



Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Curso de Graduação em Engenharia Elétrica

ELI DIAS INOCÊNCIO

Modelagem, Simulação e Controle de Evaporador usando CLP

Campina Grande, Paraíba
Junho de 2012

ELI DIAS INOCÊNCIO

Modelagem, Simulação e Controle de Evaporador usando CLP

Trabalho de Conclusão de Curso submetido
à Unidade Acadêmica de Engenharia Elétrica
da Universidade Federal de Campina Grande
como parte dos requisitos necessários para a
obtenção do grau de Bacharel em Ciências no
Domínio da Engenharia Elétrica.

Área de Concentração: Automação Industrial

Orientador:
Professor João Batista Moraes dos Santos, M. Sc.

Campina Grande, Paraíba
Junho de 2012

ELI DIAS INOCÊNCIO

Modelagem, Simulação e Controle de Evaporador usando CLP

Trabalho de Conclusão de Curso submetido
à Unidade Acadêmica de Engenharia Elétrica
da Universidade Federal de Campina Grande
como parte dos requisitos necessários para a
obtenção do grau de Bacharel em Ciências no
Domínio da Engenharia Elétrica.

Área de Concentração: Automação Industrial

Aprovado em: / /

Professor Avaliador
Universidade Federal de Campina Grande
Avaliador

Professor João Batista Moraes dos Santos, M. Sc.
Universidade Federal de Campina Grande
Orientador, UFCG

Agradecimentos

Agradeço primeiramente a Deus por ter me concedido todos os meios necessários para a conclusão deste curso, uma vez que toda boa dádiva e todo dom perfeito são lá do alto, descendo do Pai das luzes, em quem não pode existir variação ou sombra de mudança.

Agradeço ao meu orientador, professor João Batista, pela orientação e por todo subsídio para este trabalho.

Agradeço ao professor George Acioli por ter me ajudado sempre que precisei.

Agradeço ao professor Péricles, o qual sugeriu o tema e traçou as linhas gerais para o trabalho.

Agradeço aos meus pais por, desde cedo, investirem em minha educação, por me ensinarem a disciplina e o respeito, e por todo o afeto que me acompanha desde o meu nascimento.

Agradeço também a minha futura esposa, Simone, pela compreensão e pelo incentivo nos momentos de desestímulo.

Resumo

No presente trabalho foram exploradas técnicas de modelagem e simulação com o intuito de projetar e verificar a eficiência de controladores atuando em processos industriais. Além disso, objetivou-se comprovar a utilidade da comunicação entre *hardwares* através da interface OPC. Para tanto, modelou-se matematicamente um sistema evaporador industrial, o qual pôde ser simulado ao se desenvolver um simulador no *software* Matlab. Isso permitiu a realização virtual de diversos ensaios para o processo em questão, permitindo assim a aplicação de técnicas de identificação de sistemas, as quais permitiram a representação do sistema em questão por meio de funções de transferência. Uma vez estas obtidas, projetou-se controladores PID, fazendo uso de diversas técnicas diferentes, para a realização do controle do processo. Estes foram executados utilizando-se de um Controlador Lógico Programável (CLP), o qual foi interligado ao simulador através da interface de comunicação OPC, o que permitiu a troca de dados entre ambos. Após a simulação do processo com a intervenção de cada controlador projetado, pôde-se concluir qual das técnicas utilizadas apresentou melhor resultado.

Palavras Chave: Modelagem; Matlab; Controlador Lógico Programável; OPC.

Sumário

Sumário	v
Lista de Figuras	vii
Lista de Tabelas	viii
Lista de Termos	ix
1 Introdução	1
1.1 Motivação	1
1.2 Revisão Bibliográfica	1
1.3 Objetivos	2
1.4 Organização do Trabalho	3
2 Fundamentação Teórica	4
2.1 Controlador PID	4
2.2 Técnicas de Sintonia	6
2.2.1 Ziegler-Nichols	6
2.2.2 Chien, Hrones, and Reswick	7
2.2.3 Cohen-Coon	7
2.2.4 Sintonia Lambda	7
2.2.5 Método de Haalman	8
2.2.6 Método SIMC	8
2.3 Controlador Lógico Programável (CLP)	9
2.4 Comunicação OPC	10
3 Modelagem	12
3.1 O Evaporador	12
3.2 O Modelo	13
3.2.1 O Separador	13
3.2.2 O trocador de Calor	14
3.2.3 A Jaqueta	15
3.2.4 O Condensador	15
3.2.5 O Produto	16
3.2.6 Equações de Estado	17
4 Identificação do Sistema e Projeto dos Controladores	18
4.1 Identificação do Sistema	18
4.2 Projeto dos Controladores	20

5	Resultados Obtidos Com o Controlador PID Real	23
5.1	<i>Hardware in the Loop</i>	23
5.2	Dados Coletados	24
6	Conclusão	28
	Referências Bibliográficas	29
A	Código do Simulador do Evaporador (Matlab)	30
B	Código em Ladder para o CLP	33
C	Código do Simulador para a comunicação OPC (Matlab)	34

Lista de Figuras

2.1	Processo em malha fechada.	4
2.2	Diagrama de aplicação de CLP. (Modificado de: ver [9])	9
2.3	Representação de um diagrama em Ladder. (Extraído de: ver [9].)	10
2.4	Acesso aos servidores OPC usando as interfaces Padrão e de Automação.	11
3.1	Evaporador Típico. (Modificado de: ver [1])	12
4.1	Concentração X2 com degrau em F2.	19
4.2	Pressão de operação P2 com degrau em F2.	20
4.3	Modelos em malha fechada (Simulink)	21
4.4	Melhores resultados para o controle da concentração X2.	22
4.5	Melhores resultados para o controle da pressão de operação P2.	22
5.1	Comunicação cliente-servidor	24
5.2	Representação dos sistemas em malha fechada.	24
5.3	Resultados obtidos para diversos controladores para o controle da concentração X2.	25
5.4	Resultados obtidos para diversos controladores para o controle da pressão de operação P2.	26

Lista de Tabelas

2.1	Parâmetros do controlador para o método da resposta ao degrau.	6
2.2	Parâmetros do controlador para o método do período crítico.	6
2.3	Parâmetros do controlador para o método de Chien, Hrones e Reswick.	7
2.4	Parâmetros do controlador para o método de Cohen-Coon.	7
2.5	Parâmetros do controlador para o método Lambda.	7
2.6	Parâmetros do controlador para o método de Haalman.	8
2.7	Parâmetros do controlador para o método SIMC.	8
3.1	Constantes utilizadas.	14
4.1	Critérios de desempenho para diversos controladores - X2.	21
4.2	Critérios de desempenho para diversos controladores - P2.	21
5.1	Critérios de desempenho para diversos controladores - X2.	25
5.2	Critérios de desempenho para diversos controladores - P2.	26

Lista de Termos

CLP Controlador Lógico Programável

PID Controlador Proporcional - Integal - Derivativo

OPC OLE for Process Control

SP Set Point

MV Manipulated Variable

PV Process Variable

Capítulo 1

Introdução

1.1 Motivação

O controle de processos industriais foi trazido ao campo da engenharia elétrica à medida que a tecnologia se desenvolveu, principalmente após o advento dos controladores. Com este objetivo, diversos métodos e dispositivos foram desenvolvidos, entre os quais, destaca-se o Controlador Proporcional - Integral - Derivativo, ou PID.

Para a realização da sintonia de controladores PID, é necessário, primeiramente, o conhecimento do processo o qual se deseja controlar, tendo-se, de preferência a função de transferência do mesmo. Como nem sempre é fácil realizar ensaios em determinadas plantas a fim de se obter curvas de resposta para diversas entradas do sistema, seja pela complexidade e tempo dispendido ou pelos gastos demandados, uma alternativa para o problema é a modelagem computacional do processo. A partir de um modelo, é possível se obter os dados necessários para o projeto de controladores, os quais podem em seguida ser utilizados no sistema real.

Desse modo, a motivação para a elaboração deste trabalho foi a familiarização com técnicas de modelagem e controle, as quais são de grande utilidade para o engenheiro eletricitista atuante na área de controle e automação, seja no exercício de atividades de pesquisas acadêmicas ou na atividade industrial.

Além disso, serviu como motivação o contato inicial com a interface de comunicação OPC, a qual é largamente utilizada para a comunicação entre *hardwares* diversos, o que a torna uma ferramenta valiosa para um profissional de automação.

1.2 Revisão Bibliográfica

O evaporador é um sistema industrial, bastante comum nas indústrias de papel e açúcar, onde um solvente (e.g. água) é evaporado ao absorver calor proveniente de um fluxo de vapor. O modelo deste sistema é descrito em detalhes em [Newell e Lee, 1989], onde foi obtido, primeiramente, o modelo não linear para o processo, o qual serviu como base para o uso de diversas técnicas de controle, como o controle preditivo, controle adaptativo, controle usando lógica fuzzy, etc.

Sendo de grande utilidade para trabalhos acadêmicos, outros autores usaram o sistema evaporador para desenvolver pesquisas. Dentre eles, Cao, em [Cao, 2004], desenvolveu

uma estrutura para implementar o controle auto adaptativo aplicado ao evaporador, mostrando que o sistema de controle proposto alcança auto otimização para vários distúrbios. Além dele, Haugwitz, em [Haugwitz, 2007] modelou o sistema usando o software de simulação Modelica para implementar um modelo de controle preditivo, o que mostra que o processo pode ser analisado e controlado de diversas formas.

Aproveitando a versatilidade do modelo do sistema evaporador, ele foi utilizado neste trabalho como base para a realização do controle do simulador do processo usando a interface de comunicação OPC (*OLE for Process Control*) para realizar a comunicação entre o simulador desenvolvido no *software* Matlab e o controlador PID, o qual foi executado em um CLP da *Rockwell Automation*. Essa técnica de simulação e controle é comumente conhecida como *Hardware in the Loop* (HIL), a qual começou a ser usada na década e 1980, antes mesmo da existência do OPC, tendo suas raízes na aviação.

O padrão OPC é uma interface de comunicação que visa a troca de informações em tempo real entre dispositivos de controle de diferentes fabricantes. A interface foi fruto do trabalho da *OPC Task Force*, um grupo de empresas formado pelas companhias Fisher-Rosemount, Intellution, Intuitive Technology, Opto22, Rockwell e Siemens AG, com o apoio da Microsoft. O padrão OPC começou a ser desenvolvido em 1995, tendo vindo a público a sua primeira versão em 1996.

Para a identificação do sistema, usou-se a técnica da resposta ao degrau, onde a curva de reação foi aproximada por um modelo de primeira ordem com atraso. A técnica foi proposta inicialmente em [Callender, 1936], tendo sido usada também em [Ziegler and Nichols, 1942]. Uma vez obtido o modelo para o sistema em estudo, utilizou-se os dados obtidos para a sintonia do controlador PID.

Os controladores PID foram utilizados extensivamente no desenvolvimento inicial do controle das décadas de 1870 a 1920. A forma moderna do controlador PID surgiu no desenvolvimento do controle de processos nas décadas de 1930 e 1940, como é discutido em [Bennett, 1979] e [Bennett, 1993]. Ainda hoje, esse tipo de controlador é usado extensivamente para controlar processos industriais.

As técnicas de sintonia utilizadas foram todas baseadas no modelo de primeira ordem com atraso, sendo elas: a técnica de sintonia de Ziegler-Nichols para resposta ao degrau proposta em [Ziegler and Nichols, 1942]; a técnica de sintonia proposta Chien, Hrones, e Reswick proposta em [Chien et al., 1952]; a técnica de Cohen-Coon; a técnica de sintonia Lambda, a qual foi abordada primeiramente em [Dahlin, 1968] e [Higham, 1968]; o método de Haalman, proposto em [Haalman, 1965] e o método SIMC, o qual foi proposto por Skogestad em [Skogestad, 2003].

1.3 Objetivos

O presente trabalho possui os seguintes objetivos:

1. Desenvolver simulador utilizando o software Matlab para o modelo de um evaporador industrial típico a fim de verificar seu funcionamento através do monitoramento de suas diversas variáveis de entrada e de saída.
2. Utilizar técnicas de identificação de sistemas para se obter a representação das malhas por meio de funções de transferência.

3. Utilizar diversas técnicas de sintonia a fim de projetar controladores PID para o controle das malhas referidas.
4. Fazer uso da interface de comunicação OPC para estabelecer a comunicação entre o simulador executado no Matlab e um Controlador Lógico Programável (CLP), onde pode ser executado um controlador PID.
5. Verificar a eficácia e a eficiência dos controladores projetados.

1.4 Organização do Trabalho

O trabalho é organizado da seguinte maneira: a fundamentação teórica necessária para a execução do trabalho, desde as técnicas de sintonia ao controlador utilizado estão expostos no capítulo 2. A modelagem matemática das diversas partes constituintes do sistema evaporador, bem como a definição das variáveis de estado para o modelo não linear, foi desenvolvida no capítulo 3. No capítulo 4, está detalhado o procedimento efetuado para a realização da identificação do sistema relacionado à entrada e às saídas a serem controladas, o qual foi baseado na técnica de resposta ao degrau. Os resultados obtidos para a simulação do modelo obtido usando o *software* Matlab, bem como o resultado final após o uso do controlador PID por meio da técnica *hardware in loop* (HIL) estão mostrados no capítulo 5. A conclusão para o trabalho como um todo encontra-se no capítulo 6.

Capítulo 2

Fundamentação Teórica

2.1 Controlador PID

O controlador PID (Proporcional Integral Derivativo) é, sem sombra de dúvidas, o algoritmo de controle mais utilizado na automação industrial ???. Mesmo tendo passado cerca de um século desde o seu advento, pelo menos em sua primeira forma, ele se mantém o controlador mais usado para controlar sistemas em malha fechada. Isso se dá pela sua eficiência e praticidade.

A representação matemática para o algoritmo PID é a mostrada na expressão 2.1:

$$u(t) = K \cdot (e(t) + \frac{1}{T_i} \cdot \int_0^t e(\tau) d\tau + T_d \cdot \frac{de(t)}{dt}) \quad (2.1)$$

Na expressão 2.1, onde a equação do controlador é expressa na sua forma paralela (o controlador PID pode ser expresso de diversas formas, mudando de acordo com o fabricante), u representa o sinal de controle (*Manipulated Variable* - MV), enquanto e representa o erro, o qual é caracterizado pela diferença entre o valor de referência (*Set Point* - SP) usado para o controle e o valor medido na saída do processo (*Process Variable* - PV) (SP-PV). O sinal de controle é composto pela soma de três termos: o termo-P (que é proporcional ao erro), o termo-I (que é proporcional à integral do erro) e o termo-D (que é proporcional à derivada do erro). Os parâmetros do controlador são o ganho proporcional K , o tempo integral T_i e o tempo derivativo T_d [4]. Na Figura 2.1 é representado um processo em malha fechada utilizando um controlador PID.

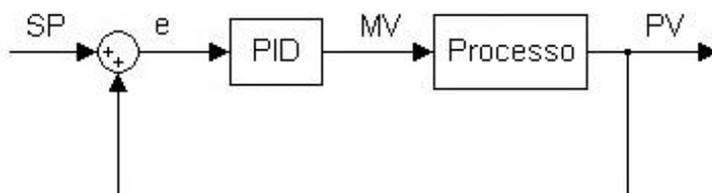


Figura 2.1: Processo em malha fechada.

O termo proporcional do controlador ($P = K \cdot e(t)$) produz um valor na saída u que é diretamente proporcional ao erro em dado instante. Dessa forma, dependendo do valor escolhido para o ganho proporcional, o controlador irá apresentar sensibilidade maior ou menor às variações do erro $e(t)$. Ou seja, para um valor elevado de K , uma pequena variação no erro acarretará em um aumento considerável no valor da saída do controlador. Isso tem o seu lado positivo, pois pode fazer com que a saída do processo se aproxime do valor de referência, mas o aumento exagerado do ganho pode tornar o sistema instável. Em contrapartida, para baixos valores de K , existirá uma sensibilidade menor do controlador para variações do erro, o que pode dificultar a saída do processo de alcançar o valor de referência, mas que possui o lado positivo de diminuir a influência de distúrbios por parte dos sensores.

O termo integral ($I = \frac{1}{T_i} \cdot \int_0^t e(\tau) d\tau$) possui como função principal fazer com que a saída do processo (PV) tem seu valor idêntico ao valor de referência adotado (SP), ou seja, fazer com que o sistema em malha fechada, em regime permanente, possua erro nulo. Como pode ser visto pela equação do termo-I, a ação integral garante a impossibilidade de haver uma situação em que o sinal de controle e o erro se encontrem ambos constantes, ou seja, um erro positivo, por menor que seja, fará com que o sinal de controle seja acrescido, enquanto um erro negativo fará com que o sinal de controle seja decrescido.

Assim como no caso proporcional, a constante de tempo T_i influencia na resposta do controlador. De modo geral, quanto maior o valor da constante, mais lentamente o sinal de controle irá variar e a saída do processo (PV) se aproximará do valor de referência (SP). Em contrapartida, quanto maior o valor de T_i , mais rápido o erro irá se aproximar do valor nulo, mas o sistema se torna mais oscilatório.

Já o termo derivativo ($D = T_d \cdot \frac{de(t)}{dt}$) possui a função principal de contribuir para a estabilidade do sistema. Como pode ser visto pela equação do termo, ele é calculado fazendo-se a estimativa da inclinação da curva do erro ao longo do tempo. Dessa forma, quanto maior a variação do erro, maior sua contribuição para o sinal de controle, sendo o inverso verdadeiro. Isso diminui a taxa de variação do MV, o que diminui a magnitude do *overshoot* causado pelo termo integral, mas também torna a resposta transitória mais lenta. Além disso, o termo derivativo torna o controlador mais sensível ao ruído contido no erro, o que pode acarretar na instabilidade do sistema para altos valores do ganho derivativo K_p .

O controlador PID pode ser tratado através de sua função de transferência, obtida aplicando-se a transformada de Laplace à equação 2.1, o que, para determinadas situações, facilita o trabalho. A função de transferência está mostrada na equação 2.2:

$$C(s) = K \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) \quad (2.2)$$

Ou, expressando-a de outra forma:

$$C(s) = K_p + \frac{K_i}{s} + K_d \cdot s \quad (2.3)$$

em que $K_p = K$, $K_i = \frac{K}{T_i}$ e $K_d = K \cdot T_d$.

2.2 Técnicas de Sintonia

Embora o controlador PID possua apenas três parâmetros, não é fácil, sem um procedimento sistemático, encontrar valores satisfatórios para eles. Na verdade, uma visita a processos industriais irá mostrar que um grande número de controladores PID estão mal sintonizados [8]. Dessa forma, diversos procedimentos foram desenvolvidos a fim de se obter o desempenho desejado, onde cada procedimento pode ter vantagens sobre os outros em situações específicas.

Neste trabalho foram utilizadas técnicas de controle úteis para processos de primeira ordem com atraso, os quais possuem função de transferência conforme mostrado na equação 2.4.

$$G(s) = \frac{k}{T \cdot s + 1} \cdot e^{-L \cdot s} \quad (2.4)$$

Assim, entre as técnicas de sintonia desenvolvidas é possível citar:

2.2.1 Ziegler-Nichols

Ziegler e Nichols desenvolveram suas técnicas de sintonia baseados na análise de dados empíricos. São elas: Análise da resposta ao degrau e o método do período crítico. A primeira delas consiste, primeiramente, em encontrar os parâmetros da função de transferência 2.4 a partir da resposta do processo para um degrau aplicado em sua entrada. Em seguida os dados são tratados como exposto na Tabela 2.1.

Tabela 2.1: Parâmetros do controlador para o método da resposta ao degrau.

Tipo de Controlador	K_p	K_i	K_d
P	$(k \cdot \frac{L}{T})^{-1}$	–	–
PI	$0,9(k \cdot \frac{L}{T})^{-1}$	$3,33L$	–
PID	$1,2(k \cdot \frac{L}{T})^{-1}$	2θ	$0,5\theta$

O método do período crítico consiste em colocar o processo em malha fechada com um controlador puramente proporcional e aumentando o valor de K_p até que a saída do processo se encontre no limite de estabilidade, ou seja, quando o sinal se comportar como uma senoide. Esse valor do ganho é chamado de K_u e é usado como descrito na Tabela 2.2. Outro parâmetro utilizado é o período da senoide gerada, o qual é chamado de T_u .

Tabela 2.2: Parâmetros do controlador para o método do período crítico.

Tipo de Controlador	K_p	T_i	T_d
P	$0,5K_u$	–	–
PI	$0,4K_u$	$0,8T_u$	–
PID	$0,6K_u$	$0,5T_u$	$0,125T_u$

Os outros métodos descritos a seguir parte da função de transferência obtida na identificação do sistema.

2.2.2 Chien, Hrones, and Reswick

Vários métodos, como o método de Ziegler-Nichols, usam as mesmas informações sobre o processo obtidas pela resposta ao degrau. O que os diferencia são os coeficientes que eles usam para o tratamento das informações. O método de Chien et al., segue esse princípio. No desenvolvimento deste, o objetivo proposto foi o de uma resposta mais rápida sem *overshoot* e o de uma resposta mais rápida com 20% de *overshoot*. Na Tabela 2.3 estão mostrados os coeficientes para o primeiro caso.

Tabela 2.3: Parâmetros do controlador para o método de Chien, Hrones e Reswick.

Tipo de Controlador	aK_p	$\frac{T_i}{L}$	$\frac{T_d}{L}$
P	0,3	–	–
PI	0,35	1,2	–
PID	0,6	1,0	0,5

em que $a = \frac{k \cdot L}{T}$.

2.2.3 Cohen-Coon

O método de Cohen-Coon também é baseado na função de transferência na forma 2.4. O critério principal para o design utilizado por este método foi a rejeição dos distúrbios de carga. Os coeficientes propostos estão mostrados na Tabela 2.4.

Tabela 2.4: Parâmetros do controlador para o método de Cohen-Coon.

Tipo de Controlador	aK_p	$\frac{T_i}{L}$	$\frac{T_d}{L}$
P	$1 + \frac{0,35\tau}{1-\tau}$	–	–
PI	$0,9(1 + \frac{0,092\tau}{1-\tau})$	$\frac{3,3-3,0\tau}{1+1,2\tau}$	–
PID	$1,35(1 + \frac{0,18\tau}{1-\tau})$	$\frac{2,5-2,0\tau}{1-0,39\tau}$	$\frac{0,37-0,37\tau}{1-0,81\tau}$

em que $a = \frac{k \cdot L}{T_1}$ e $\tau = \frac{L}{L+T}$.

2.2.4 Sintonia Lambda

A sintonia Lambda é um caso especial da alocação de pólos comumente utilizada nos processos industriais [4]. Ainda utilizando o modelo mostrado na expressão 2.4 o método propõe a sintonia de controladores PI e PID. As formas de cálculo dos parâmetros estão mostrados na Tabela 2.5.

Tabela 2.5: Parâmetros do controlador para o método Lambda.

Tipo de Controlador	K_p	T_i	T_d
PI	$\frac{1}{k} \cdot \frac{T}{L+T_{cl}}$	T	–
PID	$\frac{1}{k} \cdot \frac{L/2+T}{L/2+T_{cl}}$	$T + L/2$	$\frac{T \cdot L}{L+2T}$

Pode-se obter na tabela que o cálculo dos parâmetros do controlador PID dependem um parâmetro T_{cl} , o qual representa a constante de tempo desejada para a malha fechada. A escolha do valor para T_{cl} é de grande importância para a eficiência do método, e tarefa do projetista obtê-lo. De modo geral, alguns valores sugeridos pelo desenvolvedor do método foram: $T_{cl} = T$; $T_{cl} = 3T$ e $T_{cl} = 3L$.

2.2.5 Método de Haalman

O método de Haalman se enquadra na categoria dos métodos algébricos de sintonia, nos quais a função de transferência para o controlador é obtida a partir de especificações por meio de cálculos algébricos. Este tipo de sintonia é bem próximo do tipo de alocação de pólos, procurando minimizar critérios de erro, como o IAE, ISE e ITAE. A principal diferença é que nos métodos algébricos se especifica a função de transferência completa é especificada, em vez de somente os pólos de malha fechada. Desse modo, aplicando os seus critérios de desempenho, o método de Haalman para o cálculo dos parâmetros do controlador para uma sistema com função de transferência do tipo 2.4 pode ser visto na Tabela 2.6.

Tabela 2.6: Parâmetros do controlador para o método de Haalman.

Tipo de Controlador	K_p	T_i	T_d
PI	$\frac{2T}{3kL}$	T	–

Pode-se perceber que o tempo de integração é o mesmo para o método Lambda e que o ganho proporcional é o mesmo para o caso em que seja escolhido $T_{cl} = L/2$ para o mesmo método[4].

2.2.6 Método SIMC

O método SIMC (*Skogestad's Internal Model Controller*), que também é um método algébrico de sintonia, é uma versão do modelo de controle interno (IMC), o qual também pode ser aplicado a sistemas com função de transferência do tipo 2.4. Desse modo, procurando corrigir os inconvenientes do método IMC, Skogestad[8] propôs as regras de sintonia mostradas na Tabela 2.7.

Tabela 2.7: Parâmetros do controlador para o método SIMC.

Tipo de Controlador	K_p	T_i	T_d
PI	$\frac{T}{2kL}$	$\min(T, 8L)$	–



Figura 2.2: Diagrama de aplicação de CLP. (Modificado de: ver [9])

2.3 Controlador Lógico Programável (CLP)

Os Controladores Lógico Programáveis (CLP's), os quais podem ser considerados pertencentes à família dos computadores, usam circuitos integrados em vez de dispositivos eletromecânicos para implementar funções de controle. Eles são capazes de armazenar instruções, como sequenciamento, temporização, contagem, manipulação de dados e comunicação para realizar o controle de máquinas e processos industriais [9]. Na Figura 2.2 é mostrado um diagrama ilustrativo da aplicação dos CLP's.

De outro modo, o CLP pode ser definido como um computador de aplicação industrial especialmente desenvolvido, tanto na sua unidade central quanto na sua interface de comunicação com dispositivos de campo, para suportar situações que seriam nocivas a computadores comuns.

Tendo sido desenvolvido como alternativa ao uso dos imensos painéis à base de relés para realizar o controle de processos, os CLP's modernos apresentam diversas ferramentas que podem ser de grande utilidade para o profissional de automação. Dentre elas, é possível citar algumas que foram utilizadas neste trabalho: praticidade para programação, a execução interna de controladores PID e a troca de informações com outros dispositivos localizados em uma mesma rede ethernet.

Embora algumas linguagens de programação, como BASIC e C, tenham sido implementadas em alguns módulos de controladores, a programação da maioria dos CLP's é feita por meio da linguagem Ladder. Além desta, muitos dispositivos são programados usando a linguagem Booleana e Grafset.

A linguagem Ladder foi desenvolvida baseada nos diagramas dos painéis à relé utilizados para o controle de máquinas e processos antes dos controladores. Estes diagramas eram feitos de modo a representar a sequencia lógica do controle, mostrando a atuação de cada relé em seu escopo. Partindo dessa relação, o Ladder fornece ao programador uma interface gráfica de alto nível onde, por meio de símbolos, este pode estabelecer a lógica do comportamento esperado do controlador, onde cada linha do diagrama é executada após a anterior. Daí o nome da linguagem (Ladder - escada). Na Figura 2.3 é mostrado um exemplo de diagrama em Ladder.

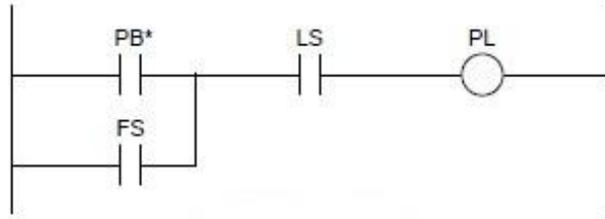


Figura 2.3: Representação de um diagrama em Ladder. (Extraído de: ver [9].)

2.4 Comunicação OPC

A comunicação OPC foi desenvolvida para atender a necessidade de comunicação entre sistemas. Antes dessa tecnologia, cada fabricante desenvolvia seus sistemas com drivers específicos, o que dificultava a interação entre sistemas de fabricantes diferentes. Caso uma indústria não quisesse comprar todos os seus sistemas de um mesmo fornecedor, isso demandaria tempo e grande esforço de programação para estabelecer a comunicação entre os dispositivos. Como é de se supor, grandes quantias em dinheiro eram dispendidas para a solução do problema.

Neste contexto, um grupo de empresas se reuniu para estabelecer um padrão de comunicação para acesso a dados em tempo real, padrão este baseado na tecnologia da Microsoft OLE/DCOM. OLE (Object Link and Embedding) é uma tecnologia que define um procedimento padronizado em que um módulo cliente e um módulo servidor podem se comunicar através de uma determinada interface. Como fruto deste esforço, surgiu o padrão OPC (OLE for Process Control), que é uma interface de comunicação entre módulos de software. Esta interface padrão permitiu a redução de esforços por parte dos fabricantes no desenvolvimento de drivers para cada hardware específico e na atualização daqueles para cada nova versão destes. No caso dos clientes, estes possuem maior liberdade de escolha para comprar dispositivos de fornecedores convenientes. O padrão OPC começou a ser desenvolvido em 1995, tendo vindo a público a sua primeira versão em 1996.

A arquitetura OPC é um modelo Cliente-Servidor, onde o Servidor OPC fornece uma interface para objetos OPC e os gerencia. O servidor pode ser um código local ou remoto que é responsável pela coleta eficiente de dados de um dispositivo físico. Um cliente OPC se comunica com um servidor OPC através de interfaces OPC específicas. Essa arquitetura pode ser vista na Figura 2.4.

À medida que o uso do OPC cresceu, sendo aplicado nos mais diversos ambientes, novas necessidades surgiram, exigindo que a fundação OPC desenvolvesse interfaces para a troca de dados específicos. Isso acarretou na existência de várias especificações OPC distintas. De acordo com a definição do próprio órgão responsável (OPC Foundation): "Uma especificação OPC descreve uma interface de software para a troca de informações entre duas aplicações de software"[10].

De modo geral, as especificações OPC são agrupadas de acordo com o tipo de dados a serem trocados através da interface estabelecida. Por exemplo, as especificações *Data Access* são usadas para monitoramento em tempo real e para a coleta de dados de um processo, enquanto as especificações *Alarms and Events* são usadas para a detecção de eventos ou para a identificação de algum comportamento anormal em alguma parte do

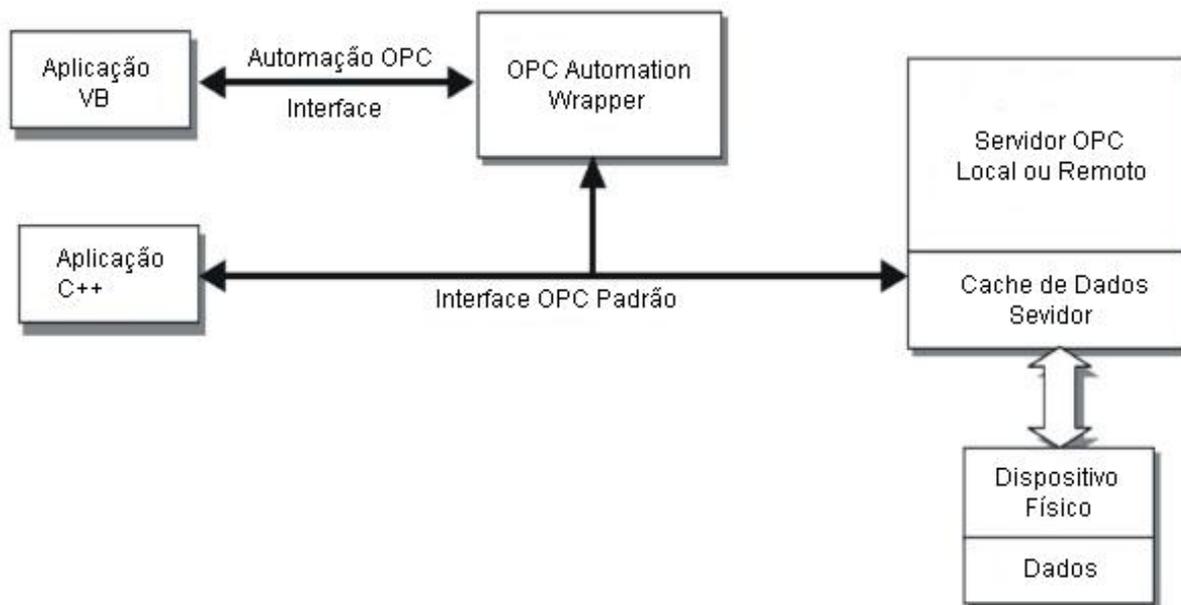


Figura 2.4: Acesso aos servidores OPC usando as interfaces Padrão e de Automação.

sistema. Como resultado disso, um produto OPC individual pode necessitar implementar mais de um tipo de especificação caso seja necessário trabalhar com vários tipos de dados[10]. Algumas especificações existentes hoje, além das duas citadas, são:

- OPC Overview: Descrição geral dos campos de aplicação das especificações OPC;
- OPC Common Definitions and Interfaces: Definição das funcionalidades básicas para as demais especificações;
- OPC Historical Data Access: Definição da interface para acesso a dados históricos;
- OPC Batch: Definição da interface para acesso aos dados de processos por batelada (batch). É uma extensão do OPC Data Access;
- OPC and XML: Integração entre OPC e XML para aplicações via Internet (web);

Capítulo 3

Modelagem

3.1 O Evaporador

O sistema evaporador é comumente utilizado em alguns setores industriais, sendo de maior importância para indústrias como usinas de açúcar, produção de alumina, fabricação de papel, entre outras. O seu objetivo principal é aumentar a concentração de soluções através da evaporação de solvente. Na figura 3.1 é mostrado a representação de um evaporador típico.

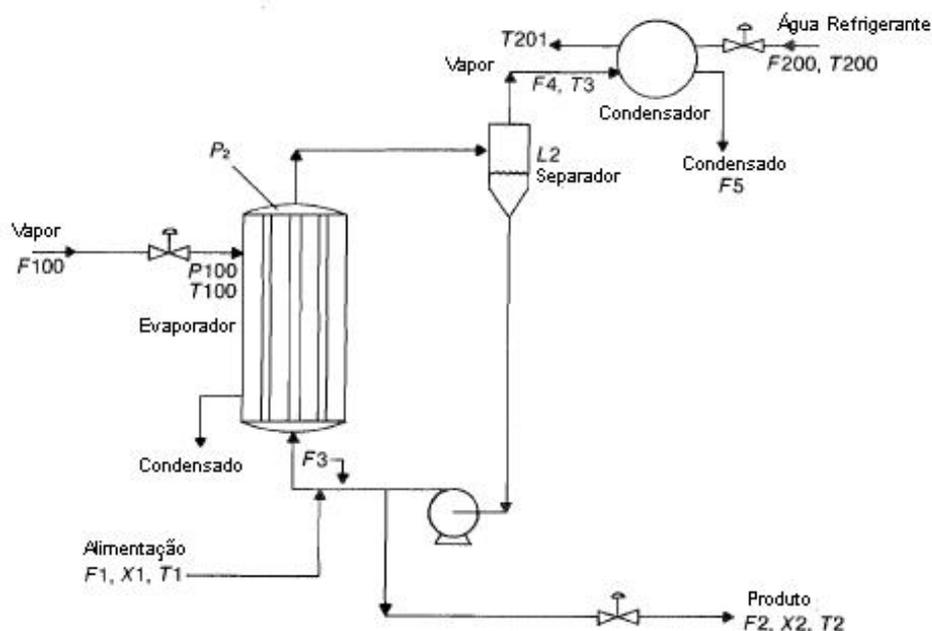


Figura 3.1: Evaporador Típico. (Modificado de: ver [1])

Ao entrar no sistema, o fluxo de alimentação $F1$, o qual é constituído de uma solução com concentração $X1$, é misturado ao licor que recircula a uma alta vazão ($F3$), sendo em seguida bombeado para o trocador vertical de calor. Normalmente, o trocador de calor é aquecido com vapor, o qual é injetado na jaqueta que o envolve, condensando em

seguida. Como o vapor da jaqueta não se mistura com o licor, a troca de calor é realizada por condução térmica através das paredes do tanque. Tendo sido aquecido até o ponto de ebulição, o conteúdo do tanque, agora constituído de líquido e vapor, passa então para o tanque separador. Vale salientar que, agora, o licor possui uma concentração maior que resultando da mistura de $F1$ e $F3$ já que ocorreu a evaporação do soluto.

O separador realiza uma função importante para o sistema. Nele, o vapor e o licor decorrido do trocador de calor são separados. Do licor obtido, uma parte é conduzida para a recirculação $F3$ e outra é retirada do sistema como produto $F2$. É interessante notar que o produto possui uma concentração desejada $X2$, já que uma parte do solvente foi retirada para o condensador. Neste último, o vapor é condensado através da troca de calor com um fluxo de material refrigerante, geralmente água.

O objetivo principal do evaporador é obter a concentração desejada para o produto final. Dessa forma, a variável a ser controlada é $X2$ (concentração do produto final). Para que isto ocorra, na construção do modelo um número finito de variáveis controláveis é definido: $F2$ (vazão de saída do produto final), $P100$ (pressão do vapor ingressante na jaqueta do evaporador) e $F200$ (vazão de entrada do líquido refrigerante). As outras variáveis de estado, além de $X2$, são $L2$ e $P2$.

Existem outras variáveis que afetam o sistema, mas para estabelecer o modelo com grau de liberdade zero, essas variáveis são consideradas como não controladas, ou seja, são consideradas como distúrbio. São elas: $F3$ (vazão do licor que recircula no sistema), $F1$ (fluxo de entrada), $X1$ (concentração do licor ingressante), $T1$ (temperatura do licor ingressante) e $T200$ (temperatura do líquido refrigerante). O restante das variáveis do sistema está ligado às equações do modelo.

3.2 O Modelo

O modelo para o evaporador pode ser estabelecido através de várias equações, as quais relacionam as variáveis citadas anteriormente. O sistema pode ser dividido em partes. São elas:

3.2.1 O Separador

O balanço de massa para o líquido total (solvente e soluto) contido no sistema nos permite estabelecer a seguinte equação:

$$\rho A \frac{dL2}{dt} = F1 - F4 - F2 \quad (3.1)$$

Onde: ρ é a densidade do líquido e A é a área da seção do separador.

A equação relaciona o nível do tanque do separador $L2$ com os fluxos de entrada e saída do sistema. Como $F3$ recircula constantemente e $F100$ e $F200$ não possuem contato direto com o licor, logo não influenciam para o balanço de massa do evaporador, pode-se considerar que o único fluxo de entrada é $F1$. No separador, o vapor decorrente do trocador de calor é separado da parte líquida para que possa ser condensado ($F4$), o que ocasiona em uma perda de massa para o sistema. Do fluxo que sai do separador, certa quantidade é retirada ($F2$) como produto final, enquanto o restante continua recirculando.

3.2.2 O trocador de Calor

O trocador de calor é de grande importância a eficácia do Evaporador. É nele que a idéia principal do sistema como um todo é posta em prática, ou seja, a evaporação do solvente para o aumento da concentração.

Uma importante variável para a análise do separador é a sua pressão de operação. Realizando-se o balanço de massa para o vapor contido no processo é possível obter uma expressão para esta variável:

$$C \frac{dP2}{dt} = F4 - F5 \quad (3.2)$$

Na expressão 3.2, C é uma constante que converte a massa do vapor em uma pressão equivalente e possui um valor de $4kg/kPa$. Por esta expressão, pode-se ver que a pressão P2 depende da saída de vapor do separador (F4) e da quantidade de vapor condensado (F5). Isso se dá pelo fato de que o único meio de saída do processo para o vapor é através da condensação do mesmo após refrigeração. Como o vapor do trocador de calor sai diretamente para o separador, torna-se natural que o balanço de massa para o mesmo seja a quantidade retirada do separador menos a parte condensada.

Prosseguindo-se na análise do vapor no sistema, pode-se obter as seguintes expressões:

$$T3 = 0,507P2 + 55,0 \quad (3.3)$$

$$F4 = \frac{Q100 - F1C_p(T2 - T1)}{\lambda} \quad (3.4)$$

A expressão 3.3 é a linearização da curva de saturação líquida para a água. Nela podemos perceber que a temperatura do vapor que sai do separador (T3) depende diretamente da pressão de funcionamento do processo. Já a expressão 3.4 mostra o fluxo de vapor, o qual foi produzido no trocador. É interessante notar que este fluxo é diretamente proporcional a potência do trocador de calor (Q100), bem como inversamente proporcional ao calor latente de vaporização do licor (λ), ou seja, dependendo da substância que compõe o licor, mais ou menos vapor será obtido. Como pode ser visto na expressão, o fluxo também é influenciado pela capacidade térmica do licor (C_p) e, como era de se esperar, a variação de temperatura para o mesmo é a temperatura de saída do produto menos a temperatura de ingresso no sistema T1.

Os valores das constantes usadas estão mostradas na Tabela 3.1.

Constante	Valor	Unidade
C_p	0,07	$kW/K(kg/min)$
M	20	kg
λ	38,5	$kW/(kg/min)$

3.2.3 A Jaqueta

Como foi falado anteriormente, o meio usado para elevar a temperatura do licor contido no trocador de calor até a temperatura de ebulição é a troca de calor entre o vapor ingressante na jaqueta que envolve o tanque e o líquido. A temperatura do vapor pode ser modelada pela equação 3.5:

$$T100 = 0,1538P100 + 90,0 \quad (3.5)$$

Por esta equação, a qual é uma aproximação, podemos perceber que a temperatura depende diretamente da pressão do vapor. Como $P100$ é uma das variáveis controláveis do sistema, através do seu controle pode-se determinar indiretamente a troca de calor entre a jaqueta e o tanque, conseqüentemente, a temperatura do licor.

Como foi falado anteriormente, $Q100$ pode ser entendido como a potência da jaqueta, a qual é estabelecida pela expressão 3.6:

$$Q100 = UA1(T100 - T2) \quad (3.6)$$

Como a jaqueta pode ser considerada a única entrada de calor no sistema no qual o licor está contido, podemos considerar que a temperatura da saída do trocador de calor até o produto final é a mesma, ou seja, $T2$. Assim, a variação de temperatura, considerando que o vapor e o licor entraram em equilíbrio térmico, é a temperatura de ingresso na jaqueta menos a temperatura de saída do trocador de calor.

Na equação 3.6, $UA1$ é o coeficiente de transferência de calor vezes a área pela qual há a transferência. Esta variável é função do fluxo de entrada no trocador de calor, ou seja, $F1 + F3$. Assim,

$$UA1 = 0,16(F1 + F3) \quad (3.7)$$

Tendo entendido isso, pode-se estabelecer uma equação para o fluxo de entrada do vapor na jaqueta:

$$F100 = \frac{Q100}{\lambda_s} \quad (3.8)$$

λ_s é o calor latente do vapor, e o seu valor é de $36,6kW/(kg/min)$.

Como era de se esperar, de acordo com a equação 3.8, o fluxo de entrada de vapor na jaqueta depende da capacidade de transferência de calor por unidade de tempo e do calor latente do mesmo. Dessa forma, ao mudar-se essas características, o controle do fluxo mantém o sistema funcionando no modo desejado.

3.2.4 O Condensador

O condensador possui a função de condensar o vapor que sai do separador. Para isso, é usado um líquido a temperatura ambiente, o qual, através da condução térmica, absorve calor do vapor, fazendo com que este condense. No condensador, o fluxo de entrada ($F200$) é tratado como variável controlável, enquanto a temperatura de ingresso do líquido é considerada como sendo distúrbio.

O fluxo de líquido refrigerante faz o trabalho inverso do vapor na jaqueta, ou seja, no trocador de calor, tinha a função de aquecer o licor contido no tanque, aqui, o líquido refrigera o vapor. Da mesma forma que no sistema jaqueta mais tanque pôde ser estabelecida uma equação para a potência do mesmo, isso pode ser feito da mesma forma para o condensador.

$$Q_{200} = \frac{UA_2(T_3 - T_{200})}{1 + UA_2/(2C_p F_{200})} \quad (3.9)$$

A equação 3.9 foi obtida, após algumas manipulações, partindo do balanço de energia para o líquido refrigerante. Nela, UA_2 representa o coeficiente de transferência de calor vezes a área pela qual há a transferência, similarmente a UA_1 . Da mesma forma do caso do vapor na jaqueta, considerando que o líquido condensado e o refrigerante ao final se encontram em equilíbrio térmico, pode-se perceber que a potência depende da diferença entre a temperatura ingressante e a temperatura final do refrigerante. Para este caso, a transferência de energia no tempo é inversamente proporcional à capacidade calorífica C_p e ao fluxo de ingresso F_{200} . Como este último é manipulável, é possível, através do seu controle, alterar o valor de Q_{200} .

A temperatura de saída do refrigerante é regida pela equação 3.10:

$$T_{201} = T_{200} + \frac{Q_{200}}{F_{200}C_p} \quad (3.10)$$

Como era de se esperar, a temperatura T_{201} depende da temperatura inicial do líquido e da energia absorvida durante o processo, a qual depende da potência, do fluxo e da capacidade calorífica. Vale notar que o controle de F_{200} também interfere na temperatura final do refrigerante.

Por fim, o fluxo de saída do vapor condensado é estabelecido pela equação 3.11:

$$F_5 = \frac{Q_{200}}{\lambda_s} \quad (3.11)$$

É possível perceber, de modo bastante simples, que a quantidade de vapor condensado depende diretamente da taxa de transferência de calor e é inversamente proporcional ao calor latente de vaporização. Para este sistema, como a água é considerada como sendo o refrigerante, o valor de λ_s é $38,5 \text{ kJ/K} (\text{kg/min})$.

3.2.5 O Produto

O balanço de massa para o soluto contido no processo pode nos dar uma expressão para a concentração do produto, como pode ser visto na equação 3.12:

$$M \frac{dX_2}{dt} = F_1 X_1 - F_2 X_2 \quad (3.12)$$

onde M é a quantidade de líquido no evaporador e pode ser considerada constante em 20 kg .

Como o soluto só ingressa no processo através da vazão de alimentação e só sai no produto final, torna-se claro o porquê de somente estas duas rotas serem consideradas na expressão.

3.2.6 Equações de Estado

As equações de estado para o sistema evaporador são, portanto:

$$\rho A \frac{dL2}{dt} = F1 - F4 - F2 \quad (3.1)$$

$$C \frac{dP2}{dt} = F4 - F5 \quad (3.2)$$

$$M \frac{dX2}{dt} = F1X1 - F2X2 \quad (3.12)$$

Uma vez obtido o modelo matemático para o sistema evaporador, o mesmo foi simulado usando o *software* Matlab. O código do programa desenvolvido está mostrado no Apêndice A.

Capítulo 4

Identificação do Sistema e Projeto dos Controladores

Em muitas situações, o comportamento de um determinado processo pode não estar totalmente disponível ao projetista dos controladores ou pode não ser viável o desenvolvimento matemático de modelos. Além disso, mesmo na existência de modelos mais complexos, como o descrito no capítulo anterior, eles podem não ser de grande utilidade para o projeto dos controladores. Uma solução comumente utilizada para situações como essas é o uso de técnicas de identificação de sistemas.

Uma forma comum de identificação de sistemas é a obtenção de funções de transferência a partir da resposta de um sistema a um degrau aplicado em sua entrada. A partir de então, uma vez que dados referentes a planta estão disponíveis, torna-se simples o uso de técnicas de sintonia para controladores PID. Este foi o procedimento realizado neste trabalho para realizar o projeto dos controladores.

No capítulo capítulo foi descrito o modelo deduzido no trabalho de Neweel e Lee [1]. Nele, foi mostrado que o sistema evaporador modelado possui três variáveis controláveis de entrada e três variáveis de saída a serem controladas. Embora isso seja verdade, neste trabalho foi escolhido, para a aplicação das técnicas propostas, uma única variável de entrada ($F2$) e duas variáveis a serem controladas ($X2$ e $P2$), sendo obtidas as funções de transferência as relacionam e respectivos controladores usados para controlar as malhas.

4.1 Identificação do Sistema

Como descrito anteriormente, diversas técnicas de sintonia para controladores usam as funções de transferência do sistema como fonte das informações necessárias para o projeto. Para tanto, usou-se a técnica da resposta ao degrau para obter as funções de transferência que relacionam a concentração do produto ($X2$) e o seu fluxo de saída ($F2$), e a pressão de operação ($P2$) e o fluxo de saída ($F2$).

Inicialmente, aplicando-se um degrau de 2 kg/min em $F2$ temos as respostas mostradas nas Figuras 4.1 e 4.2.

Analisando-se os dados obtidos, foi possível obter alguns parâmetros a partir deles. O primeiro foi o ganho k , o qual é dado por $k = \frac{\Delta y}{\Delta x}$, onde Δy é a variação do sinal de saída e Δx a variação do sinal de entrada, ou seja, a amplitude do degrau aplicado. Em segundo

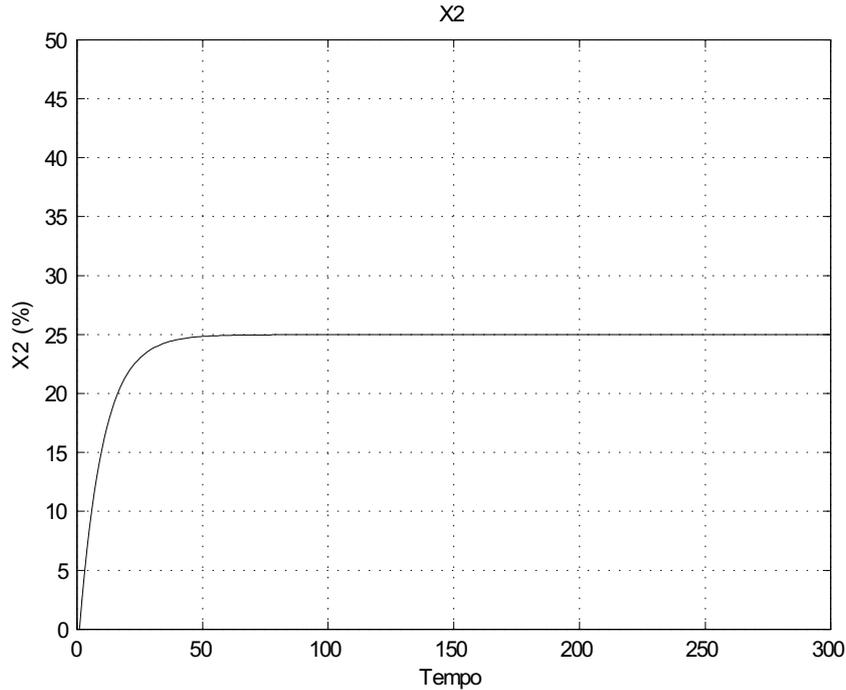


Figura 4.1: Concentração X2 com degrau em F2.

lugar, foi possível obter a constante de tempo T , que é o tempo em que $y = 63\% \Delta y$. Por fim, pôde-se constatar que o atraso de transporte é $\theta = 1$ para ambos os casos. Desse modo, os modelos obtidos para o procedimento foram do tipo primeira ordem com atraso, como mostrado na expressão 4.1, a qual está de acordo com a expressão 2.4, descrita anteriormente:

$$G(s) = \frac{k \cdot e^{-Ls}}{Ts + 1} \quad (4.1)$$

Assim, seguindo o procedimento descrito, encontrou-se os parâmetros $\Delta x = 2$, $\Delta y = 25$, $k = 12,5$ e $T = 10$ para a função de transferência G_1 , a qual relaciona $X2$ e $F2$. Do mesmo modo, encontrou-se os valores $\Delta x = 2$, $\Delta y = 54$, $k = 27$ e $T = 23,1$ para a função de transferência G_2 , a qual relaciona $P2$ e $F2$. Desse modo, temos as seguintes funções de transferência:

$$G_1(s) = \frac{12,5e^{-s}}{10s + 1} \quad (4.2)$$

$$G_2(s) = \frac{27e^{-s}}{23,1s + 1} \quad (4.3)$$

A partir das expressões 4.2 e 4.3 é possível usar as diferentes técnicas citadas no capítulo 2 para o projeto dos controladores, bem como verificar qual deles apresenta o melhor comportamento.

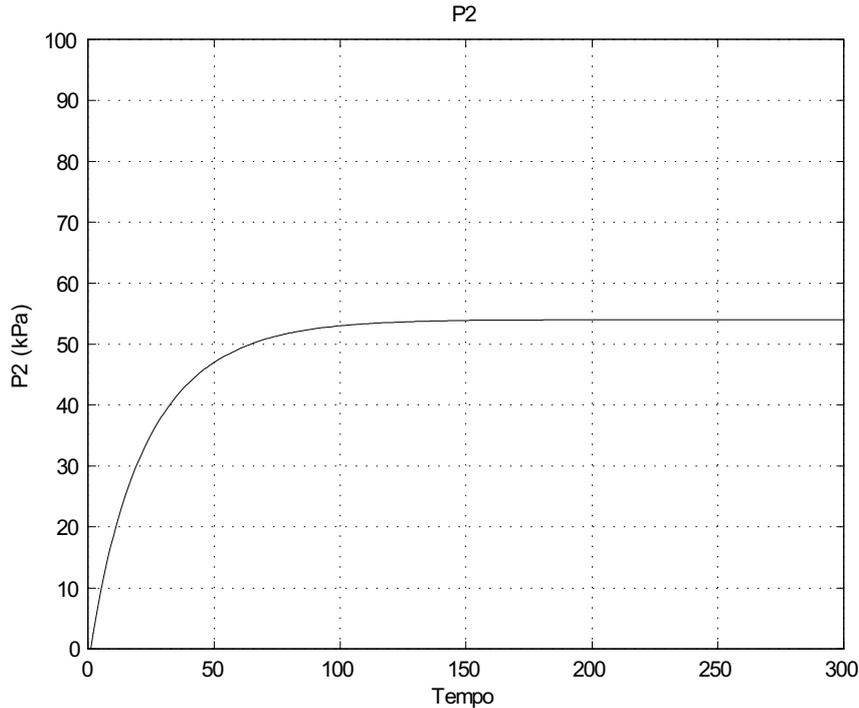


Figura 4.2: Pressão de operação P2 com degrau em F2.

4.2 Projeto dos Controladores

O projeto dos controladores foi feito seguindo os procedimentos estabelecidos pelas diversas técnicas mostradas no capítulo 2. Para tanto, os valores dos parâmetros para cada método foram calculados a partir das informações contidas nas funções de transferência. Para efeito comparativo, cada controlador foi simulado juntamente com o processo na configuração de malha fechada, o que pode nos indicar qual apresentaria melhor comportamento no processo real.

Vale salientar que os resultados foram obtidos fazendo o uso da ferramenta do Matlab Simulink, onde foi estabelecido o modelo de malha fechada para o processo, como pode ser visto na Figura 4.3. Vale destacar também que, para análise do comportamento do controlador, foram aplicados diferentes valores de referência para os processos. No caso da concentração X_2 , foi aplicado inicialmente, como referência, um degrau de valor 25, sendo que no instante 50, esse valor foi reduzido para 20. Para o caso da pressão P_2 , foi aplicado, inicialmente, um valor de referência de 50, sendo que, no instante 50, este valor foi reduzido para 40. Estes valores escolhidos se baseiam nos valores de regime permanente esperados para as variáveis no funcionamento da planta.

Os melhores resultados estão mostradas nas Figuras 4.4 e 4.5.

É interessante notar que os melhores resultados foram apresentados por controladores do tipo PI. Isso se dá pelo tipo de sistema a ser controlado, no qual o uso do termo derivativo prejudica o desempenho do controle.

Um meio de se analisar a eficiência de controladores é o uso de critérios de desempenho, como o IAE, ISE e ITAE. Para a comparação entre os controladores desenvolvidos foram os usados os critérios IAE (Integrated Absolute Error), o qual consiste na integral do

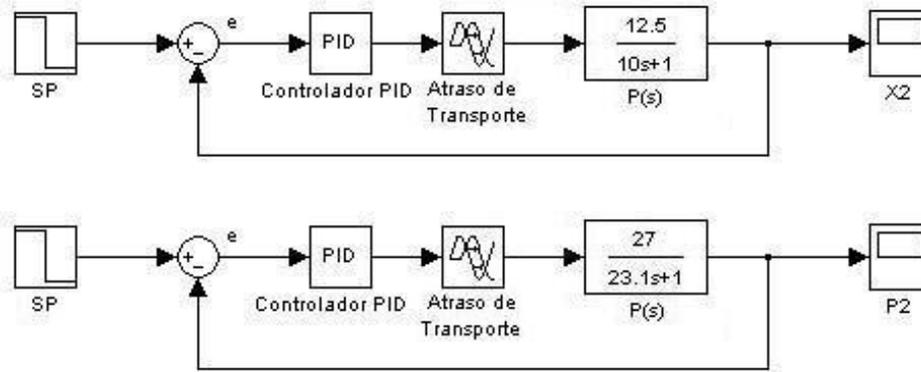


Figura 4.3: Modelos em malha fechada (Simulink)

módulo do erro ao longo da simulação, e ISE (Integrated Square Error), que consiste na integral do quadrado do erro. Desse modo, quanto menores os valores para estes critérios, mais eficientes podem ser considerados os controladores. Nas Tabelas 4.1 e 4.2 estão mostrados os melhores resultados para o controle da concentração X_2 e da pressão P_2 .

Tabela 4.1: Critérios de desempenho para diversos controladores - X_2 .

Método	IAE	ISE
Zigler-Nichols	109,6	1439
Lambda	68,6	1154
Halmaan	64,4	1014
SIMC	75,4	1108

Tabela 4.2: Critérios de desempenho para diversos controladores - P_2 .

Método	IAE	ISE
Zigler-Nichols	252,2	6687
Lambda	131,3	4421
Halmaan	129,1	4055
SIMC	168,6	4683

Ao se realizar a comparação entre os critérios de desempenho obtidos, pode-se concluir que o melhor método de sintonia, segundo os resultados teóricos, para controle da concentração X_2 é o método de Halmaan, uma vez que este apresentou os menores valores para o IAE e o ISE, e o método com desempenho menos satisfatório foi o de Ziegler-Nichols, apresentando os maiores valores. Quanto ao controle da pressão de operação P_2 , os resultados se repetiram, pois quem apresentou o melhor e o pior resultado, respectivamente, foram os métodos de Halmaan e o de Ziegler-Nichols.

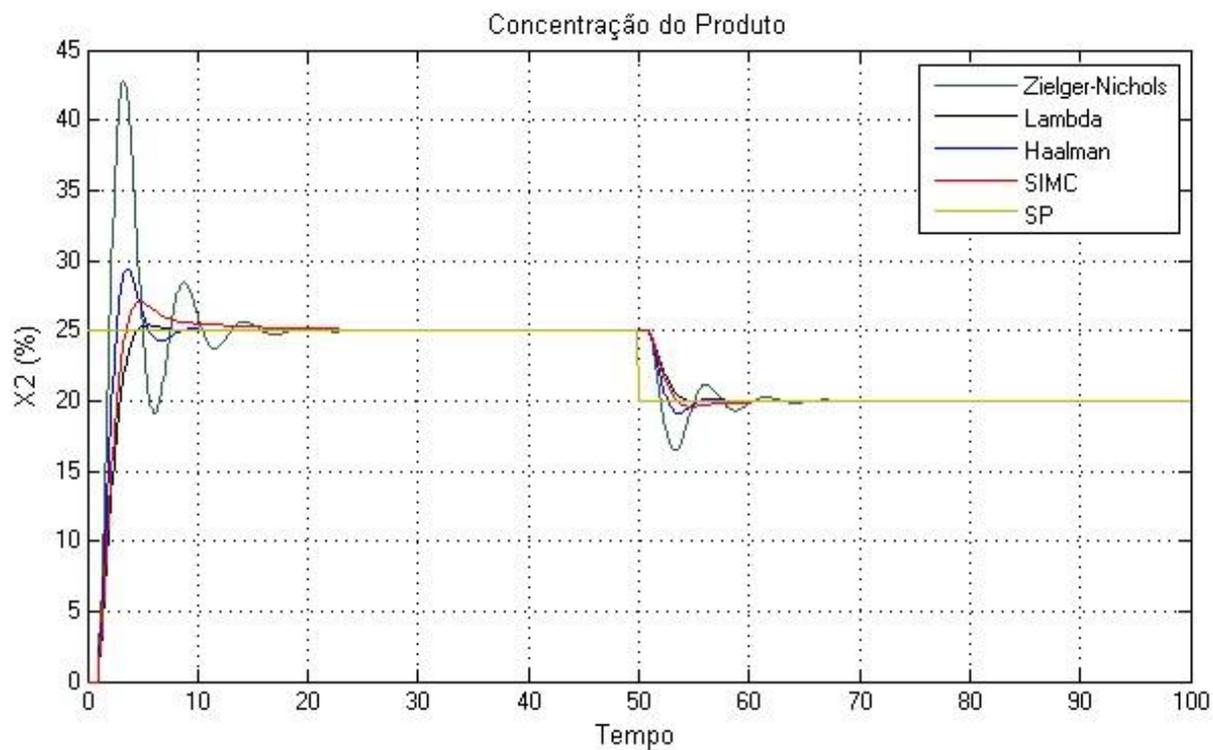


Figura 4.4: Melhores resultados para o controle da concentração X2.

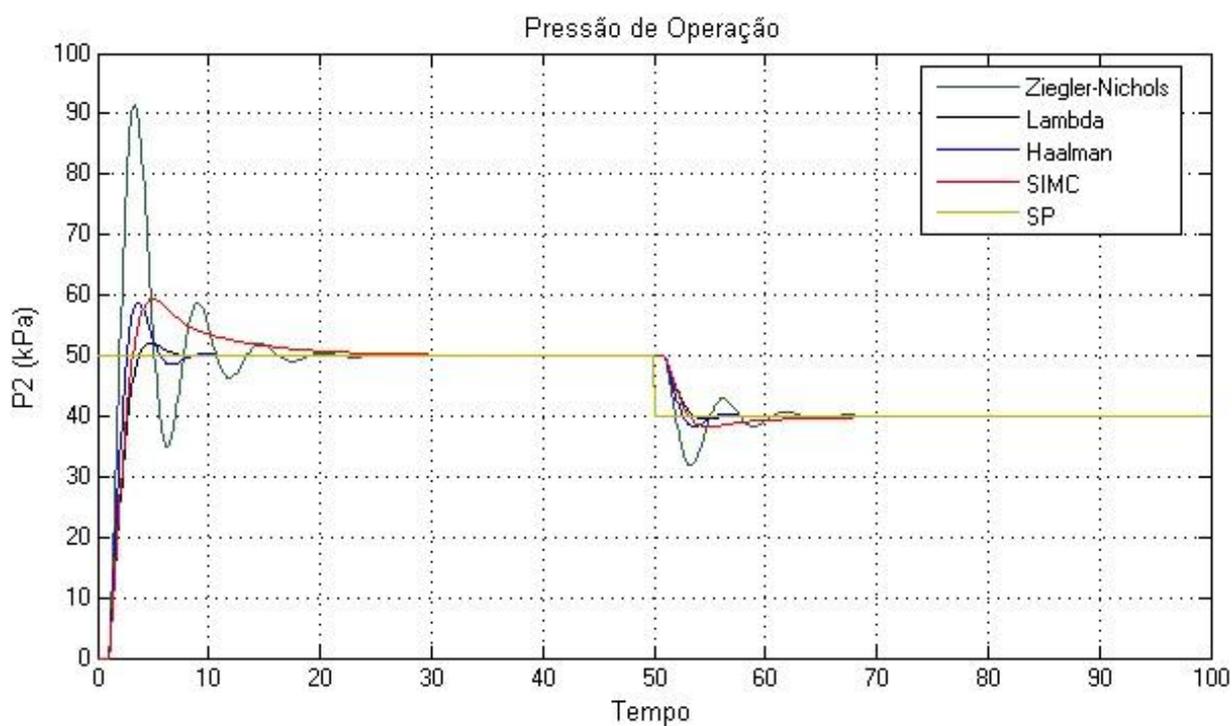


Figura 4.5: Melhores resultados para o controle da pressão de operação P2.

Capítulo 5

Resultados Obtidos Com o Controlador PID Real

5.1 *Hardware in the Loop*

No caso deste trabalho, tanto o servidor quanto o cliente OPC foram executados no mesmo computador. Um servidor OPC é um aplicativo de software que atua como um API (*Application Programming Interface*). O servidor OPC se conecta a um dispositivo, como um PLC, ou uma fonte de dados como um banco de dados, e traduz os dados em um formato padrão OPC. No caso do cliente, quem exerceu esta função foi a rotina do Matlab que executou o simulador, o qual se conectou ao servidor OPC, permitindo tanto a leitura quanto a escrita de dados. No caso do servidor, quem desempenhou o papel foi o software RSLinx, da Rockwell Automation, o qual realizou a comunicação com o CLP. Desse modo, como pode ser percebido, a interface OPC serviu para a comunicação entre sistemas de fabricantes diferentes: uma rotina Matlab e um CLP Rockwell. A representação para a comunicação cliente-servidor para a plataforma está mostrada na Figura 5.1.

A comunicação OPC usada neste projeto teve em vista a simulação do controle do sistema evaporador. Para tanto, o simulador desenvolvido no ambiente Matlab serviu como modelo da planta, enquanto o CLP permitiu o uso do controlador PID, realizando os devidos cálculos das variáveis de controle. Desse modo, a saída do simulador é escrita no CLP como o PV (*Process Variable*) do controlador e, após os cálculos realizados, a saída do controlador (MV) é lida pelo Matlab como entrada da planta, realizando assim o controle devido.

Para a primeira malha obtida na identificação de sistemas, a variável de entrada da planta é o fluxo de ingresso no sistema $F2$, que é lido do CLP como sendo MV, enquanto a variável a ser controlada é a concentração do produto final $X2$, que é lida pelo CLP como sendo PV. No caso da segunda malha obtida na identificação de sistemas, a entrada do processo é $F2$ e a saída é a pressão de operação do sistema $P2$, sendo que o controle é realizado do mesmo modo. A representação do sistema em malha fechada está mostrada na Figura 5.2.

Vale destacar que no caso do controle da concentração de saída $X2$, o erro foi calculado como sendo $e = PV - SP$, enquanto no caso do controle da pressão de operação $P2$ o erro foi calculado do modo mais comum, ou seja, $e = SP - PV$. Isso se deve ao fato de

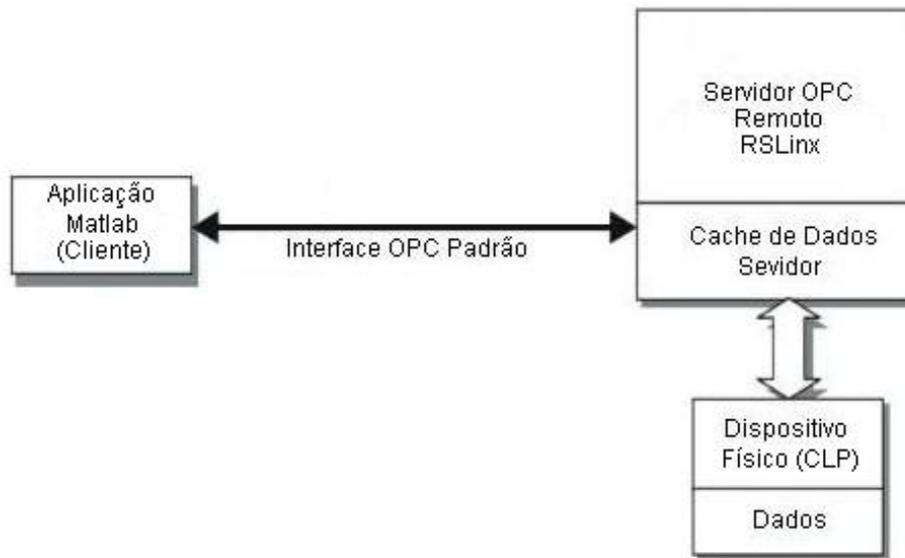


Figura 5.1: Comunicação cliente-servidor

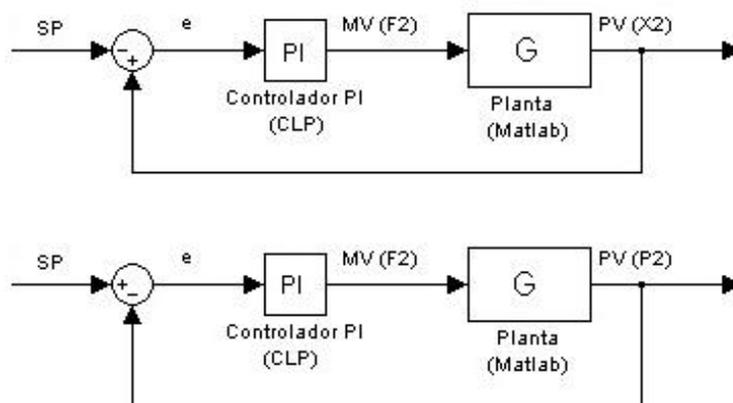


Figura 5.2: Representação dos sistemas em malha fechada.

que, no primeiro caso, ao se aumentar a entrada na planta, a variável de saída tem o seu valor decrescido, o que exige o uso de controle inverso.

5.2 Dados Coletados

Uma vez efetuada a comunicação OPC descrita anteriormente, ajustou-se os valores dos parâmetros do PID, executado no CLP, de acordo com os valores obtidos através dos diferentes métodos de sintonia, a fim de realizar o controle da planta simulada. Como dentre os quatro melhores métodos, o método de Zigler-Nichols foi o que apresentou resultados mais discrepantes, foram testados apenas os parâmetros obtidos pelos métodos Lambda, SIMC e de Haalman. Os resultados obtidos para a simulação do controle da

concentração X_2 referentes às diferentes sintonias estão mostrados na Figura 5.3. Os valores para os critérios de desempenho estão mostrados na Tabela 5.1.

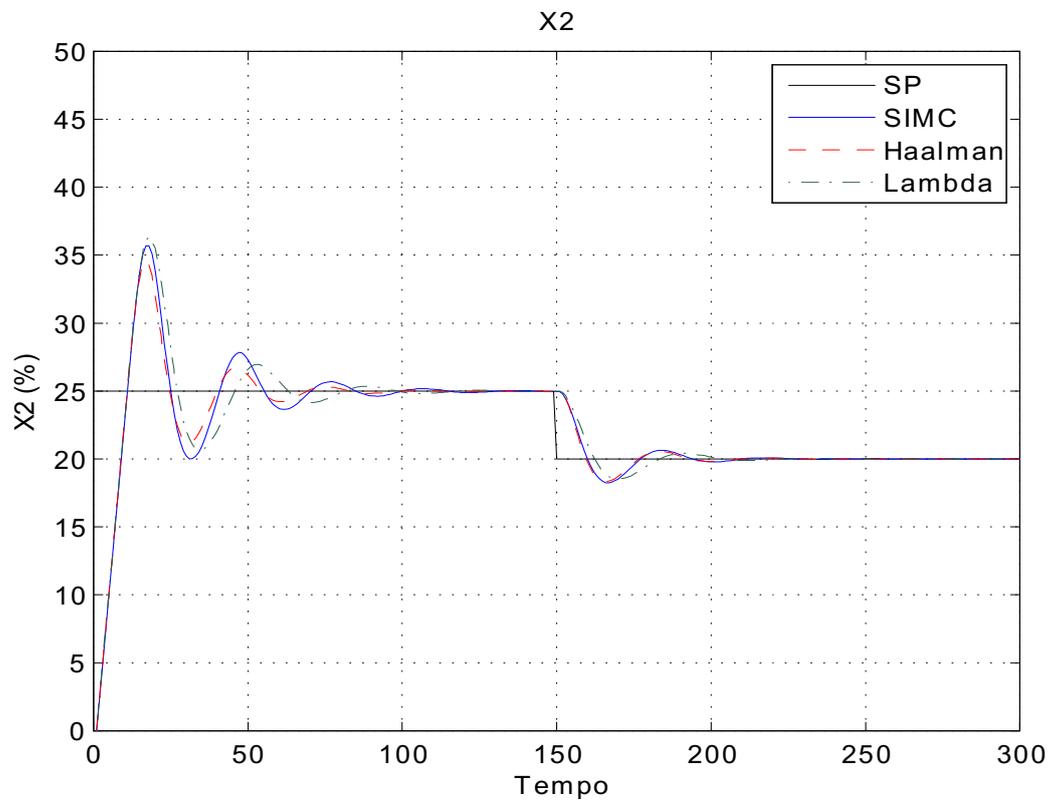


Figura 5.3: Resultados obtidos para diversos controladores para o controle da concentração X_2 .

Tabela 5.1: Critérios de desempenho para diversos controladores - X_2 .

Método	IAE	ISE
Lambda	380, 31	3293
Halmaan	327, 29	2974
SIMC	390, 26	3470

Como pode ser visto nos valores da Tabela 5.1, o método de sintonia que apresentou o melhor resultado foi o método de Halmaan, o que ratifica o resultado obtido na simulação durante o projeto dos controladores. Vale destacar que os valores do IAE e do ISE estão maiores que no caso da simulação em decorrência do uso de uma escala de tempo maior.

Para a simulação do controle da pressão de operação P_2 , foi usado um procedimento semelhante, mudando apenas o cálculo do erro, como foi descrito anteriormente, e os valores dos parâmetros do controlador PID. Os melhores resultados para as diferentes sintonias estão mostrados na Figura 5.4, enquanto os valores para os critérios de erro estão mostrados na Tabela 5.2.

Pode-se perceber que o resultados para o método SIMC não foram mostrados na figura e na tabela. Isso se deve ao fato de que o controlador sintonizado de acordo com este método apresentou instabilidade, diferentemente do previsto na simulação.

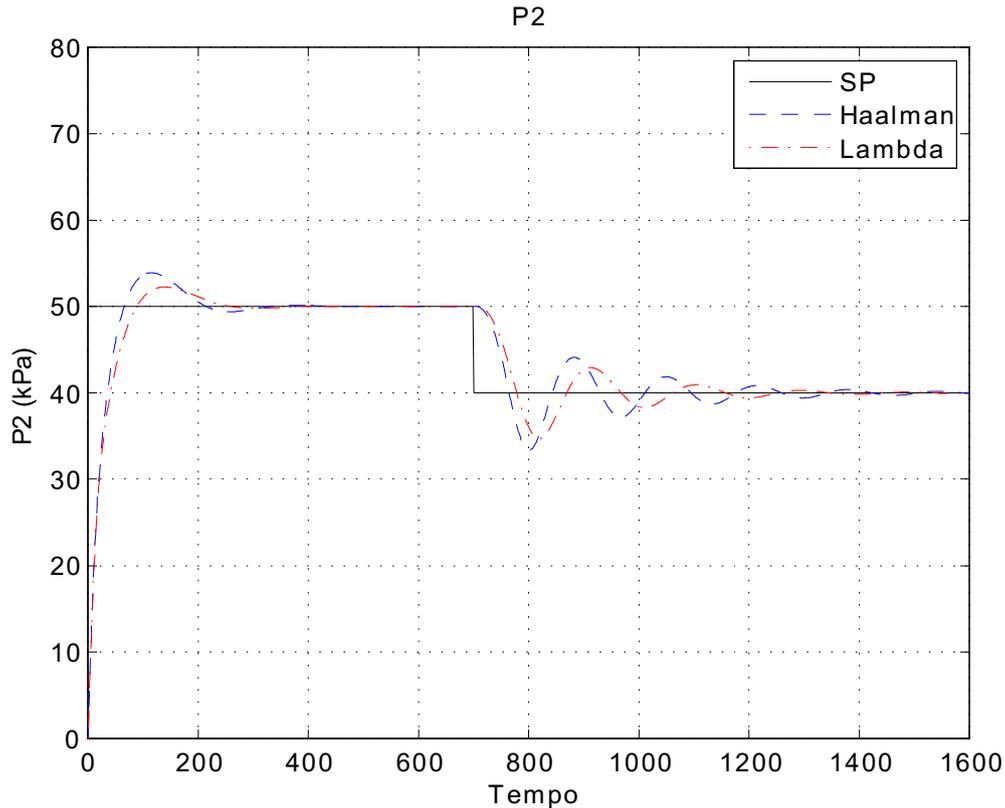


Figura 5.4: Resultados obtidos para diversos controladores para o controle da pressão de operação P2.

Tabela 5.2: Critérios de desempenho para diversos controladores - P2.

Método	IAE	ISE
Lambda	2608	34026
Halmaan	2807	33263

Quanto a comparação dos melhores métodos para o controle de $P2$, pode-se perceber que, assim como na simulação usando o Simulink, o método Lambda e o método de Haalman apresentaram critérios de desempenho com valores bastante próximos. No caso da simulação usando um PID real, o critério IAE apresentou valor mais apropriado para o método Lamba, enquanto o método de Haalman apresentou melhor resultado segundo o critério ISE. Uma vez que havia sido previsto comportamentos semelhantes para os dois métodos, pode-se concluir que o resultado corrobora com a previsão estabelecida anteriormente.

É importante destacar que os valores dos critérios de desempenho obtidos no controle de $X2$ foram bem menores que no caso de $P2$. Isso pode ser explicado ao perceber que no segundo caso foi usado uma escala de tempo diferente. Isso foi necessário para a realização da simulação, uma vez que o sistema apresentou instabilidade caso se usasse a mesma escala de tempo usada para o controle de $X2$. Como a rotina no Matlab é executada a uma velocidade várias vezes maior que a do CLP para atualizar o PID, existe o desafio de estabelecer a relação correta entre o período de execução dos dois sistemas.

Desse modo, o estabelecimento de uma comunicação mais precisa entre simulador executado no Matlab e o controlador PID executado no CLP fica proposto para um trabalho posterior que aborde o mesmo tema.

Apesar das dificuldades no estabelecimento da comunicação entre *hardwares* distintos, a partir dos resultados obtidos, pode-se perceber que o método de identificação de sistemas e os projetos dos controladores a partir das funções de transferência apresentaram resultados satisfatórios, uma vez que, considerando o simulador desenvolvido a partir do modelo como sendo suficientemente fiel ao processo real, o comportamento do processo em malha fechada com o acréscimo do controlador apresentou comportamento aproximado ao previsto.

Capítulo 6

Conclusão

Neste trabalho foi abordado o modelo do sistema evaporador usando um método de identificação de sistemas para o projeto de controladores, bem como o uso da técnica *hardware in the loop* para verificação dos resultados. Isso comprovou como este procedimento pode ser útil para o profissional de automação quando se fizer necessário realizar o controle de um processo industrial.

Além disso, mais uma vez ficou comprovada a utilidade do *software* Matlab para a realização de simulações em situações que experimentos com um sistema real é muito dispendioso ou inviável. Essa ferramenta se tornou ainda mais valiosa após o advento da interface OPC, que também foi abordada com êxito, a qual permite a comunicação entre sistemas de fabricantes distintos sem a necessidade de grandes labores no desenvolvimento de *drivers* específicos.

Também é necessário destacar que este trabalho pode servir como ponto de partida para projetos posteriores, uma vez que ele se limitou ao projeto e verificação de controladores usando as ferramentas citadas. Como sugestão, pode-se citar o desenvolvimento de um sistema supervisorio, uma vez que já foi desenvolvido o simulador para o controle do sistema em malha fechada.

Por fim, pode-se concluir que os objetivos propostos para este trabalho foram atingidos. Ao longo do mesmo, as técnicas propostas foram utilizadas de modo a contribuir para o fim principal, que foi a realização do controle do sistema evaporador. Além disso, o trabalho permitiu a familiarização com procedimentos que foram apresentados apenas de forma introdutória ao longo do curso, o que é de grande valia para o desenvolvimento profissional.

Referências Bibliográficas

- [1] NEWELL II, R.B.; LEE, P.L., *Applied Process Control, A Case Study*, New York: Prentice-Hall, p. 7-13, 1989.
- [2] CAO, Y., *Direct Gradient Control for Static Optimization*, 2004
- [3] HAUGWITZ, S., *Modeling, Control and Optimization of an Evaporator System a Project Description*, 2007
- [4] ÅSTRÖM, K. J.; HÄGGLUND, *Advanced PID Control*, ISA, p. 62, 2006
- [5] ZIEGLER, J.G.; NICHOLS, N.B., *Optimum Settings for Automatic Controllers*, Trans. ASME 64, pp. 759-768, 1942.
- [6] CHIEN, K.L.; HRONES, J.A.; RESWICK, J.B., *On the Automatic Control of Generalized Passive Systems*. Trans. ASME 74, pp 175-185, 1952.
- [7] COHEN, G.H.; COON, G.A., *Theoretical Consideration of Retarded Control*. Trans. ASME 75, pp 827-834, 1953.
- [8] SKOGESTAD, S., *Simple analytic rules for model reduction and PID controller tuning*, 2002
- [9] BRYAN, L. A.; BRYAN, E. A., *Programmable Controllers: Theory and Implementation*, 2nd ed., Industrial Text Company, Atlanta, 1997.
- [10] <http://www.opcfoundation.org/Products/Specifications.aspx?CM=-1> (Acesso em: 11 de maio de 2012)

Apêndice A

Código do Simulador do Evaporador (Matlab)

```
%Simulacao de um evaporador
%Baseado no livro "APPLIED PROCESS CONTROL - R. B. NEWELL"
clc
clear all
close all

%Variaveis do processo

%Entradas Controlaveis
F2 = 2;
P100 = 200;
F200 = 208.0;

%Entradas NAO Controlaveis (Disturbios)
F1 = 10.0;
F3 = 50.0;
X1 = 5.0;
T1 = 40.0;
T200 = 25.0;

%Variaveis internas( Valores iniciais )
F4 = 8.0;
F5 = 8.0;
T2 = 84.6;
T3 = 80.6;
F100 = 9.3;
T100 = 119.9;
Q100 = 339.0;
T201 = 46.1;
Q200 = 307.9;

%Parametros do processo
rhoA = 20;
```

```

M = 20;
C = 4;
Cp = 0.07;
lamb = 38.5;
lambS = 36.6;
lambA = 38.5;
UA2 = 6.84;

%Parametros da simulacao
stp = 1;

%Inicializacao das variaveis que serao integradas
L2 = 0;
X2 = 0;
P2 = 0;

%degraus
degrauF2=0;
degrauP100=0;
degrauF200=0;

for i = 1:5000

    %Equacoes algebricas
    %%%%%%%%%%
    %Balanco de energia
    T2(i) = 0.5616*P2(i) + 0.3126*X2(i) + 48.43;
    T3(i) = 0.0507*P2(i) + 55.0;

    %Equacoes referentes a jaqueta
    T100(i) = 0.1538*(P100+degrauP100) + 90.0;
    Q100(i) = 0.16*((F1+degrauF1) + F3)*(T100(i) - T2(i));
    F100(i) = Q100(i)/lambS;

    %Balanco de energia
    F4(i) = (Q100(i) - (F1+degrauF1)*Cp*(T2(i) - T1))/lamb;

    %Equacoes referentes ao condensador
    Q200(i) = (UA2*(T3(i) - T200))/(1+UA2/(2*Cp*(F200+degrauF200)));
    T201(i) = T200 + Q200(i)/((F200+degrauF200)*Cp);
    F5(i) = Q200(i)/lambA;

    %%%%%%%%%%

    %Equacoes diferenciais

    dL2dt = ((F1+degrauF1)-F4(i)-(F2+degrauF2))/rhoA;
    dX2dt = ((F1+degrauF1)*X1 - (F2+degrauF2)*X2(i))/M;

```

```

dP2dt = (F4(i) - F5(i))/C;

%Integracao numerica
L2(i+1) = integrate_euler(L2(i),dL2dt,stp);
X2(i+1) = integrate_euler(X2(i),dX2dt,stp);
P2(i+1) = integrate_euler(P2(i),dP2dt,stp);

end

figure(1)
plot(L2,'k'),grid,TITLE('L2')
axis([0 500 -100 200]);
xlabel('Tempo');
ylabel('L2 (m)');

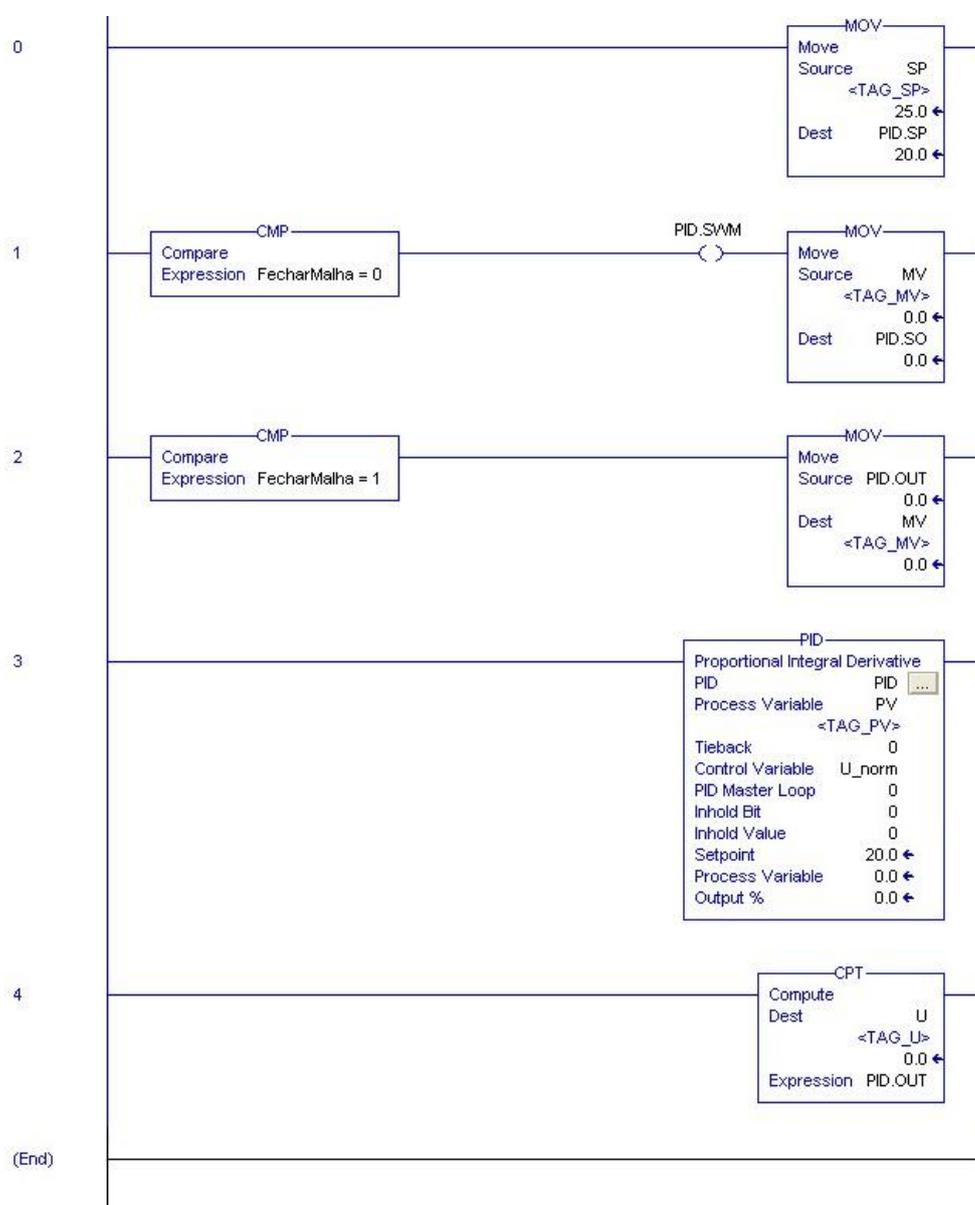
figure(2)
plot(X2,'k'),grid,TITLE('X2')
axis([0 500 0 50]);
xlabel('Tempo');
ylabel('X2 (%)');

figure(3)
plot(P2,'k'),grid,TITLE('P2')
axis([0 500 0 100]);
xlabel('Tempo');
ylabel('P2 (kPa)');

```

Apêndice B

Código em Ladder para o CLP



Apêndice C

Código do Simulador para a comunicação OPC (Matlab)

```
%Programa feito para comunicacao via OPC

clc
clear all
close all

%Conecta ao servidor e cria o grupo
server_clp = opcda('localhost', 'RSLinx Remote OPC Server');

connect(server_clp);

processPV = addgroup(server_clp);
processMV = addgroup(server_clp);
processSP = addgroup(server_clp);

%Adiciona item ao objeto server_clp e o associa as variaveis do CLP
% SP = additem(process,'[Peltier_compact]PID1.SP','double');
PV = additem(processPV,'[Peltier_compact]TAG_PV','double');
MV = additem(processMV,'[Peltier_compact]TAG_U','double');
SP = additem(processSP,'[Peltier_compact]TAG_SP','double');

%Variaveis do processo

%Entradas Controlaveis
P100 = 0*194.7;
F200 = 0*208.0;

%Entradas NAO Controlaveis (Disturbios)
F1 = 10.0;
F3 = 50.0;
X1 = 5.0;
T1 = 40.0;
T200 = 25.0;
```

```

%Variaveis internas( Valores iniciais )
F4 = 0*8.0;
F5 = 0*8.0;
T2 = 0*84.6;
T3 = 0*80.6;
F100 = 0*9.3;
T100 = 0*119.9;
Q100 = 0*339.0;
T201 = 0*46.1;
Q200 = 0*307.9;

%Parametros do processo
rhoA = 20;
M = 20;
C = 4;
Cp = 0.07;
lamb = 38.5;
lambS = 36.6;
lambA = 38.5;
UA2 = 6.84;

%Parametros da simulacao
stp = 1;

%Inicializacao de variaveis que serao integradas
L2 = 0*1;
X2 = 0*25;
P2 = 0*50.5;
F2 = 0;
%
n=1600;
tempo_amostragem = .1;
for i=1:n

    delay1 = cputime;
    delay2 = cputime;
    while (delay2-delay1)<=tempo_amostragem
        delay2=cputime;
    end

    if i == 700
        write(SP,40);
    end

F2 = MV.Value;

```

```

%Equacoes algebricas
%%%%%%%%%
%Balanço de energia
T2(i) = 0.5616*P2(i) + 0.3126*X2(i) + 48.43;
T3(i) = 0.0507*P2(i) + 55.0;

%Equacoes referentes a jaqueta
T100(i) = 0.1538*P100 + 90.0;
Q100(i) = 0.16*(F1 + F3)*(T100(i) - T2(i));
F100(i) = Q100(i)/lambS;

%Balanco de energia
F4(i) = (Q100(i) - F1*Cp*(T2(i) - T1))/lamb;

%Equacoes referentes ao condensador
Q200(i) = (UA2*(T3(i) - T200))/(1+UA2/(2*Cp*F200));
T201(i) = T200 + Q200(i)/(F200*Cp);
F5(i) = Q200(i)/lambA;

%%%%%%%%%

%Equacoes diferenciais

dL2dt = (F1-F4(i)-F2)/rhoA;
dX2dt = (F1*X1 - F2*X2(i))/M;
dP2dt = (F4(i) - F5(i))/C;

%Integracao numerica
L2(i+1) = integrate_euler(L2(i),dL2dt,stp);
X2(i+1) = integrate_euler(X2(i),dX2dt,stp);
P2(i+1) = integrate_euler(P2(i),dP2dt,stp);

write(PV,P2(i+1));

pv(i) = PV.Value;

figure(1);
plot(1:i,pv(1:i),'-b'),grid,TITLE('X2')
xlabel('Tempo');
ylabel('X2 (%)');
axis([0 i 0 100]);
drawnow;

end
%
disconnect(server_clp);

```