G.H.M., **Sistemas de Realimentação por Relé**. Tese de Doutorado, Departamento de Engenharia Elétrica, UFCG, Campina Grande, PB, Agosto de 2003.

Experimental Evaluation of PI Controllers Tuning Rules and Controller Redesign Applied to a Thermal Process

George Acioli Jr.
Marcus André Ramos Berger
Péricles Rezende Barros

Universidade Federal de Campina Grande

Departamento de Engenharia Elétrica

Av. Aprigio Veloso 882, Caixa Postal 10105, CEP: 58109-970, Campina Grande-PB, Brasil
georgeacioli@yahoo.com.br, marcusarb@yahoo.co.uk, prbarros@dee.ufcg.edu.br
www.dee.ufcg.edu.br

Abstract— In this paper it is presented an experimental evaluation of a few step and relay experiments based PI tuning techniques applied to a Thermal Process. The resulting closed loop gain and phase margin are identified using closed loop relay experiments. Controller redesign is performed to yield desired specifications and the resulting closed loops are once more evaluated.

I. INTRODUCTION

The use of PID and PI controllers is common in most industrial process ([1] and [8]) as yield good results when applied to control process which have a simple process dynamics and modest performance requirements. Classical techniques are widely used to tune these controllers as they require simple experiments and are very simple to apply. However, the classical methods provide good performance only for the process models for which they were tuned. Closed loop tests can be applied to evaluate the obtained performance and controller redesign can be used to achieve better stability or performance results. For instance, in [2] gain and phase margins are evaluated while in [4] the closed loop is evaluated in relation to a Symmetrical Optimum design. In both cases the controller is redesigned to get a closed loop closer to the specifications.

In this paper are presented some experimental results of a few standard step and relay based PI tuning procedures applied to a laboratory thermal process. The resulting closed loops are evaluated. The controller is redesigned based in relay experiments on the closed-loop system. It should be remarked that closed loop procedures are very attractive to industrial applications. PID controller results are similar and will not be presented.

Initially the experimental setup and the identification experiments are presented. Following the PI tuning rules are reviewed and the experiment results are presented. The closed loop redesign techniques are discussed and applied to selected closed loops. Finally the results are presented and the techniques compared.

II. EXPERIMENTAL SETUP AND MODEL IDENTIFICATION

In this section the experimental setup is discussed: the process is described and then the tests are presented.

A. The Process

The laboratory scale process consists of two thermoelectric Peltier module acting as heat pumps on two parts of a metal plate heat load. Two air coolers are used to extract heat from the other faces of each Peltier module. The process works as a coupled TITO system with temperature varying between 10°C and 70°C when operating at a room temperature of around 24°C . Power is applied using PWM actuators while the temperatures are measured using LM35 sensors. The modelling of each thermoelectric module results in a complex model that is highly nonlinear as can be seen in [6]. Linearization and model reduction usually results in a second order model (without including the actuator and sensor dynamics). In this paper, the model is assumed to be unknown and the process is approximated either by a FOPDT model obtained from a step response experiment, or by a critical point estimated from a relay experiment, or even not defined when only the closed loop evaluated.

B. The Step Response Experiment

An open loop process step response experiment is performed on the process and a First Order Plus Time Delay model

$$G(s) = \frac{K_p}{1 + sT} e^{-s\tau_d}$$

is estimated using the procedure presented in [5]. The model is characterized by three parameters: the static gain K_p , the time constant T , and the dead time τ_d . Identification results in

$$K_p = 1.07, \quad T = 98.4\text{s} \quad \text{and} \quad \tau_d = 17\text{s}.$$

C. Standard Relay Experiment

A standard relay test presented in [1] is used to estimate the critical point and frequency. The idea is to use the

relay with amplitude d in series with the process with negative feedback. For most types of processes, a relay with amplitude d in an unit closed loop feedback leads to limit cycle operation, with oscillation conditions given by

$$G(j\omega_\pi) \cong m = -\frac{\pi a}{4d},$$

with ω_π the critical frequency (phase = $-\pi$ rad) and a the process output amplitude. Application of the standard relay to the process results in

$$K_\pi = 6.0 \text{ and } T_\pi = 83.52s$$

with K_π and T_π the critical gain and period, respectively.

It can be shown [7] that if this relay test is applied to a closed loop system, with transfer function $T(s)$, the limit cycle occurs at the closed loop critical frequency and the gain margin can be computed from the loop gain

$$L(j\omega_{gm}) = G(j\omega_{gm})C(j\omega_{gm}) \cong \frac{m}{1-m}.$$

In Figure 1 an example of such a gain margin identification experiment is shown.

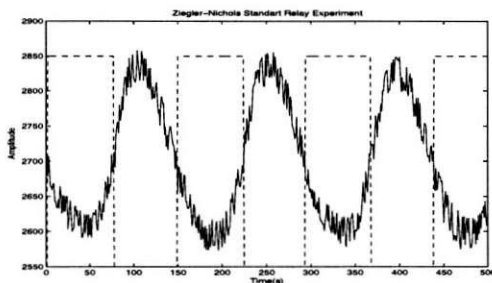


Fig. 1. Standard Relay Experiment Applied to the Closed Loop (Gain Margin).

D. Loop Gain Experiment

A general relay procedure to estimate the frequency point for which a given transfer function has a desired gain is presented in [3]. If the loop gain is under test, the feedback structure is presented in Fig. 2. The conditions of the limit cycle operation are defined by the following proposition.

Consider a stable closed loop $T(s)$, with Loop Gain $L(s)$, and a real positive number r so that the transfer function

$$F(s) = \frac{2}{r} \frac{T(s)}{T(s) \left(\frac{1-r}{r} \right) + 1} - 1 \quad (1)$$

is also stable. Then if a limit cycle is present it oscillates at a frequency ω_o such that

$$|L(j\omega_o)| \approx r.$$

Proof: See [3]. ■

This procedure allows the estimation of the frequency at which the loop transfer function magnitude is close to r .

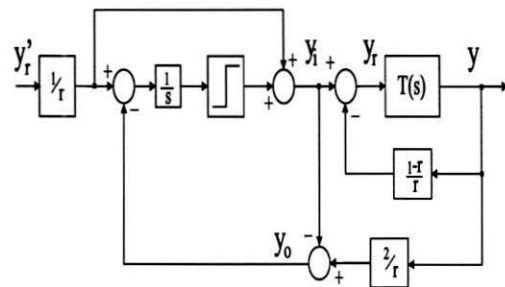


Fig. 2. Loop Gain Transfer Function Estimation.

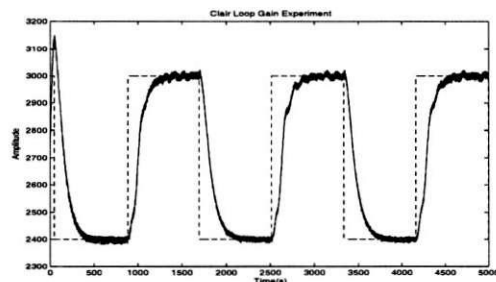


Fig. 3. Example of a Loop Gain Test (Phase Margin).

Here $r = 1$ will be used, so that a limit cycle develops at the loop gain crossover frequency. The phase margin can be estimated and used for stability evaluation and controller redesign. In Figure 3 an example of such a phase margin identification experiment is shown.

III. THE DESIGN TECHNIQUES

Using experiments described above, several design techniques can be applied. The design techniques presented here are standard in the control field and thus just a brief review will be done. The readers are referred to the books [1] and [8] and references therein.

A. Step Response Based Techniques

The following techniques are based on the Step Response experiments previously described.

A.1 Ziegler-Nichols Method

Based on the parameters τ_d , T and K_p , the Ziegler-Nichols step response method can now provide the controller parameters in Table I, where $\Theta = \frac{\tau_d}{T}$. In Table II are shown the controller parameters for all designs, including the present one. The step response for the closed loop with this controller is shown in Figure 4.

A.2 Cohen-Coon Method

The Cohen-Coon method is very similar to the Ziegler-Nichols step response method (Table III). Comparing the tables, it may be seen that the methods are quite similar for small values of Θ . The step response for the closed loop with this controller is shown in Figure 5.

TABLE I
ZIEGLER-NICHOLS METHOD PARAMETERS

| Controller | K_c | T_i | T_d |
|-------------|--------------------------|-------|-------|
| P | $\frac{1}{K_p \Theta}$ | - | - |
| PI | $\frac{0.9}{K_p \Theta}$ | $3L$ | - |
| PID(Series) | $\frac{0.6}{K_p \Theta}$ | L | L |

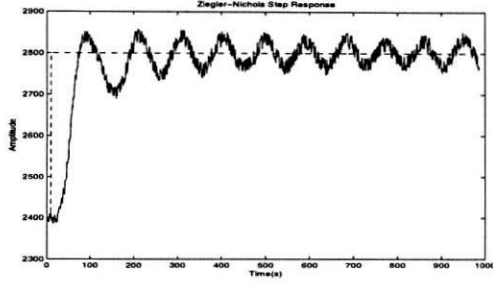


Fig. 4. Ziegler Nichols (Time) Controller Closed Loop Step Response.

A.3 Chien, Hrones and Reswick Method

Chien, Hrones and Reswick (CHR) proposed to use "quickest response without overshoot" or "quickest response with 20% overshoot" as design criteria. They also made the important observation that tuning for setpoint response or load disturbance response are different. In this paper, we use the tuning for setpoint response without overshoot (Table IV). The step response for the closed loop with this controller is shown in Figure 6.

A.4 Internal Model Controller (IMC)

Consider a closed-loop system with a controller based on IMC where G_m denotes a model of the process, G_m^+ is an approximate inverse of G_m , and G_f is a low-pass filter introduced to obtain a system that is less sensitive to modelling errors. A common choice is $G_f(s) = \frac{1}{1+sT_f}$, where T_f is a design parameter

$$G_c = \frac{G_f G_m^+}{1 - G_f G_m^+ G_m}$$

For a FOPDT model, an approximate inverse is given by

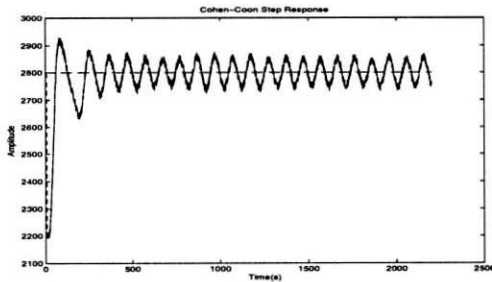


Fig. 5. Cohen-Coon Controller Closed Loop Step Response.

TABLE II
CONTROLLER PARAMETERS PARAMETERS

| | K_c | T_i |
|--------------------------|-------|--------|
| Step Response Experiment | | |
| Ziegler Nichols | 4.85 | 51 |
| Cohen Coon | 4.93 | 41.3 |
| Chien SetPoint 0% | 1.89 | 118.05 |
| IMC | 2.18 | 98.38 |
| Relay Experiment | | |
| Ziegler Nichols | 2.72 | 69.6 |
| Tyres-Luyben | 2.13 | 191.4 |
| Clair | 2.72 | 87 |

TABLE III
COHEN-COON METHOD PARAMETERS

| Controller | K_c | T_i | T_d |
|------------|--|--|------------------------------------|
| P | $\frac{1}{K_p} (0.35 + \frac{1}{\Theta})$ | - | - |
| PI | $\frac{1}{K_p} (0.083 + \frac{0.9}{\Theta})$ | $\frac{3.3+0.31\Theta}{1+2.2\Theta} \tau_d$ | - |
| PID | $\frac{1}{K_p} (0.25 + \frac{1.35}{\Theta})$ | $\frac{2.5+0.46\Theta}{1+0.61\Theta} \tau_d$ | $\frac{0.37}{1+0.19\Theta} \tau_d$ |

$$G_m^+ = \frac{1+sT}{K_p}$$

Approximating the time delay by $e^{-sL} \approx 1 - sL$, a PI controller is obtained.

$$G_c(s) = \frac{1+sT}{K_p s(\tau_d + T_f)} = \frac{T}{K_p(\tau_d + T_f)} \left(1 + \frac{1}{Ts}\right)$$

If the time delay is approximated by $e^{-sL} \approx \frac{1-\frac{sL}{2}}{1+\frac{sL}{2}}$, a PID controller is obtained.

$$G_c(s) = \frac{(1 + \frac{s\tau_d}{2})(1+sT)}{K_p s(\tau_d + T_f + \frac{sT_f\tau_d}{2})} \approx \frac{(1 + \frac{s\tau_d}{2})(1+sT)}{K_p s(\tau_d + T_f)}$$

The step response for the closed loop with this controller is shown in Figure 7.

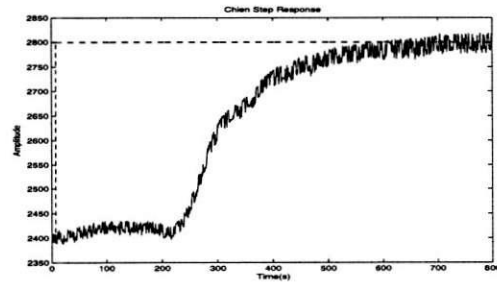


Fig. 6. Chien-Hrones-Reswick Controller Closed Loop Step Response.

TABLE IV
CHIEN, HRONES AND RESWICK METHOD PARAMETERS

| Controller | K_c | T_i | T_d |
|------------|---------------------------|-------------|--------------|
| P | $\frac{0.3}{K_p \Theta}$ | - | - |
| PI | $\frac{0.6}{K_p \Theta}$ | $4\tau_d$ | - |
| PID | $\frac{0.95}{K_p \Theta}$ | $2.4\tau_d$ | $0.42\tau_d$ |

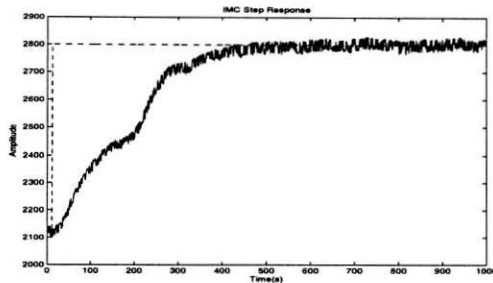


Fig. 7. IMC Controller Closed Loop Step Response.

B. Standard Relay Based Techniques

The the standard relay based techniques use the critical point information In this paper we consider the following techniques.

B.1 Ziegler Nichols Method

The standard relay can be used with the classic Ziegler-Nichols Frequency Response Method. The critical period T_π and critical gain K_π can be used to controller design using the formulas presented in Table V. The step response for the closed loop with this controller is shown in Figure 8.

TABLE V
ZIEGLER NICHOLS METHOD FREQUENCY RESPONSE METHOD
PARAMETERS

| Controller | K_c | T_i | T_d |
|------------|---------------------|---------------------|-------------------|
| PI | $\frac{K_\pi}{2.2}$ | $\frac{T_\pi}{1.2}$ | - |
| PID | $\frac{K_\pi}{1.7}$ | $\frac{T_\pi}{2}$ | $\frac{T_\pi}{8}$ |

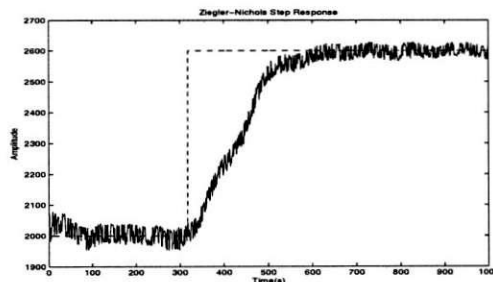


Fig. 8. Ziegler-Nichols (Frequency) Controller Closed Loop Step Response.

B.2 Clair Method

The Clair method tune is made using Table VI while the step response for the closed loop with this controller is shown in Figure 9.

TABLE VI
CLAIR METHOD PARAMETERS

| Controller | K_c | T_i | T_d |
|------------|------------|---------|-------------------|
| PI | $0.4K_\pi$ | T_π | - |
| PID | $0.5K_\pi$ | T_π | $\frac{T_\pi}{8}$ |

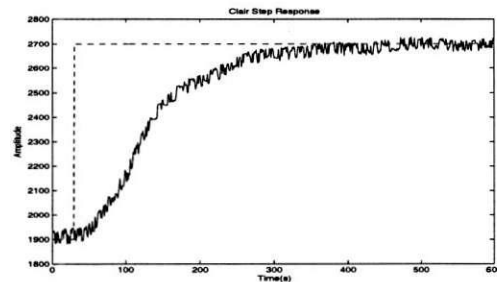


Fig. 9. Clair Controller Closed Loop Step Response.

C. Tyreus-Luyben Method

This method gives a more conservative setting than Ziegler-Nichols. The controller parameters are given as presented in table VII while the step response for the closed loop with this controller is shown in Figure 10.

TABLE VII
TYREUS-LUYBEN METHOD PARAMETERS

| Controller | K_c | T_i | T_d |
|------------|---------------------|------------|---------------------|
| PI | $\frac{K_\pi}{3.2}$ | $2.2T_\pi$ | - |
| PID | $\frac{K_\pi}{2.2}$ | $2.2T_\pi$ | $\frac{T_\pi}{6.3}$ |

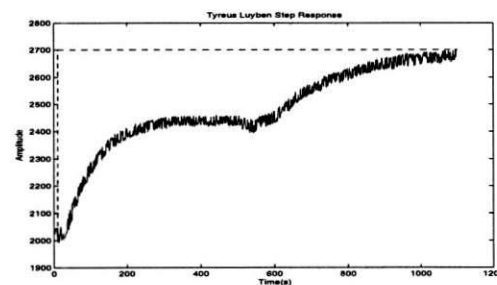


Fig. 10. Tyreus-Luyben Controller Closed Loop Step Response.

IV. CLOSED LOOP EVALUATION AND REDESIGN

In this sections the redesign procedures are presented and some examples illustrate the techniques. This techniques are applied to closed loop system using Standard Relay and Loop Gain Experiments.

A. Techniques

The closed loop performances are evaluated here on the phase and gain margins sense. This information will be used to redesign the controllers as proposed at [2]

The problem is summarized as follows: given the closed loop system, how one can redesign the controller in such way that a new phase and gain margin specifications can be achieved. This problem is solved using an iterative approach applied to the following equations:

$$\angle G(j\omega_u) C(j\omega_u) = -\pi, \quad (2)$$

$$|G(j\omega_u) C(j\omega_u)| = \frac{1}{A_m}, \quad (3)$$

$$|G(j\omega_g) C(j\omega_g)| = 1, \quad (4)$$

$$\angle G(j\omega_g) C(j\omega_g) = -\pi + \phi_m. \quad (5)$$

where A_m is the desired gain margin and ϕ_m the desired phase margin. The iterative algorithm is used such that it only requires the knowledge of the frequencies ω_u and ω_g at each iteration. These frequencies are the solutions to Eqs. (2) and (4), and estimates can be obtained using the *standart relay experiment* and the *loop gain experiment* respectively. This algorithm uses the following lemmas to update the controller's parameters.

Controller Redesign for Gain Margin: The controller gain can be calculated for achieving the gain margin A_m using Eq. (3). That is, with the current gain margin, GM^k , and the critical frequency, ω_u^k , one can compute the controller proportional gain, \bar{K}_c^{k+1} , from

$$\bar{K}_c^{k+1} = \frac{K_c^k GM^k}{A_m}. \quad (6)$$

Now a new intermediate controller is

$$\bar{C}^{k+1}(s) = \bar{K}_c^{k+1} \left(\frac{s + 1/T_i^k}{s} \right). \quad (7)$$

Controller Redesign for Phase Margin: The controller gain can be calculated for achieving the phase margin ϕ_m using Eq. (5). This step is separated into two parts:

1. (a) Determine T_i^{k+1} such that Eq. (5) is satisfied, i.e.,

$$T_i^{k+1} = \frac{\tan[-\pi + \phi_m - PM^k + \tan^{-1}(\omega_g^k T_i^k)]}{\omega_g^k}. \quad (8)$$

The phase contribution from the PI controller ranges from -90° to 0° , and this information must be used in order to avoid invalid values of T_i^{k+1} . Since $\angle G(j\omega_g^k) + \angle C^{k+1/2}(j\omega_g^k) = -180^\circ + \phi_m$, then the following condition must be satisfied in the above steps

$$-180^\circ + \phi_m < \angle G(j\omega_g^k) < -90^\circ + \phi_m, \quad (9)$$

If Eq. (9) is not satisfied, stop the iteration.

(b) Now, update the controller proportional gain K_c^{k+1} such that the loop gain at the frequency ω_g^k is equal to one,

$$K_c^{k+1} = \bar{K}_c^{k+1} \frac{\sqrt{(1/T_i^k)^2 + \omega_g^2}}{\sqrt{(1/T_i^{k+1})^2 + \omega_g^2}}. \quad (10)$$

The controller at the end of the iteration is finally given by

$$C^{k+1} = K_c^{k+1} \left(\frac{s + 1/T_i^{k+1}}{s} \right). \quad (11)$$

B. Redesign Examples

The gain and phase margin experiments were applied to selected closed loops. The estimates are presented in Table VIII.

TABLE VIII
ESTIMATED GAIN AND PHASE MARGINS

| | L. G. | | Relay | | GM | PM |
|-------------|-------|--------|-------|---------|-------|-------|
| | Mag. | Ph. | Mag. | Ph. | | |
| C. C. | 0.965 | -28.16 | 0.951 | -175.89 | 1.051 | 79.97 |
| IMC | 0.939 | -29.44 | 0.143 | -177.90 | 7.010 | 82.07 |
| Z.-N. Freq. | 0.986 | -23.74 | 0.286 | -178.13 | 3.498 | 80.02 |
| Clair | 0.947 | -28.76 | 0.204 | -176.08 | 4.900 | 81.67 |

Using the data estimated and defining new gain margins specifications yields the results presented in Table IX.

TABLE IX
PI CONTROLLERS AFTER GAIN MARGIN REDESIGN

| | K_c | T_i |
|------------------------------|-------|--------|
| Cohen Coon ($GM = 2.5$) | 2.073 | 41.307 |
| Cohen Coon ($GM = 3$) | 1.727 | 41.307 |
| Cohen Coon ($GM = 3.5$) | 1.480 | 41.307 |
| IMC ($GM = 2.5$) | 6.118 | 98.377 |
| IMC ($GM = 3$) | 5.098 | 98.377 |
| IMC ($GM = 3.5$) | 4.370 | 98.377 |
| Ziegler-Nichols ($GM = 3$) | 3.177 | 69.6 |
| Clair ($GM = 2.5$) | 5.343 | 87 |
| Clair ($GM = 3$) | 4.453 | 87 |
| Clair ($GM = 3.5$) | 3.817 | 87 |

After applying the new controllers, the Phase Margin Experiments results in the controllers shown in Table X

TABLE X
PI CONTROLLERS PHASE MARGIM REDESIGN

| | K_c | T_i |
|--|--------|--------|
| C. C. ($GM = 3$ and $PM = 73^\circ$) | 1.1657 | 27.570 |
| IMC ($GM = 3$ and $PM = 70^\circ$) | 2.4 | 43.830 |
| Z.-N. ($GM = 3.5$ and $PM = 75^\circ$) | 1.512 | 38.106 |

Some of the new controllers closed loop step response are presented in the Figures 11 to 13.

As expected, the controller redesign in the direction to decrease the gain margin results in more aggressive controllers (IMC and Clair) while increasing the gain margin yields more stable closed loops. The gain in applying the experiments is that the closed loop is evaluated and the redesign can be made based on solid ground.

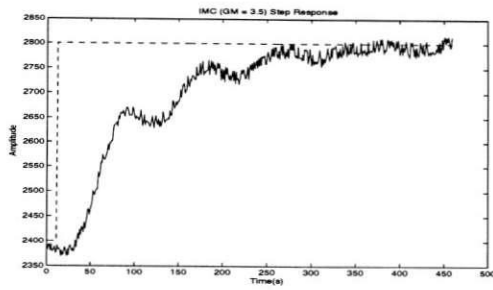


Fig. 11. Redesign IMC Controller Closed Loop Step Response.

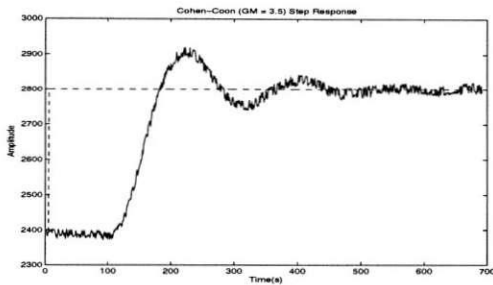


Fig. 12. Redesign of Cohen-Coon Controller Closed Loop Step Response.

V. CONCLUSION

In this paper it was presented an experimental evaluation of PI controller tuning rules when applied to a thermal process. The closed loops are evaluated in relation to gain and phase margins. Controller redesign is done based on the estimated margins and the results point to the effectiveness of the procedure.

REFERENCES

- [1] K. J. Åström and T. Hägglund, *PID Controllers: Theory, Design and Tuning*, Instrument Society of America, Research Triangle Park, North Carolina, 1995.
- [2] G. H. M. de Arruda and P. R. Barros, "Relay Based Gain and Phase Margins PI Controller Design," *IEEE Transactions on Inst. and Meas. Tech.*, vol. 52, pp. 1548–1553, 2003.
- [3] G. H. M. de Arruda and P. R. Barros, "Transfer Function Relay Based Frequency Points Estimation," *Automatica*, vol. 39, pp. 309–315, 2003.
- [4] M. A. R. Berger and P. R. Barros, "Controller Evaluation and

Redesign using Relay Experiments," *IFAC05 World Congress*, Prague, (To be presented), 2005.

- [5] F. S. Coelho e P. R. Barros, "Continuous-time identification of first-order plus dead-time models from step response in closed loop," *13th IFAC Symposium on System Identification (SYSID-2003)*, Rotterdam, 2003.
- [6] B.J.Huang,C.L.Duang, "System dynamic model and temperature control of a thermoelectric cooler," *Internacional Journal of Refrigeration*, vol. 10, pp. 197–207, 2000.
- [7] T. S. Schei, "Automatic Tuning of PID Controllers Based on Transfer Function Estimation," *Automatica*, vol. 30, pp. 1983–1989, 1994.
- [8] K. K. Tan, Q. G. Wang and C. C. Hang , *Advances in PID Control*, Springer-Verlag, London, 1999.

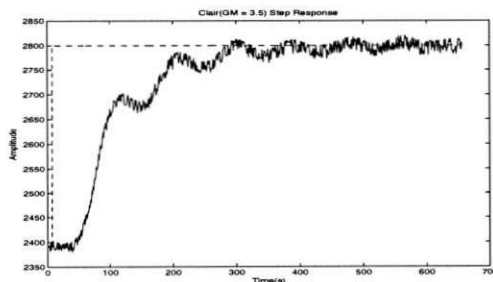


Fig. 13. Redesign of Clair Controller Closed Loop Step Response GM = 3.5.