



UNIVERSIDADE DE UBERABA

PRÓ REITORIA DE PESQUISA, PÓS-GRADUAÇÃO E EXTENSÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA QUÍMICA
MESTRADO PROFISSIONAL

LUCIANO RANGEL PINHEIRO NETO

CRIAÇÃO DE UM AMBIENTE DE SIMULAÇÃO DE CONTROLE AVANÇADO DE
PROCESSOS EM CONTROLADORES LÓGICOS PROGRAMÁVEIS

UBERABA – MG

2019



LUCIANO RANGEL PINHEIRO NETO

**CRIAÇÃO DE UM AMBIENTE DE SIMULAÇÃO DE CONTROLE AVANÇADO DE
PROCESSOS EM CONTROLADORES LÓGICOS PROGRAMÁVEIS**

Dissertação apresentada como parte dos requisitos para obtenção do Título de Mestre em Engenharia Química do Programa de Pós-Graduação em Engenharia Química – Mestrado Profissional da Universidade de Uberaba (PPGEQ-MP/UNIUBE).

Orientador:

Profº. Dr. Edilberto Pereira Teixeira

UBERABA – MG

2019

Catálogo elaborado pelo Setor de Referência da Biblioteca Central UNIUBE

P655c Pinheiro Neto, Luciano Rangel.
Criação de um ambiente de simulação de controle avançado de processos em controladores lógicos programáveis / Luciano Rangel Pinheiro Neto. – Uberaba, 2019.
109 f. : il. color.

Dissertação (mestrado) – Universidade de Uberaba. Programa de Mestrado em Engenharia Química, concentração: Desenvolvimento de Processos Químicos Agroindustriais.
Orientador: Prof. Dr. Edilberto Pereira Teixeira.

1. Controladores programáveis. 2. Lógica difusa. 3. Controle de processos químicos. I. Teixeira, Edilberto Pereira. II. Universidade de Uberaba. Programa de Mestrado em Engenharia Química. III. Título.

CDD 629.895

LUCIANO RANGEL PINHEIRO NETO

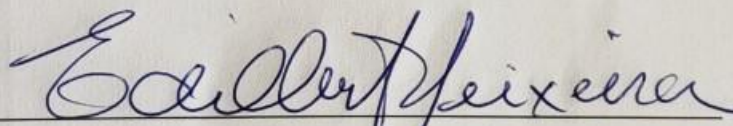
**CRIAÇÃO DE UM AMBIENTE DE SIMULAÇÃO DE CONTROLE AVANÇADO DE
PROCESSOS EM CONTROLADORES LÓGICOS PROGRAMÁVEIS**

Dissertação apresentada como parte dos requisitos para
obtenção do Título de Mestre em Engenharia Química
do Programa de Pós-Graduação em Engenharia
Química – Mestrado Profissional da Universidade de
Uberaba (PPGEQ-MP/UNIUBE).

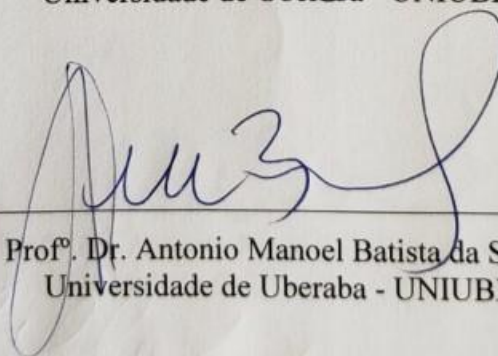
Área de Concentração: Desenvolvimento de Processos
Químicos Agroindustriais

Aprovado em: 11/10/2019

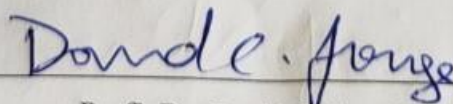
BANCA EXAMINADORA:



Prof^o. Dr. Edilberto Pereira Teixeira – Orientador
Universidade de Uberaba - UNIUBE



Prof^o. Dr. Antonio Manoel Batista da Silva
Universidade de Uberaba - UNIUBE



Prof^o. Dr. David Calhau Jorge
Universidade Federal do Triângulo Mineiro - UFTM

“Não importa o que aconteça, continue a nadar”.

(WALTERS, GRAHAM; PROCURANDO NEMO, 2003).

DEDICATÓRIA

Dedico essa dissertação...

Aos meus pais Roberto e Fátima, pelo amor, carinho e apoio em todos estes anos.

À minha namorada Mariana, pelo incentivo e compreensão pelos dias que tive que ficar estudando ao invés de ficar com ela.

À minha irmã Giovanna e meu cunhado Eusímio pelo constante apoio.

AGRADECIMENTOS

Acredito que nunca iremos realizar um trabalho sozinho, sempre terá contribuição, seja com ideias, conselhos ou alguma forma de incentivo. Para realizar esse trabalho contei com o apoio de muitas pessoas e tenho alegria de agradecer a todos:

À Deus por me oferecer tantas oportunidades de aprendizado.

Ao professor e orientador Edilberto, por confiar em mim e me dar a oportunidade de participar do projeto de iniciação científica propondo um tema muito interessante ao qual eu me identifiquei bastante e foi o início deste trabalho. Agradeço também pela orientação dedicada, disponibilidade e interesse com que me ajudou durante a elaboração do trabalho, além dos conhecimentos passados nas aulas ao longo do curso.

Ao professor José Delalibera Finzer pela qualidade com que conduz o Programa de Pós-Graduação em Engenharia Química.

À professora Elizabeth, por te me auxiliado ao longo do programa com suas contribuições.

À Secretária Rayane do Mestrado pelo auxílio técnico.

Ao meu colega de trabalho Fernando por todo aprendizado contínuo e ao Luiz Fabiano por apoiar meu desenvolvimento profissional.

Ao meu colega Evandro Assis pela parceria nesse período.

À Tereos por contribuir na minha formação profissional e por apoiar iniciativas de parceria universidade-indústria.

À Fundação de Amparo a Pesquisa de Minas Gerais – FAPEMIG

À CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.

À UNIUBE - Universidade de Uberaba pelo suporte dado ao desenvolvimento dessa dissertação.

Enfim, agradeço a todas as pessoas que contribuíram para o desenvolvimento deste trabalho.

RESUMO

Com a evolução da capacidade computacional dos modernos controladores lógicos programáveis, tornou-se possível a implementação de estratégias avançadas de controle que tradicionalmente são realizadas em sistemas digitais de controle distribuído. Desta forma, os controladores lógicos programáveis estão cada vez mais competitivos na área de controle avançado de processos. De forma geral, este trabalho apresenta a criação de um ambiente de simulação de controle avançado de processos por meio de uma rotina de controle no *software* RSLogix5000[®], com a utilização da ferramenta FuzzyDesigner[®] para a criação do controlador do tipo *Fuzzy*. Utilizou-se o *software* Matlab[®] como suporte para o cálculo de parâmetros do controlador e o *software* PlantTriage[®] para identificação da função de transferência do processo. Neste trabalho, foi simulado o controle de vazão de embebição utilizando-se um controlador *Fuzzy* cuja topologia tem-se como entradas o erro e a derivada do erro. Foi adotado o controlador SoftLogix5860[®], no entanto, os conceitos que serão utilizados são aplicáveis à maioria dos controladores programáveis comuns. A planta do processo foi simulada no próprio CLP e o sistema opera de forma integrada com um supervisor desenvolvido por meio do *software* FactoryTalk[®]. Com o término do trabalho foi obtido como resultado uma rotina em *ladder* no CLP capaz de simular sistemas industriais a partir de modelos matemáticos, gerando gráficos de forma intuitiva permitindo que estratégias de controle sejam previamente testadas nas indústrias pelo engenheiro de automação, em um ambiente de simulação, antes de serem implantadas no processo industrial. Além disso, foi feito um *how-to* de utilização de ferramentas com o foco na criação de rotinas de controle inteligente, em alternativa ao controle tradicional PID, e foi possível se concluir a simplicidade de uso da ferramenta FuzzyDesigner[®] mesmo se considerando a complexidade do assunto que é a lógica *Fuzzy*.

Palavras-chave: *Ladder*. Supervisor. Controlador *Fuzzy*. Lógica *Fuzzy*. Controle de Processos.

ABSTRACT

With the evolution of the computational capacity of modern programmable logic controllers, it has become possible to implement advanced control strategies that are traditionally performed in digital distributed control systems. In this way, programmable logic controllers are increasingly competitive in the area of advanced process control. In general, this work presents the creation of an advanced process control simulation environment by means of a control routine in RSLogix5000® software, using the FuzzyDesigner® tool to create the Fuzzy type controller. Matlab® software was used as support for calculating controller parameters and the PlantTriage® software for process transfer function identification. In this work, the control of imbibition flow was simulated using a Fuzzy controller whose topology has as inputs the error and the derivative of the error. The SoftLogix5860® controller has been adopted, however, the concepts that will be used are applicable to most common programmable controllers. The process plant was simulated in the CLP itself and the system operates in an integrated way with a supervisor developed through FactoryTalk® software. With the end of the work a ladder routine in the CLP was able to simulate industrial systems from mathematical models, generating graphs in an intuitive way allowing control strategies to be previously tested in the industries by the automation engineer, in an environment of simulation, before being implanted in the industrial process. In addition, a tool-based how-to was developed with a focus on creating intelligent control routines as an alternative to the traditional PID control, and it was possible to conclude the simplicity of using the FuzzyDesigner® tool even if considering the complexity of the which is Fuzzy logic.

Keywords : Ladder. Supervisory, Fuzzy controller. Fuzzy Logic. Process Control.

LISTA DE FIGURAS

Figura 1 - Pirâmide de automação.....	3
Figura 2 - Alta variabilidade com controle tradicional versus baixa variabilidade com controle avançado.....	4
Figura 3 - Combinação de diferentes técnicas de controle obtendo o desempenho ótimo.	5
Figura 4 - Estrutura de um sistema contendo controlador <i>Fuzzy</i>	14
Figura 5 - Etapas do processo em uma usina de cana-de-açúcar.....	15
Figura 6 - Modelos identificados pelo AMCT do PlantTriage.....	18
Figura 7 - Janela de tempo utilizada pelo AMCT para identificar o modelo de processo.....	18
Figura 8 - Modelo identificado pelo software PlantTriage.....	19
Figura 9 – Malha de controle com realimentação negativa.....	20
Figura 10 - Processo de conversão digital-analógica. Fonte: Nise, 2009.....	21
Figura 11 - Sinal amostrado por meio de um extrapolador de ordem zero.....	21
Figura 12 - Esquema de um amostrador.....	22
Figura 13 - Função de Transferência Discreta da planta.....	25
Figura 14- Comandos no Matlab para obtenção da função de transferência.....	25
Figura 15 – Resultado da conversão do Gprocesso(s) em cascata com o ZOH em $G(z)$ utilizando-se o Matlab®.....	26
Figura 16 - Diagrama de bloco no <i>software</i> Matlab® para comparação dos sistemas contínuo e discreto.....	26
Figura 17 - Resposta da simulação dos sistemas contínuo e discreto.....	27
Figura 18–Diagrama de bloco no <i>software</i> Matlab® para comparação dos sistemas contínuo e discreto com tempo morto.....	28
Figura 19 - Resposta da simulação dos sistemas contínuo e discreto com tempo morto.....	28
Figura 20 - Entrada e saída do sistema dinâmico discreto.....	29
Figura 21 - Rotina em linguagem <i>ladder</i> responsável em simular a planta.....	31
Figura 22 - Funcionamento do temporizador.....	33
Figura 23 - Indicadores do modo de operação do CLP.....	34
Figura 24 - Rotina <i>ladder</i> em execução.....	34
Figura 25 - Monitoração das tags.....	35
Figura 26 - Opção para monitorar uma <i>tag</i> por meio de gráfico.....	35
Figura 27 - Resposta da planta para $MV = 0.3$	36
Figura 28 - Estrutura de um controlador <i>Fuzzy</i> . Fonte: Haroldo, 1999.....	38
Figura 29- Representação dos conjuntos <i>Fuzzy</i> por meio de funções contínuas.....	40
Figura 30 - Função de pertinência do conjunto NEGATIVO.....	42
Figura 31 - Função de pertinência do conjunto ZERO.....	42
Figura 32 - Função de pertinência do conjunto POSITIVO.....	42
Figura 33 - Funções de pertinência para a variável ERRO.....	43
Figura 34 - Curva de um processo genérico. Fonte: HAROLDO, 1999.....	44
Figura 35 - Processo simplificado.....	45
Figura 36 - Análise da região I da curva genérica do processo.....	46
Figura 37 - Análise da região II da curva genérica do processo.....	47
Figura 38 - Diagrama de bloco de uma planta de vazão de embebição controlada por um controlador <i>Fuzzy</i> , controlador PID e em malha aberta.....	49
Figura 39 - Resposta da malha de controle com o controlador <i>Fuzzy</i>	50
Figura 40 - Bloco de instrução do controlador <i>Fuzzy</i> na aba <i>Add-On</i>	53
Figura 41 - Criação da TAG do controlador <i>Fuzzy</i>	53
Figura 42 - Rotina <i>ladder</i> contendo o controlador <i>Fuzzy</i>	54
Figura 43 - Janela de visualização de erros para a rotina completa.....	56
Figura 44 - Rotina <i>ladder</i> em execução – parte I.....	56
Figura 45 - Rotina <i>ladder</i> em execução – parte II.....	57
Figura 46 - Monitoração on-line de todas as TAGs.....	58
Figura 47 - Resposta da variável de processo.....	58
Figura 48 - Criando-se um projeto no <i>software</i> Factory Talk®.....	60
Figura 49 - Nomeando o projeto.....	61
Figura 50 - Tela principal do <i>software</i> Factory Talk®.....	61
Figura 51 - Criação de uma nova área para criação do supervisão.....	61
Figura 52 - Adicionando um HMI Server ao projeto.....	62
Figura 53 - Configurações do HMI Server.....	62
Figura 54 - Adicionando o dispositivo <i>RSLinx Enterprise</i>	63
Figura 55 - Configurações do novo Server.....	63

Figura 56 - Configurando a comunicação.	63
Figura 57 - Procedimento para criação da tela do supervísório.	64
Figura 58 - Símbolos convencionais de válvulas oferecidos pelo software Factory Talk®.	65
Figura 59 - Inserindo um campo de input.	65
Figura 60 - Procedimento para vincular o Numeric Input a uma Tag declarada na rotina <i>ladder</i>	66
Figura 61 - Seleção das variáveis declaradas na rotina <i>ladder</i>	66
Figura 62 - Tela do sistema supervísório do controle de vazão de embebição.	67
Figura 63 - Mensão honrosa no II EDEPA Uniube 2018.	71
Figura 64 - Criação e configuração do controlador SoftLogix 5860.	87
Figura 65 - Seleção do tipo de controlador a ser criado.	87
Figura 66 - Tela de configurações gerais do controlador virtual.	88
Figura 67 - Tela de configurações gerais do controlador virtual.	88
Figura 68 - Seleção dos modos de operação do SoftLogix 5860.	89
Figura 69 - Criação do projeto no RSLogix 5000®.	89
Figura 70 - Tela principal do <i>software</i> RSLogix 5000®.	90
Figura 71 - Tela de monitoração de <i>tags</i> e seus tipos.	90
Figura 72 - Tela para vincular uma <i>tag</i> ao temporizador.	91
Figura 73 - Janela de visualização de erros.	92
Figura 74 - <i>Download</i> do programa no controlador.	92
Figura 75 - Procedimento para enviar o programa ao controlador.	93
Figura 76 - Aviso de confirmação de <i>download</i> de programa para o CLP.	93
Figura 77 - Mudança do modo de operação por meio do <i>software</i> SoftLogix Chassis Monitor®.	94
Figura 78 - Tela inicial da <i>toolbox Fuzzy</i> do <i>software</i> Matlab®.	94
Figura 79 - Tela com o controlador com duas entradas.	95
Figura 80 - Configurando o tipo de função de pertinência.	95
Figura 81 - Configuração das funções de pertinência.	96
Figura 82 - Inserindo a base de regras.	96
Figura 83 - Visualização da Aplicação do critério MIN-MAX na aplicação das regras.	98
Figura 84 - Simulação em tempo real do FIS. Fonte: Autor.	98
Figura 85 - Superfície que ilustra a atuação das regras <i>fuzzy</i>	99
Figura 86 - Exportar o controlador <i>Fuzzy</i> para o <i>Workspace</i> do Matlab®.	99
Figura 87 - Tela inicial do FuzzyDesigner®.	100
Figura 88 - Barra de ferramentas para inserção dos blocos de montagem.	101
Figura 89 - Janela de parametrização dos blocos Input Port.	101
Figura 90 - Inserção dos blocos Input Port.	101
Figura 91 - Configurando o nome da variável e a interligação do bloco <i>Input Linguistic Variable</i>	102
Figura 92 - Configurando os valores de limite inferior e superior no bloco <i>Input Linguistic Variable</i>	102
Figura 93 - Inserção dos blocos <i>Input Linguistic Variable</i>	103
Figura 94 - Configuração da função de pertinência da variável linguística erro.	103
Figura 95 - Configuração da função de pertinência da variável linguística derivada do erro.	104
Figura 96 - Inserção do bloco <i>Output Linguistic Variable</i>	104
Figura 97 - Configuração da função de pertinência da variável linguística saída.	105
Figura 98 - Inserção do bloco Output Port.	105
Figura 99 - Inserção do bloco <i>Rule Block</i>	106
Figura 100 - Inserindo a base de regras no Rule Block.	106
Figura 101 - Simulação do sistema <i>Fuzzy</i> no FuzzyDesigner®.	107
Figura 102 - Configurando o <i>Add-On Instruction</i> a ser criado.	108
Figura 103 - Procedimento utilizado para importar o controlador Fuzzy.	108
Figura 104 - Selecionando o arquivo XML de importação.	109

LISTA DE ABREVIATURAS E SIGLAS

AMCT	-	<i>Automatic Model Capture Technology</i>
AOI	-	<i>Add-On Instructions</i>
BIT	-	<i>Binary Digit</i>
CLP	-	Controlador Lógico Programável
CLPM	-	<i>Control Loop Performance Monitoring</i>
DCS	-	<i>Distributed Control Systems</i>
EN	-	<i>Enable</i>
FFL	-	Carga <i>FIFO</i>
FFU	-	Descarga <i>FIFO</i>
FIFO	-	<i>First In First Out</i>
FIS	-	<i>Fuzzy Inference Systems</i>
IA	-	Inteligência Artificial
MV	-	Variável Manipulada
PID	-	Proporcional-Integral-Derivativo
PV	-	Variável de Processo
RUN	-	<i>Running</i>
SP	-	<i>Set-Point</i> ou Valor Desejado
TON	-	<i>Timer On-Delay</i>
TT	-	<i>Timer timing</i>
XML	-	<i>eXtensible Markup Language</i>
ZOH	-	<i>Zero-Order Hold</i>

SUMÁRIO

Capítulo 1 - Introdução	1
1.1 Evolução dos Controladores Lógicos Programáveis	1
1.2 Contexto Industrial	2
1.3 Justificativa	6
1.4 Delimitação do Tema	6
1.5 Objetivos	6
1.6 Forma de Desenvolvimento do Trabalho	7
1.7 Publicações	9
Capítulo 2 - Estado da Arte	10
2.1 Introdução aos Controladores <i>Fuzzy</i>	10
2.2 Topologia do Controlador <i>Fuzzy</i>	12
2.3 Conclusão	14
Capítulo 3 - Identificação da Função de Transferência e Simulação no CLP	15
3.1 Introdução Geral	15
3.2 Processo Industrial Escolhido	15
3.3 Identificação da Função de Transferência	17
3.4 Modelagem Matemática dos Conversores A/D e D/A	20
3.5 Determinação de $G(z)$	23
3.6 Simulação da Planta no CLP	30
3.6.1 <i>Softwares</i> utilizados para criação da rotina de simulação	30
3.6.2 Rotina de simulação	31
3.7 Conclusão	36
Capítulo 4 - Projeto do Controlador <i>Fuzzy</i>	37
4.1 Introdução Geral	37
4.2 Estrutura do Controlador <i>Fuzzy</i>	37
4.3 Conjuntos <i>Fuzzy</i>	38
4.4 Criação da Base de Regras	43
4.5 Simulação do Controlador <i>Fuzzy</i>	48
4.6 Conclusão	51
Capítulo 5 - Implementação do Controlador <i>Fuzzy</i> na Rotina do CLP	52
5.1 Introdução Geral	52
5.2 Parametrização do Controlador <i>Fuzzy</i> e Implementação na Rotina CLP	52
5.3 Conclusão	59
Capítulo 6 - Desenvolvimento do Sistema Supervisório	60
6.1 Introdução Geral	60
6.2 Criação do Sistema Supervisório	60
6.3 Conclusão	67
Capítulo 7 - Conclusões Gerais	68
Referências	69
Apêndice A – Artigos Publicados e Prêmio	71
Apêndice B – Passo-a-passo para criação da rotina <i>ladder</i> de simulação da planta ...	87
Apêndice C – Passo-a-passo para criação do Controlador <i>Fuzzy</i> utilizando-se o <i>software</i> Matlab	94
Apêndice D – Passo-a-passo para criação do Controlador <i>Fuzzy</i> utilizando-se o <i>software</i> FuzzyDesigner e posterior importação na rotina no RSLogix	100

Capítulo 1 - Introdução

1.1 Evolução dos Controladores Lógicos Programáveis

A criação dos microprocessadores na década de 1970 propiciou uma mudança drástica no desenvolvimento da humanidade. Eles estão presentes na grande maioria dos equipamentos que hoje utilizamos. Da mesma forma, nas indústrias, os microprocessadores são os grandes responsáveis pelo alto grau de automação que se observa nas mais diversas modalidades de processos.

Os controladores lógicos programáveis (CLPs) somente se tornaram viáveis com a criação dos microprocessadores. No início da década de 1970, a General Motors lançou um projeto para que se criasse um dispositivo microprocessado capaz de substituir os painéis de controle compostos de relés e de dispositivos lógicos. Várias empresas participaram dessa concorrência e, daquele projeto, nasceram os modernos controladores lógicos programáveis. Naquela época, os CLPs somente eram utilizados em lógica de intertravamento e sequenciamento de operações, operando somente com números inteiros. O controle de processos industriais era realizado pelos sistemas pneumáticos e pelos controladores proporcionais-integrais-derivativos (PID) construídos com eletrônica analógica. Com o avanço dos microprocessadores, o controle PID foi implementado de forma digital nos CLPs, permitindo uma grande redução de custos nos processos de produção.

Os CLPs foram então integrados aos sistemas de automação por meio de redes de dados viabilizando-se a sua operação por meio dos sistemas de *software* supervisórios instalados nas salas de operação das unidades industriais. Tais sistemas são denominados “sistemas abertos”, por permitirem a multiplicidade de fabricantes que integram a instrumentação aos sistemas supervisórios e aos CLPs.

Embora os CLPs tenham evoluído bastante em termos de capacidade computacional, ainda estão muito aquém dos modernos computadores e sistemas servidores. Sendo assim, na década de 1980, paralelamente à evolução dos sistemas abertos, houve uma grande evolução nos sistemas DCS (*Distributed Control Systems*), que se constituem em sistemas de automação onde o controle de processos tem, como base, avançados sistemas computacionais que, por sua grande capacidade computacional, permitem a implantação de sistemas avançados de controle. Os DCSs são considerados “sistemas fechados” porque, em geral, o *software* supervisório, o *hardware* e o *software* de controle são fornecidos por um único fabricante. São, em geral, muito

mais caros que os sistemas abertos e, por esse motivo, são preferidos somente por grandes indústrias de processo, como siderúrgicas e petroquímicas.

Paralelamente, à evolução dos sistemas de automação, houve um grande desenvolvimento das estratégias de controle, incluindo os sistemas multivariáveis e inteligentes. A criação de sistemas de *software* de simulação colaborou de forma definitiva para o desenvolvimento de sistemas avançados de controle. Dentre esses, destaca-se o sistema Matlab®, amplamente utilizados pelas universidades e pelos institutos de pesquisa. Infelizmente, esses sistemas de simulação não se difundiram nas indústrias, principalmente pelo seu alto custo e por exigir pessoal especializado e com conhecimentos avançados de programação e de controle de processos. Sendo assim, há uma lacuna de sistemas de simulação que propiciem um ambiente amigável aos técnicos e engenheiros de controle, nas indústrias.

1.2 Contexto Industrial

A competição global e a pressão dos acionistas fazem com que a indústria tenha que superar metas. Para permitir a continuidade da empresa no mercado que exige rápidas mudanças, as indústrias têm focado muito em melhoria contínua, em aumento da eficiência operacional, redução de custos, aumento da qualidade do seu produto final e investimento em projetos de otimização do seu processo industrial. As linhas de processo de uma planta devem estar continuamente em melhoria, mantendo a sustentabilidade dos negócios.

O sistema de controle de processo automatiza ambos, rotinas e condições anormais de operação, e auxilia a operação a manter o controle de seu processo. O desempenho do processo, e, portanto, os resultados dos negócios aumentam por meio de operações aplicadas e conhecimento do processo, via sistemas de controle de processos. Infelizmente, muitos estudos nas indústrias têm mostrado que grande parte das malhas de controle não operam em modo normal, sofrem oscilação e aumentam a variabilidade do processo, reduzindo a eficiência da planta.

O impacto direto no desempenho do processo é significativo, podendo citar: perda de produção, problemas de qualidade, consumo excessivo de energia, incidentes com segurança e meio ambiente, paradas não planejadas e alto custo de manutenção. Neste contexto, todas as camadas da pirâmide de automação devem ser tratadas.



Figura 1 - Pirâmide de automação.

Na Figura 1, a base da pirâmide de automação são os instrumentos que devem estar calibrados e com alta confiabilidade. Já na camada acima os PIDs devem estar bem sintonizados e com estratégias de controle bem definidas para cada aplicação industrial. Porém, nesta camada de controle regulatório o cenário que encontra-se na prática é ruim, com controladores abaixo do desempenho e estratégias de controle inapropriadas.

Segundo Fonseca (2004), uma malha de controle com desempenho ruim implica em perda de energia, qualidade e produção, já uma malha de controle com bom desempenho permite aumentar a qualidade e produtividade no processo.

A grande parte dos processos industriais utiliza o controle do tipo PID. A simplicidade aliada à eficiência encontrada neste tipo de controle é o principal motivo dele estar implementado em grande parte dos controladores industriais. Em Rubai et al. (2008) é citado que 96% dos controladores industriais são do tipo PID.

Entretanto, alguns sistemas são, muitas vezes, intratáveis pelos métodos convencionais de controle. Incluem-se, nessa classe, sistemas com:

- Características não lineares,
- Grandes atrasos de transporte,
- Equações variantes no tempo,
- Múltiplas entradas e múltiplas saídas,
- Equações difíceis de serem estimadas,
- Perturbações não modeladas,
- Equações conhecidas, porém muito complexas,

- Sistemas com amplas faixas de operação.

Em algumas aplicações mais complexas todos esses itens podem estar presentes de forma relevante em uma só planta industrial. Sendo assim, o fato de muitos sistemas possuírem características que inviabilizam ou dificultam ação de controles mais simples, o que torna necessário a utilização de alternativas como o controle *Fuzzy* para se operar o processo com maior eficiência e menor variabilidade conforme mostra a Figura 2.

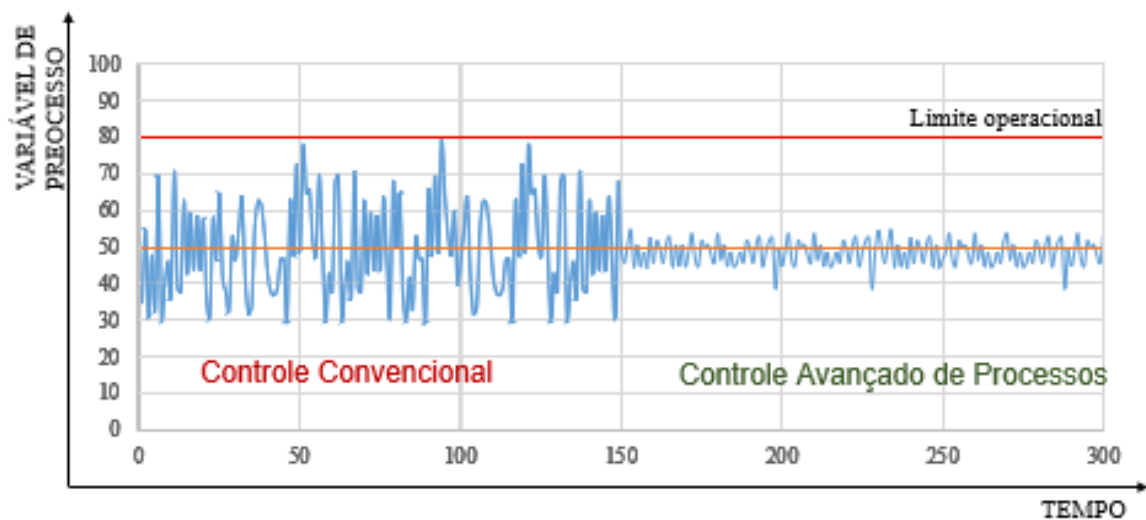


Figura 2 - Alta variabilidade com controle tradicional versus baixa variabilidade com controle avançado.

Conforme mostrado na Figura 3, as indústrias podem obter uma redução na variabilidade por meio da sintonia dos parâmetros PID das malhas de controle. A utilização de sistemas de otimização em tempo real que escrevem os Set-Points ótimos eleva o processo à um outro patamar com menor variação das variáveis de processo e atuação mais próximo ao limite operacional. Porém, a proposta do controle avançado deste trabalho é uma oportunidade de aumentar ainda mais o desempenho industrial, substituindo o controle tradicional que atua em uma variável e é baseado no erro, por um controle avançado multivariável que incorpora e antecipa distúrbios de processo. Com a combinação dessas técnicas, as variáveis de processo ficarão com variabilidade muito pequena e as malhas de controle poderão ser setadas para atuarem bem próximas ao limite operacional. É neste contexto que entra o controle avançado de processos utilizando a lógica *Fuzzy*.

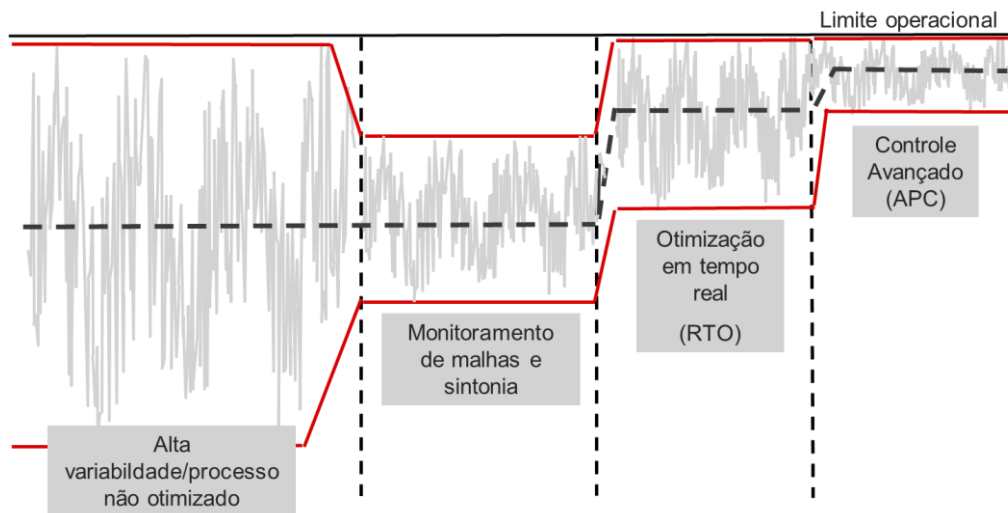


Figura 3 - Combinação de diferentes técnicas de controle obtendo o desempenho ótimo.

Criada por Zadeh em 1965, por meio da teoria dos conjuntos *Fuzzy*, evoluiu para a área de controles de processos por meio dos trabalhos de Mandani e Sugeno, nas décadas de 70 e 80, respectivamente. A lógica *Fuzzy* tem se tornado a opção mais prática e promissora para a implementação de sistemas especialistas. Segundo Haroldo (1999), atualmente, ela está sendo empregadas nos mais diferentes ramos, desde eletrodomésticos, como câmeras de vídeo e lavadoras, até centrais nucleares, indústrias químicas, automóveis etc. A criatividade humana e a sua capacidade de se adaptar às mais diversas situações foi provavelmente o que impulsionou a criação do controle do tipo *Fuzzy*. Os controladores *Fuzzy* são parametrizados de modo a “imitar” a ação de operadores de máquinas e de processos industriais, aplicando ações de controle para manter a estabilidade do processo assim como um ser humano faria caso estivesse operando.

Quando o homem faz parte de um sistema de controle, agindo como um controlador, em geral se obtém um efeito altamente estabilizante. Um exemplo que pode ser citado é o ato de andar de bicicleta. A bicicleta é um veículo inerentemente instável. Entretanto, quando o ser humano passa a fazer parte do sistema, desempenhando o papel de controlador, pode ser obtido um sistema bastante estável (Haroldo, 1999).

A falta de conhecimento de muitos sobre o assunto faz com que as ferramentas que auxiliam na criação de lógicas do tipo *Fuzzy* aplicadas a controle não sejam tão difundidas, o que torna ainda mais difícil para pessoas que não tiveram um primeiro contato com o assunto conseguirem dar os primeiros passos.

1.3 Justificativa

Considerando o desafio das indústrias de diminuir a variabilidade do processo e a necessidade de implantar controle avançado, e com a evolução da capacidade computacional, os CLPs estão cada vez mais competitivos tornando possível a implementação de estratégias avançadas de controle. E levando-se em consideração que há uma lacuna de sistemas de simulação de controle avançado que propiciem um ambiente amigável aos técnicos e engenheiros de controle nas indústrias. Neste contexto, a relevância deste trabalho será auxiliar as indústrias que pretendem implantar estratégias de controle avançado no controlador lógico programável porém que necessitem de um ambiente de simulação de baixo custo permitindo que a estratégia seja previamente testada na indústria antes de ser implantado no processo industrial, afim de atingir melhores resultados.

Como contribuição prática, alinhada à proposta do Mestrado Profissional em Engenharia Química, o material fruto do desenvolvimento deste trabalho será também direcionado a profissionais da área de automação que desejam implantar estratégias de controle avançado em CLPs. Contribuirá para a disseminação deste assunto, facilitando o processo de implantação e permitindo testar as estratégias de controle antes de colocar em operação aumentando a chance de sucesso.

1.4 Delimitação do Tema

O tema deste trabalho é a criação de um ambiente de simulação de controle avançado de processos. O que vai ser buscado com esta pesquisa é definido pela seguinte questão: **“Com a evolução dos controladores programáveis é possível criar um ambiente de simulação permitindo que as estratégias de controle avançado sejam previamente testadas no próprio CLP antes de serem implementadas?”**.

1.5 Objetivos

Dentro desse contexto, o objetivo geral do trabalho é:

- Criar um ambiente de simulação de controle avançado de processos no próprio CLP permitindo que estratégias de controle sejam previamente testadas nas indústrias antes de serem implantadas no processo industrial.

Para realizar esse objetivo, os seguintes objetivos específicos foram estabelecidos:

- Utilizar o *software* PlantTriage® para identificar a função de transferência do processo;
- Projetar e simular o controlador *Fuzzy* com auxílio do *software* Matlab®;
- Comparar o desempenho do controlador tradicional PID *versus* controlador *Fuzzy* com auxílio do *software* Matlab®;
- Criar o controlador *Fuzzy* com auxílio da ferramenta FuzzyDesigner®;
- Simular o processo no CLP inserindo a equação de diferenças na rotina *ladder*;
- Implementar o controlador *Fuzzy* na rotina do controlador lógico programável;
- Criar uma tela de sistema supervisor.

Os dados utilizados foram obtidos em uma usina sucroenergética situada no oeste paulista e pertencente ao grupo Tereos. A utilização dos dados foi realizada com consentimento dos departamentos ligados à operação. O autor deste trabalho é colaborador no grupo e possui acesso aos dados de sistema de supervisão e aos equipamentos industriais.

1.6 Forma de Desenvolvimento do Trabalho

O trabalho está dividido em 7 capítulos, incluindo este capítulo introdutório. De uma forma geral, nestes capítulos são demonstradas as etapas ou passos necessários para realizar a criação de um ambiente de simulação em uma rotina no CLP. A implementação é demonstrada por meio de imagens e descrições detalhadas dos procedimentos adotados, de modo que um usuário que já teve um contato prévio com rotinas de controle poderia por meio desta pesquisa se familiarizar melhor com o controle inteligente e como ele pode ser trabalhado nas ferramentas computacionais que serão abordadas.

O primeiro Capítulo, referente à introdução, apresenta os objetivos esperados, motivo da escolha do assunto e delimitação do tema.

O Capítulo 2 apresenta o estado da arte, ou seja, uma pesquisa da evolução da lógica *Fuzzy* e aonde vem sendo aplicado até o momento possibilitando o mapeamento de estudos que estão sendo, ou já foram realizados no Brasil e exterior com temáticas, ou linhas de pesquisa

iguais ou parecidas ao tema deste trabalho. O objetivo deste capítulo é detalhar o contexto em que se encontram as pesquisas relacionadas ao controle de processos utilizando controlador *Fuzzy*. Pretende-se com este capítulo mostrar que mesmo que a lógica *Fuzzy* tenha se difundido bastante, há uma escassez de um material passo-a-passo para implementar e incentivar a disseminação de tal tecnologia, ou seja, um material para proporcionar um primeiro contato com essa tecnologia. Há também uma escassez de sistemas de simulação de baixo custo. Isso justifica o desenvolvimento deste trabalho.

No Capítulo 3, inicialmente é demonstrada a modelagem para determinação da equação de diferença responsável por simular a função de transferência da vazão de embebição, para que a mesma alimente virtualmente os dados para atuação do controlador. Posteriormente, serão apresentados os procedimentos necessários para implementar a rotina em linguagem *ladder* por meio do *software* RSLogix5000[®] para simular a equação de diferença. Foi adotado o controlador SoftLogix 5860. Por fim será realizado o download da rotina no CLP e análise dos resultados.

Já no Capítulo 4 são definidas as variáveis de entrada e saída do controlador para cobrir o intervalo da respectiva variável por conjuntos *Fuzzy*, isto é, funções de pertinência. Também será descrita a teoria *Fuzzy* aplicada a controle de processos. Será apresentada a criação das regras de controle que é a parte mais importante na implementação do controlador. A criação da base de regras será por meio da análise da curva do sistema. O controlador *Fuzzy* será implementado com auxílio da *toolbox* FUZZY e simulado por meio do ambiente de simulação SIMULINK do *software* Matlab[®] para análise da resposta do processo diante do controlador projetado.

No Capítulo 5 é demonstrada a utilização da ferramenta FuzzyDesigner[®], criada pela Rockwell Automation. É um aplicativo que acompanha o *software* RSLogix5000[®] e que será responsável pela criação do controlador do tipo *Fuzzy*. Neste capítulo serão listados os passos necessários para a montagem e parametrização do controlador no FuzzyDesigner[®]. Após a criação do controlador serão demonstrados os procedimentos necessários para importar o controlador para o RSLogix5000[®] permitindo sua utilização nas mais diversas lógicas e rotinas de controle, neste caso, para o controle de vazão de embebição.

No Capítulo 6, é tratado o desenvolvimento de um sistema supervisório para monitorar o controle inteligente de uma planta simulada no próprio CLP. É demonstrada a utilização da ferramenta Factory Talk View Studio[®], criada pela Rockwell Automation.

Por fim, no Capítulo 7 são realizadas as conclusões gerais do trabalho.

1.7 Publicações

Os esforços durante o processo de pesquisa para esta dissertação possibilitaram as seguintes publicações:

PINHEIRO NETO, L. R.; TEIXEIRA, E. P.. **Criação de um ambiente de simulação de controle avançado de processos em uma usina sucroenergética**. Em: CONGRESSO BRASILEIRO DE INSTRUMENTAÇÃO, SISTEMAS E AUTOMAÇÃO, 2019, Campinas. Anais eletrônicos. Campinas, GALOÁ, 2019. Disponível em: <<https://proceedings.science/cobisa-2019/papers/criacao-de-um-ambiente-de-simulacao-de-controle-avancado-de-processos-em-uma-usina-sucroenergetica>> Acesso em: 01 julho. 2019.

PINHEIRO NETO, L. R.; TEIXEIRA, E. P.. **Utilização de ferramenta de control loop performance monitoring e Matlab para sintonia de malhas de controle**. Em: CONGRESSO BRASILEIRO DE INSTRUMENTAÇÃO, SISTEMAS E AUTOMAÇÃO, 2019, Campinas. Anais eletrônicos. Campinas, GALOÁ, 2019. Disponível em: <<https://proceedings.science/cobisa-2019/papers/-utilizacao-de-ferramenta-de-control-loop-performance-monitoring-e-matlab-para-sintonia-de-malhas-de-controle->> Acesso em: 1 jul. 2019.

PINHEIRO NETO, L. R. P.; TEIXEIRA, E. P.. **Utilização de ferramenta de Control Loop Performance Monitoring e Matlab para sintonia de malhas de controle em uma usina sucroenergética**. Em: II Encontro de Desenvolvimento de Processos Agroindustriais (EDEPA), 2018, Uberaba - MG. Disponível em: <<https://www.uniube.br/eventos/edepa/2018/downloadAnexo.php?pessoa=47738&sequencia=15>> Acesso em: 1 mar. 2019.

PINHEIRO NETO, L. R. P.; TEIXEIRA, E. P.. **Criação de um ambiente de simulação de controle avançado de processos em uma usina sucroenergética**. Em: II Encontro de Desenvolvimento de Processos Agroindustriais (EDEPA), 2018, Uberaba - MG. Disponível em: <<https://www.uniube.br/eventos/edepa/2018/downloadAnexo.php?pessoa=47738&sequencia=13>> Acesso em: 1 mar. 2019.

Os artigos mencionados acima estão no Apêndice A.

Capítulo 2 - Estado da Arte

2.1 Introdução aos Controladores *Fuzzy*

Nos últimos anos houve um grande crescimento de pesquisas de inteligência artificial motivadas por interesses econômicos e estratégicos. Essas pesquisas foram direcionadas para que as máquinas consigam tomar decisões inteligentes. Podemos citar várias vertentes de IA. A primeira delas, inteligência artificial baseada na lógica booleana, teve seu auge nas décadas de 70 e 80 e era aplicada aos sistemas especialistas. Posteriormente, surgiram as redes neurais que eram uma excelente opção para imitar o cérebro humano, sendo que no final da década de 80 obteve o seu reconhecimento. Destaca-se que as redes neurais apresentam a capacidade de aprendizagem conseguindo imitar de maneira rudimentar os seres humanos e animais. Esta é uma área em crescimento e que promete soluções fantásticas no futuro.

Os algoritmos genéticos foram criados por John Holland por meio da publicação do livro "*Adaption in Natural and Artificial Systems*" (Holland, 1975) baseado na teoria da computação evolucionária proposta por Rechenberg no seu trabalho "Estratégias de Evolução" (Rechenberg, 1975). Os algoritmos genéticos levam em consideração a evolução das espécies sendo suas técnicas inspiradas na biologia evolutiva como mutação, hereditariedade, mutação e crossing over. São utilizados na busca de soluções em problema de otimização e busca.

Embora não seja a última cronologicamente, a lógica *Fuzzy*, também conhecida por lógica nebulosa ou lógica difusa foi criada por meio da publicação do artigo "*Fuzzy sets, Information and Control*" que foi escrito por Lotfi Asker Zadeh, professor em Berkeley na Universidade da Califórnia, e que foi publicado em 1965.

A ideia de grau de pertinência, conceito que se tornou a base da teoria dos conjuntos *Fuzzy*, foi formulada em 1964, levando à publicação do seu artigo introdutório sobre os conjuntos nebulosos em 1965 e ao que poderia ser denominado nascimento da tecnologia baseada na lógica nebulosa (Yen, 1999).

A teoria dos conjuntos *Fuzzy* proposta por Zadeh (Zadeh, 1965) evoluiu para a área de controles de processos por meio da publicação do artigo "*An experiment in linguistic synthesis with a fuzzy logic controller*", escrito por Mamdani, E.H. & Assillan, S. em 1975, e publicação do artigo "*Industrial Applications of Fuzzy Control*", escrito por Michio Sugeno em 1985. Mamdani & Assilam foram os pioneiros de aplicação de lógica *Fuzzy* em controle de processos baseando-se na base teórica proposta por Zadeh (Zadeh, 1965). Mamdani (Mamdani, E.H. &

Assillan, S., 1975) desenvolveu um controlador que foi bem sucedido para o controle de uma máquina a vapor que antes não era possível com o controlador PID. A abordagem proposta por Sugeno (1985) baseia-se na utilização de um conjunto de regras *Fuzzy* para descrever um sistema não linear em termos de submodelos localmente lineares que são conectados por funções de pertinência, ou seja, neste artigo ele abordou a combinação de submodelos lineares para descrever um comportamento de um sistema não linear.

Motivadas pelos trabalhos de Mamdani & Assilam e pelo trabalho de Sugeno, várias aplicações apareceram na área de controle de processos, destacando-se, por exemplo, os controladores *Fuzzy* de plantas nucleares, refinarias, processos biológicos e químicos, trocador de calor, máquina diesel, tratamento de água e sistema de operação automática de trens (Jerry, 1995).

Em 1972 foi criado no Japão um grupo de pesquisa de sistemas *Fuzzy*. E em 1984 foi criada a Sociedade Internacional de Sistemas *Fuzzy* que era formada por pesquisadores de diversos países. Desde então, a lógica *Fuzzy* tem se tornado a opção mais prática e promissora para a implementação de sistemas especialistas. Atualmente, ela está sendo empregada nos mais diferentes ramos, desde eletrodomésticos, como câmeras de vídeo e lavadoras, até centrais nucleares, indústrias químicas, automóveis etc.

Devido ao desenvolvimento e às inúmeras possibilidades práticas dos sistemas *Fuzzy* e o grande sucesso comercial de suas aplicações, a lógica *Fuzzy* é considerada hoje uma técnica "*standard*" (do inglês, padrão) e tem uma ampla aceitação na área de controle de processos industriais.

Dentre as várias pesquisas na aplicação de lógica *Fuzzy* na área de controle automático de processos e sistemas de engenharia, podemos citar em 1976 a utilização de controlador nebuloso para o controle de fornos industriais levando-se em consideração a experiência dos operadores de máquinas e processo. Esta aplicação foi desenvolvida pelo Circle Cement e SIRA, na Dinamarca (Reznik, 1976). Em 1977, Didier Dubois aplicou a lógica *Fuzzy* para analisar as condições de tráfego. Em 1987, a lógica *Fuzzy* foi aplicada para o controle do metrô de Sendai, no Japão. Neste mesmo ano, a Yamaha desenvolveu um helicóptero não tripulado controlado por um controlador nebuloso. Em 1990, a Matsushita Electric Industrial criou uma máquina de lavar roupas utilizando controlador *Fuzzy*.

Hoje é possível encontrar, principalmente no Japão, toda a sorte de eletrodoméstico cujo sistema é baseado em controles *Fuzzy* (televisão, câmera fotográfica, panela para cozimento de arroz, vídeos, etc.) e existem atualmente várias empresas (Siemens, Daimler-Benz, Klockner-Moeller, SGS-Thomson, General Motors, Motorola, Hewlett-Packard, etc.)

que possuem laboratórios de pesquisa em lógica *Fuzzy* para desenvolvimento de seus produtos (Ortega, 2001).

Além disso, com a evolução da capacidade computacional dos modernos controladores lógicos programáveis, tornou-se possível a implementação de controladores *Fuzzy*. Podemos citar o artigo “*Fuzzy* Implementado em *Ladder* com Funções de Pertinência Descontínuas” (Oliveira & Braga & Almeida, 2010) que desenvolveu um controlador nebuloso para aplicações em sistemas industriais controlados por CLPs.

No artigo apresentado por Bhandari & Makwana (2016) aplicou-se o controle *Fuzzy* em uma caldeira a carvão de baixa pressão (21kgf/cm²) e, segundo os autores, a utilização de 625 regras *Fuzzy* e aplicação do controle inteligente contribuiu para a caldeira operar na faixa de 6,5% de teor de gás de oxigênio mesmo quando a demanda de vapor for variável.

Em um trabalho mais recente, uma publicação demonstrou a aplicabilidade do controle *Fuzzy* aplicado à combustão em caldeira a bagaço de cana-de-açúcar, como apresentado por (Mello, 2018) que destaca a aplicabilidade do controle *Fuzzy* na combustão em caldeira a bagaço de cana, com variabilidade 40% menos do que o controle PID.

Dessa forma houve uma gama enorme de sistemas nebulosos aplicados numa diversidade de aplicações, nas mais diversas áreas. Atualmente tem-se aplicações diversas relacionadas à implementação do controle *Fuzzy* em controladores lógicos programáveis, porém, há uma carência de um material para usuários que estão tendo o primeiro contato com a tecnologia. Este trabalho visa ser um manual referência para guiar os usuários na implementação desta estratégia em CLPs.

2.2 Topologia do Controlador *Fuzzy*

Da mesma forma que no controle convencional, os controladores *Fuzzy* podem ser classificados em dois grupos:

- Controle *Fuzzy* direto
- Controle *Fuzzy* realimentado

O controle *Fuzzy* direto normalmente é utilizado em sistemas com múltiplas entradas, em sistemas onde as perturbações não são modeladas ou em plantas que apresentam alto atraso de transporte. Esses sistemas são intratáveis pelos métodos convencionais, ficando, na maioria

das vezes, somente aos cuidados da operação manual de operadores especializados. O problema de atraso de transporte poderia ser resolvido, em grande parte dos casos, colocando-se os medidores e atuadores próximos ao processo. Entretanto, em geral, os custos são proibitivos. Considerando-se que os operadores sejam capazes de operar os sistemas nessas condições, cria-se uma base de regras baseada em suas experiências e se obtêm resultados muito satisfatórios com a aplicação de um controlador *Fuzzy* direto.

O controle *Fuzzy* realimentado é uma opção conveniente para sistemas com características não-lineares e com grandes faixas de atuação, porém com atrasos de transporte que se enquadrem dentro de uma faixa que venha a sugerir, por exemplo, o uso do controlador de Smith, também conhecido como preditor de Smith e que é uma estratégia de controle especial utilizada para compensação do tempo morto. Atrasos maiores que este somente poderá ser resolvido com o controle direto.

Há várias propostas de controladores fuzzy realimentados na literatura (Pedrycz, 1998) e (Haroldo, 1999), sendo que este assunto continua sendo objeto de muitas pesquisas. Soluções para sistemas que sejam considerados intratáveis têm sido propostas e implementadas com sucesso, nos mais diversos ramos de engenharia, utilizando-se os controladores *Fuzzy*. Nesta dissertação, apresenta-se uma das possíveis soluções, mostrando-se a viabilidade e a simplicidade de sua implementação. Também, nesse caso, torna-se necessário utilizar a experiência dos operadores. Quando isso não for possível, principalmente para plantas novas ou processos em desenvolvimento, o recurso da simulação é bastante valioso para auxiliar o processo de tentativa e erro, na sintonia do controlador. Será utilizado o *software* Matlab® como suporte para o cálculo de parâmetros do controlador.

A proposta apresentada utiliza os valores do erro de controle e a variação do erro de controle como entradas para o controlador *Fuzzy* como ilustra a Figura 4. Com o erro como entrada mede-se a distância da saída do processo em relação à referência desejada. A sua derivada é utilizada para a medida do grau de acentuação ou decréscimo deste, possibilitando assim uma melhor noção do que está ocorrendo no processo. Este é então aplicado como se fosse um controlador PID.

Como se observa, é possível se ter mais de uma entrada no controlador. Tal fato não é possível no controlador PID.

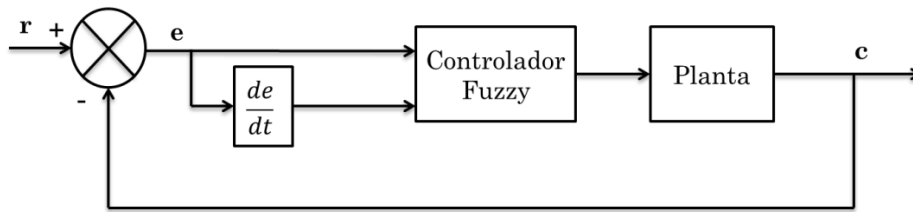


Figura 4 - Estrutura de um sistema contendo controlador *Fuzzy*.

2.3 Conclusão

O objetivo deste capítulo foi apresentar uma revisão geral da evolução da lógica *Fuzzy* e apresentar o estado da arte do tema destacando-se a falta de sistemas de simulação de baixa custo de controle avançado de processos e também da falta de materiais que descrevam o passo-a-passo para criar uma rotina inteligente em um CLP.

Neste capítulo também foi apresentada a topologia de controle adotada, onde serão utilizados os valores do erro de controle e a variação do erro de controle como entradas para o controlador *Fuzzy*.

Capítulo 3 - Identificação da Função de Transferência e Simulação no CLP

3.1 Introdução Geral

Para criar a rotina de controle inteligente em linguagem *ladder* é necessário o desenvolvimento do controlador e ter uma planta para testar o desempenho. Foi utilizado o algoritmo AMCT (*Automatic Model Capture Technology*) do *software* PlantTriage® para identificação do modelo matemático do processo.

Identificada a função de transferência no domínio de S, primeiramente, realizou-se a conversão para sua forma discreta, $G(z)$. Esta conversão foi realizada por meio de equacionamento matemático e posterior comprovação no *software* Matlab®. A função de transferência discreta foi obtida com um segurador de ordem zero com período de amostragem igual a 0,5s. Por fim, calculou-se a equação de diferença.

Para a simulação da malha de controle *Fuzzy* no controlador, a função de transferência do processo escolhido é inserida no formato de equação de diferença em uma rotina desenvolvida em linguagem *ladder*.

3.2 Processo Industrial Escolhido

Conforme mostrado na Figura 5, uma usina de cana-se-açúcar pode ser dividida em várias áreas: recepção/preparo/moagem, tratamento do caldo, fábrica de açúcar, fábrica de etanol, utilidades, disposição de efluentes e estocagem dos produtos.



Figura 5 - Etapas do processo em uma usina de cana-de-açúcar.

Uma importante operação unitária na usina é a extração do caldo da cana-de-açúcar que é realizado pelas moendas ou por difusores. Este processo consiste em separar o máximo possível de caldo da cana-de-açúcar. É importante que o controle desta área esteja otimizado para extrair o máximo possível de caldo o que significa aumento de produtividade para a unidade industrial. Na extração do caldo da cana-de-açúcar, um dos controles de processo é conhecido como controle de vazão de embebição que foi escolhido para simulação do controle inteligente. O controle de vazão de embebição tem como objetivo medir e controlar a vazão de água utilizada para a lavagem da cana-de-açúcar. Esta malha de controle é de extrema importância no processo de extração do caldo da cana-de-açúcar, pois quanto mais caldo for extraído da cana mais se tem para produção e menor a perda no bagaço.

O processo de embebição é uma etapa do processo ao qual aplica-se água ou caldo no bagaço que sai dos ternos das moendas com o objetivo de aumentar a extração de sacarose. Segundo Manela (2012, p. 64), a embebição pode ser simples quando é aplicada somente água como agente de embebição. A embebição pode ser composta, quando a água é aplicada somente antes do último terno de moenda, sendo o caldo extraído pelo último terno aplicado como embebição, no antepenúltimo terno assim sucessivamente até a saída do bagaço do primeiro terno de moenda.

Na unidade industrial da Tereos, onde identificou-se a função de transferência, existe um sistema de RTO (*Real Time Optimization*). Este sistema é baseado no modelo matemático da planta para verificar cenários próximos da operação atual e determinar qual a melhor condição de operação definindo um valor de *set-point* mais adequado para atingir o objetivo. Este sistema é capaz de prever o comportamento fluidodinâmico e termodinâmico do processo industrial e atuar para que o processo trabalhe no seu ponto ótimo, ou seja, baseado em equacionamento matemático, princípios de engenharia, dados e restrições de processo, este sistema determina a melhor condição de operação para o processo industrial no momento atual. Todas as operações unitárias da planta têm um modelo rigoroso que, com base em dados de entrada medidos comumente, é capaz de simular o comportamento e performance dos equipamentos.

O controle de vazão de embebição pode ter um *set-point* local para manter uma vazão fixa ou ter um sistema de otimização em tempo real (*Real-Time Optimization*) gerando *set-point* remoto. Uma das atuações do RTO é justamente definir o *set-point* remoto da malha de embebição que foi escolhida para simulação. Para isso, ele calcula um balanço de massa e energia online e atua enviando *set-point* para a malha de controle de embebição com o objetivo

da constância da embebição % fibra da usina e por consequência o aumento da extração da moenda. O RTO analisa a variação da fibra da cana processada, porcentagem em massa de sacarose e umidade do bagaço, e seus impactos no controle da vazão de embebição. O ajuste da quantidade ideal de água garante a melhor extração da moenda, sem excesso de água, que afeta umidade do bagaço que está sendo queimado na caldeira. A quantidade correta de embebição evita que seja gerado excesso de caldo, que na sequência consome vapor no seu tratamento e que também pode desequilibrar a operação do processo. Com o método tradicional de controle a operação não conseguia analisar em tempo real as variáveis de qualidade (fibra, porcentagem em massa de sacarose, umidade e extração), deixando o controle dependente apenas da rotação.

Definido o *set-point* ótimo para esta malha por meio de um sistema de *real-time-optimization*, o objetivo do controle avançado é diminuir a variabilidade da malha, ou seja, o RTO define o ponto ótimo de atuação da malha que é o *set-point* ótimo e o controlador *Fuzzy* reduzirá a variabilidade da variável de processo. A redução da variabilidade se reflete em aumento da extração do caldo-da-cana e consequente aumento de produção além de redução de consumo de vapor nas válvulas redutoras e sobra direta de bagaço para cogeração em períodos que a planta não está operando, como paradas programadas, paradas por chuva e entressafra.

3.3 Identificação da Função de Transferência

Para identificação da função de transferência, foi utilizado o algoritmo AMCT (*Automatic Model Capture Technology*) do *software* PlantTriage®. Trata-se de uma ferramenta de CLPM (*Control Loop Performance Monitoring*) desenvolvida pela empresa norte-americana ExpertTune. Este *software* opera monitorando continuamente e avaliando as malhas de controle e informa quando o desempenho começa a se deteriorar, disponibilizando indicadores de desempenho das malhas de controle. O AMCT funciona em *real-time* procurando e reportando modelos de processos por meio da avaliação automática de todos os testes de resposta que ocorrem na malha de controle, tais como: mudanças de *set-point*, rampas ou movimentos de válvula no modo manual. Ele filtra os dados e procura locais em que a variável de processo (PV) e a variável manipulada (MV) estejam estáveis, sofrem alteração e em seguida retornam novamente para o regime de estabilidade. Dessa forma, se o operador realizar uma mudança de *set-point* ou movimentar a válvula em manual, o AMCT avaliará os dados após o movimento da válvula e irá identificar e validar o modelo matemático do processo. Para validar o modelo,

os dados devem ser suaves, dentro do ruído normal de processo. A Figura 6 mostra os dez últimos modelos encontrados.

	Start	P	I	D	F	Quality	Gain	DT	Lag1	Lag2	Intg	Stability	RRT	Notes
<input type="checkbox"/>							%/%	sec	sec	sec			sec	
<input checked="" type="checkbox"/>	AMCT 03/10/2018 23:41	0,37	13	0	0	4	0,34	25,2	0	0	False	1,7	96	
<input checked="" type="checkbox"/>	AMCT 03/10/2018 18:58	0,39	50	0	0	2	0,37	27,3	0	0	False	6,1	100	
<input checked="" type="checkbox"/>	AMCT 03/10/2018 14:15	0,6	30	0	0	2	2,2	2	0	0	False	2,4	43	
<input checked="" type="checkbox"/>	AMCT 03/10/2018 13:14	0,13	3,6	0	0	3	1,4	4	0	0	False	1,3	16	
<input checked="" type="checkbox"/>	AMCT 03/10/2018 11:07	0,16	9,9	0	0	3	1,3	11	5,2	0	False	3	69	
<input checked="" type="checkbox"/>	AMCT 03/10/2018 10:44	0,043	39	0	0	2	3,5	70	0	0	False	43	200	
<input checked="" type="checkbox"/>	AMCT 02/10/2018 23:31	0,092	5,9	0	0	4	1,8	7	2,7	0	False	3,1	40	
<input checked="" type="checkbox"/>	AMCT 02/10/2018 21:51	0,43	6,3	0	0	1	0,58	6	3,8	0	False	0,7	43	
<input checked="" type="checkbox"/>	AMCT 02/10/2018 02:25	0,52	4,8	0	0	2	0,27	10	0	0	False	0,44	27	

Figura 6 - Modelos identificados pelo AMCT do PlantTriage.

A coluna “Quality” indica a qualidade do modelo que é um número entre 1 e 7. Sendo que 1 é uma qualidade excelente e 7 uma qualidade muito baixa. O AMCT só armazena modelos com qualidade de 4 (regular) ou melhor. A penúltima linha foi escolhida por ser um modelo de qualidade excelente. Ao clicar neste modelo, é mostrada a Figura 7, onde temos a janela de tempo onde o AMCT encontrou uma oportunidade de detecção do modelo.

Para a janela de tempo da Figura 7, o modelo do processo identificado está representado na Figura 8.

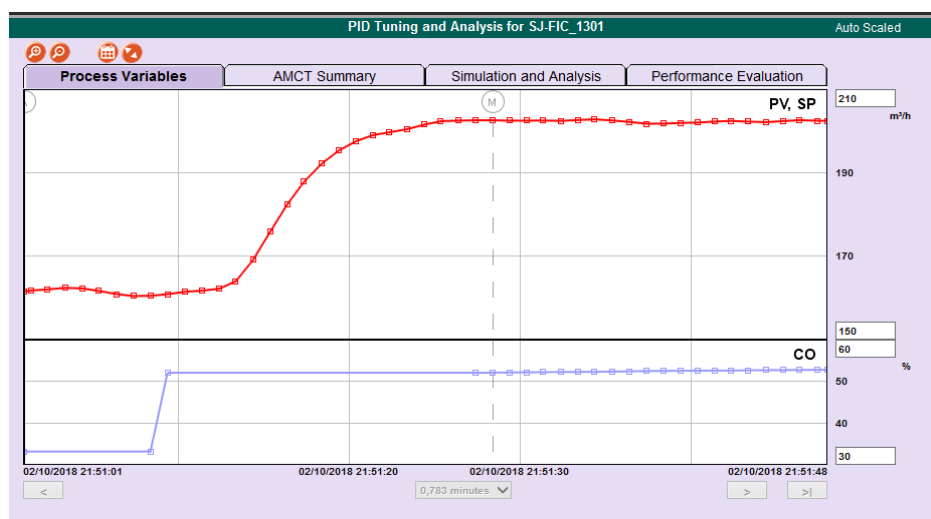


Figura 7 - Janela de tempo utilizada pelo AMCT para identificar o modelo de processo.

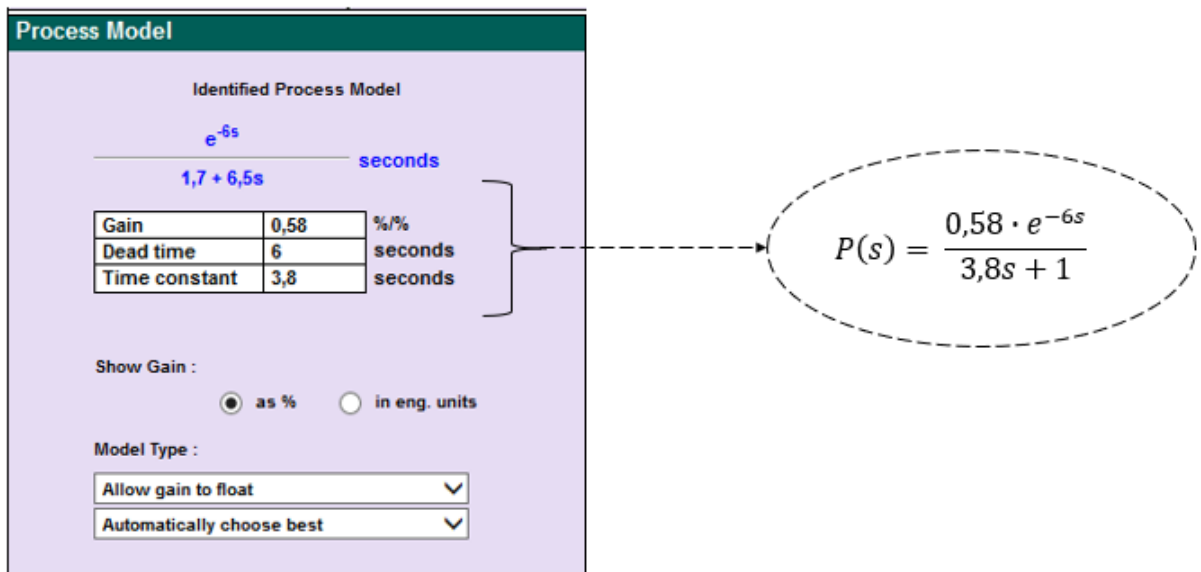


Figura 8 - Modelo identificado pelo software PlantTriage.

Sendo assim, o modelo do processo, $P(s)$ é conhecido.

$$P(s) = \frac{0,58 \cdot e^{-6s}}{3,8s + 1} \quad (1)$$

A função de transferência identificada é uma função de primeira ordem com tempo morto, onde o ganho $K = 0,58$, constante de tempo $\tau = 3,8s$ e tempo morto $\theta = 6s$.

Nos processos industriais, a presença de tempo morto é frequente. Esse problema surge quando a variável de saída de um determinado processo responde a uma variação no sinal de entrada, apresentando uma “inércia”, em outras palavras, depois de um determinado tempo desta variação ter realmente ocorrido (Neto, 2007). O tempo morto surge devido a transporte de fluidos em linhas longas, medidores com tempo de análise longo ou vários sistemas em série. Além disso, o controlador convencional não funciona bem em processos com tempo morto. Sendo assim, torna-se viável a utilização de técnicas inteligentes para melhorar o desempenho de sistemas com atraso de transporte, como por exemplo, a utilização do controlador *Fuzzy* em controladores lógicos programáveis.

3.4 Modelagem Matemática dos Conversores A/D e D/A

As vantagens de se substituir componentes analógicos por um computador digital são numerosas. Entre elas pode-se citar: redução de custo, flexibilidade na realização de alterações de projeto e imunidade a ruídos.

Definida a FT da planta de nível, procedeu-se na próxima seção a modelagem dos conversores A/D e D/A da malha de controle de nível.

A Figura 9 representa uma malha de controle de realimentação negativa.

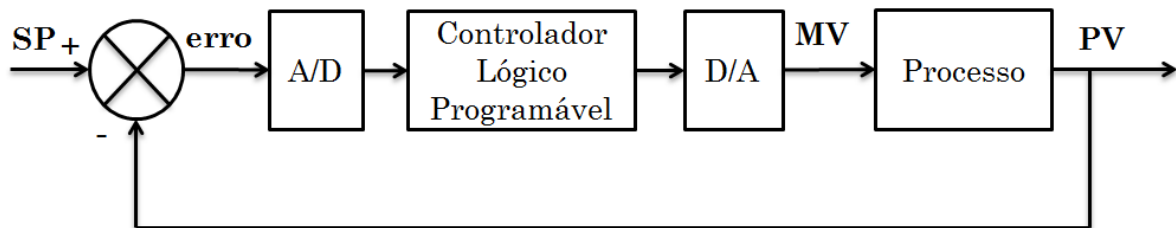


Figura 9 – Malha de controle com realimentação negativa.

A malha da Figura 9 contém sensores e atuadores que geralmente trabalham com sinais analógicos, como por exemplo, sinais elétricos de 4-20 mA. A malha também contém o controlador lógico programável que é um instrumento digital. Então é necessário ter dispositivos para converter sinais analógicos em sinais digitais e vice-versa de acordo com a necessidade. O dispositivo que converte sinais analógicos em sinais digitais é chamado de conversor analógico-digital (ou conversor A/D) e o dispositivo que converte sinais digitais em sinais analógicos é chamado de conversor digital-analógico (ou conversor D/A). Portanto, o controlador possui em sua entrada conversor A/D, e em sua saída, conversor D/A.

A conversão digital-analógica é simples e pode ser efetuada de forma instantânea. Portanto, apresenta função de transferência unitária. Conforme apresentado por Nise (2009) na Figura 10, as tensões elétricas ponderadas são somadas de modo a produzir a saída analógica.

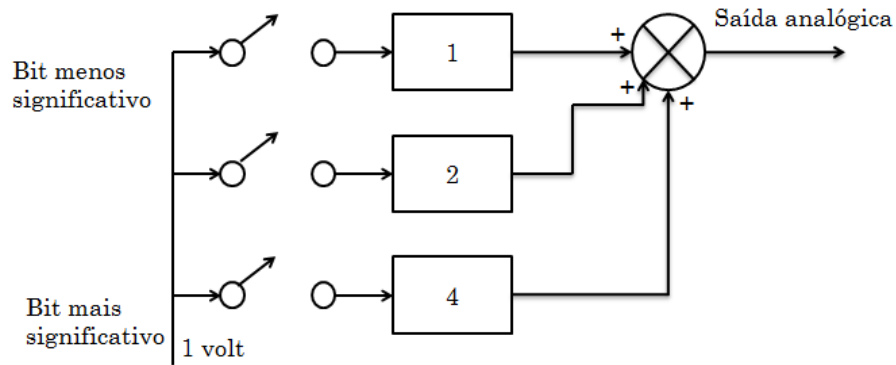


Figura 10 - Processo de conversão digital-analógica. Fonte: Nise, 2009.

O processo de conversão analógica-digital apresenta duas etapas e não é realizado de forma instantânea. Existe uma defasagem entre a tensão analógica de entrada e o sinal digital de saída. Nesta conversão o sinal analógico é inicialmente convertido em um sinal amostrado e depois transformado em um sinal digital. A taxa de amostragem deve ser igual a pelo menos duas vezes a banda passante do sinal, caso contrário haverá distorção. Essa sequência mínima de amostragem é chamada de frequência de amostragem de Nyquist (Nise, 2009).

A Figura 11-a apresenta um sinal analógico que pode ser amostrado em intervalos periódicos conforme mostrado na Figura 11-b. O sinal amostrado pode ser mantido constante durante o período de amostragem por meio de um dispositivo chamado de amostrador-extrapolador de ordem zero (ZOH) que produz uma aproximação do sinal analógico em forma de degraus conforme representado na Figura 11-c.

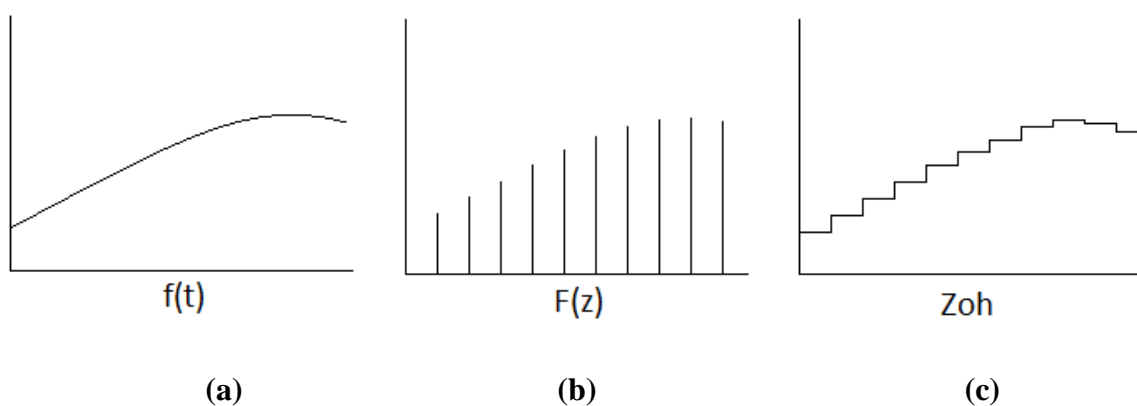


Figura 11 - Sinal amostrado por meio de um extrapolador de ordem zero.

O amostrador, conforme esquematizado na Figura 12, fecha uma chave periodicamente por um instante. Cada vez que a chave é fechada, um pulso de duração muito curta (teoricamente zero) e de magnitude igual ao sinal de entrada é produzido. O espaçamento

entre os pulsos é o período de amostragem (T). O extrapolador de ordem zero recebe este sinal e mantém na saída o último valor de sua entrada, produzindo um sinal em degraus. Um controlador discreto requer uma sequência de números na entrada. Um conversor A/D (analógico – digital) é um dispositivo que converte o sinal em degraus da saída do extrapolador de ordem zero para uma sequência de números (Dabney e Harman, 2003).

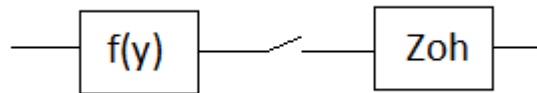


Figura 12 - Esquema de um amostrador.

Sendo que a função de transferência do segurador de ordem zero é igual à transformada de Laplace do pulso de comprimento T. Sendo assim, a resposta do conversor analógico-digital a um pulso pode ser expressa pela Equação 1:

$$p(t) = u(t) - u(t - T) \quad (1)$$

A transformada de Laplace correspondente ao pulso é dada pela Equação 2:

$$P(s) = \frac{1}{s} - \frac{e^{-Ts}}{s} = \frac{1 - e^{-Ts}}{s} \quad (2)$$

Portanto, a função de transferência do ZOH é dada pela Equação 3:

$$G_{z.o.h}(s) = \frac{1 - e^{-Ts}}{s} \quad (3)$$

Pela Equação 3, verifica-se que o extrapolador de ordem zero integra a sua entrada e retém seu valor durante o intervalo de amostragem. Como pode se observar o ZOH deve ser considerado na modelagem, pois a sua presença na malha de controle altera de maneira significativa a função de transferência equivalente da planta.

Sendo assim, o computador digital que fará o controle do processo será modelado pela associação em cascata do extrapolador de ordem zero e a função de transferência da planta. Como o CLP trabalha de maneira digital, será necessário aplicar a Transformada Z na FT resultante da associação em cascata do extrapolador e o processo.

3.5 Determinação de G(z)

Procedeu-se a modelagem matemática para determinar a função de transferência de dados amostrados, G(z), considerando que o período de amostragem, T, seja 0,5s.

A função de transferência do processo de vazão de embebição identificada anteriormente pelo PlantTriage® é dada pela Equação 4:

$$G_{processo}(s) = \frac{0,58 \cdot e^{-6s}}{3,8s+1} \quad (4)$$

Inicialmente será desprezado o tempo morto. Conforme visto no tópico anterior, o ZOH é modelado por meio da Equação 3. Realizou-se a conversão da função de transferência da planta em cascata com o ZOH e nomeou-se de G(s).

$$G(s) = \frac{1-e^{-Ts}}{s} \cdot \frac{0,58}{3,8s+1} \quad (5)$$

A função G(s) representa uma ocorrência comum nos sistemas de controle digital. Formulou-se uma solução geral para este tipo de problema deslocando o s no denominador do extrapolar de ordem zero para $G_{processo}(s)$.

$$G(s) = (1 - e^{-Ts}) \cdot \frac{G_{processo}(s)}{s} \quad (6)$$

A partir do qual,

$$G(z) = (1 - z^{-1}) Z \left\{ \frac{G_{processo}(s)}{s} \right\} = \frac{z-1}{z} Z \left\{ \frac{G_{processo}(s)}{s} \right\} \quad (7)$$

Posteriormente, foi necessário encontrar a resposta ao impulso (transformada de Laplace inversa) de $\frac{G_{processo}(s)}{s}$.

$$G_2(s) = \frac{0,58}{s(3,8s+1)} = 0,58 \frac{\frac{1}{3,8}}{s(s+\frac{1}{3,8})} \quad (8)$$

Expandindo-se em frações parciais, obtém-se:

$$G_2(s) = 0,58 \cdot \left[\frac{1}{s} - \frac{1}{\left(s + \frac{1}{3,8}\right)} \right] \quad (9)$$

$$g_2(t) = 0,58 \cdot [1 - e^{\frac{-t}{3,8}}]. \text{ Logo, } g_2(kt) = 0,58 \cdot [1 - e^{\frac{-kt}{3,8}}] \quad (10)$$

Aplicando-se a transformada Z, obtém-se

$$G_2(z) = 0,58 \cdot \left[\frac{z}{z-1} - \frac{z}{z - e^{\frac{-T}{3,8}}} \right] \quad (11)$$

A substituição de T pelo seu valor, 0,5s, resulta em:

$$G_2(z) = 0,58 \cdot \left[\frac{z}{z-1} - \frac{z}{z-0,876} \right] = 0,58 \cdot \left[\frac{0,124z}{(z-1)(z-0,876)} \right] \quad (12)$$

$$G_2(z) = \frac{0,0715z}{(z-1)(z-0,876)} \quad (13)$$

$$G(z) = \frac{z-1}{z} Z \left\{ \frac{G_{\text{processo}}(s)}{s} \right\} = \frac{z-1}{z} G_2(z) \quad (14)$$

$$G(z) = \frac{z-1}{z} \cdot \frac{0,0715z}{(z-1)(z-0,876)} \quad (15)$$

$$G(z) = \frac{0,0715}{(z-0,876)} \quad (16)$$

A função de transferência discreta, Equação 16, relaciona a transformada Z da saída da planta pela da entrada de um sistema dinâmico amostrado, considerando-se nulas as condições iniciais. A função de transferência discreta com a respectiva entrada e saída está representada na Figura 13.

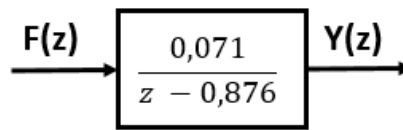


Figura 13 - Função de Transferência Discreta da planta.

A fim de comprovar o resultado encontrado fez-se uso da ferramenta Matlab[®] para resolução do problema. Foi utilizada a função *c2dm* (*continuous to discrete mode*) que apresenta a estrutura descrita a seguir.

$[\text{numeradorZ}, \text{denominadorZ}] = \text{c2dm}(\text{numeradorS}, \text{denominadorS}, T, \text{'zoh'})$

Onde:

- numeradorZ é o numerador do sistema discreto que se deseja obter
- denominadorZ é o denominador do sistema discreto que se deseja obter
- numeradorS é o numerador do sistema contínuo que se deseja converter para sistema discreto
- denominadorS é o denominador do sistema contínuo que se deseja converter para sistema discreto
- T é o tempo de amostragem, em segundos.
- 'zoh' indica que se deseja o equivalente da função contínua utilizando-se a operação *Zero-Order Hold*.

Os comandos que foram utilizados para obtenção da função de transferência discreta estão apresentados na Figura 14.

'MODELAGEM MATEMÁTICA'	%Exibe o título
T = 0.5;	%Define o período de amostragem.
numeradorS=0.58	%Define o numerador de G1(s)
denominadorS=[3.8 1]	%Define o denominador de G1(s)
'G1(s)'	%Exibe o título
G1=tf(numeradorS,denominadorS)	%Cria G1(s) e exibe
G=c2dm(G1,T,'zoh')	%G1(z) e exibe na tela

Figura 14- Comandos no Matlab para obtenção da função de transferência.

O resultado da execução do código fonte da Figura 14 está ilustrado na Figura 15.

```

G =

    0.07151
    -----
    z - 0.8767

Sample time: 0.5 seconds
Discrete-time transfer function.

```

Figura 15 – Resultado da conversão do Gprocesso(s) em cascata com o ZOH em $G(z)$ utilizando-se o Matlab®.

Utilizando-se a *toolbox* Simulink do *software* Matlab® foi possível comparar os sistemas contínuo e discreto em regime permanente. O diagrama de bloco encontra-se na Figura 16.

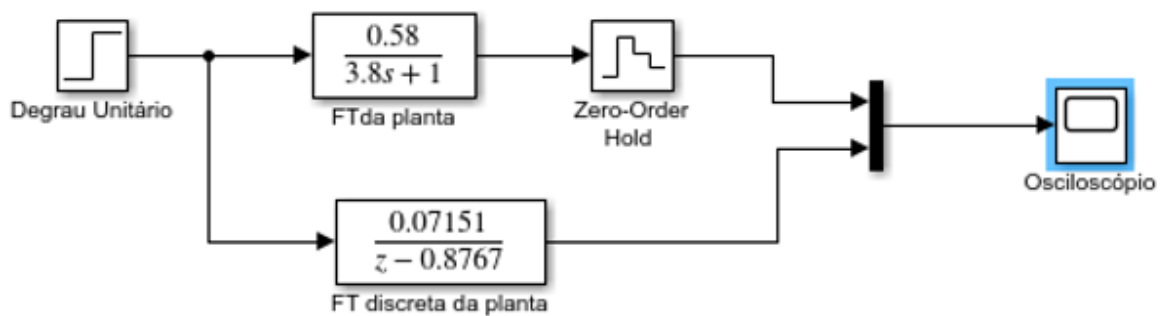


Figura 16 - Diagrama de bloco no *software* Matlab® para comparação dos sistemas contínuo e discreto.

Aplicou-se um degrau unitário no instante de zero segundo para verificar a resposta do sistema. O bloco *Scope* emula um osciloscópio plotando um gráfico do sinal de entrada. O gráfico gerado na simulação está representado na Figura 17.

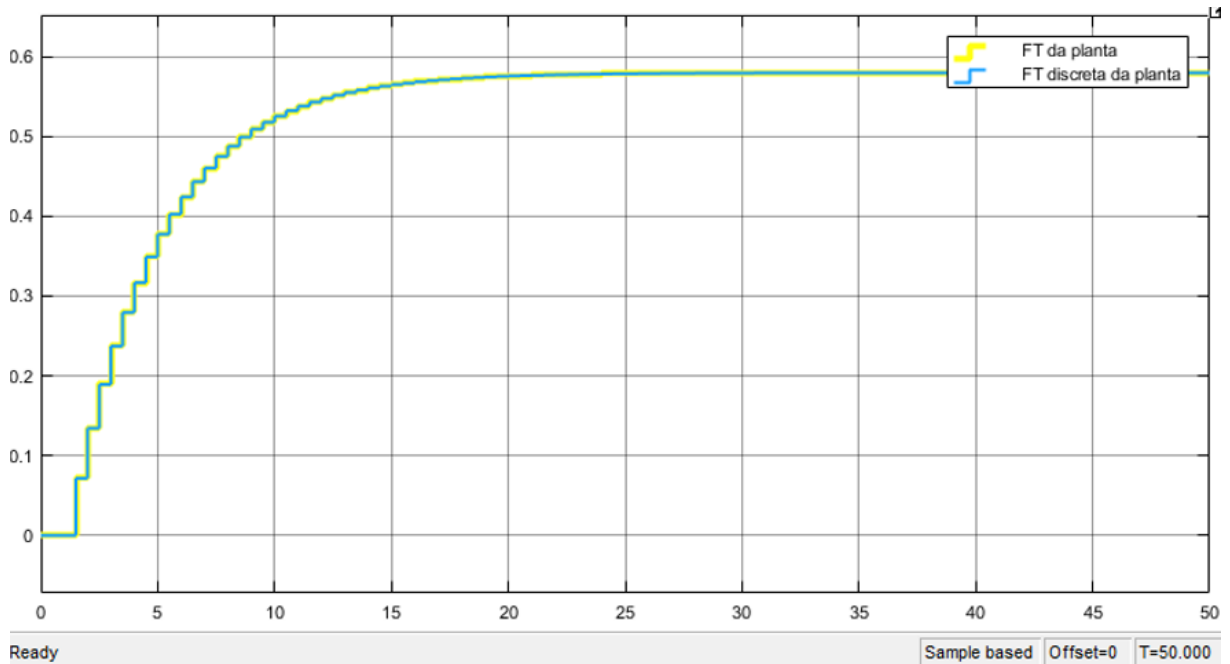


Figura 17 - Resposta da simulação dos sistemas contínuo e discreto.

Na Figura 17 as duas ondas se confundem, pois são iguais. Esta resposta comprova por meio de gráfico a teoria e a modelagem apresentada.

A simulação realizada no *software* Matlab[®] até o momento não levou em consideração o tempo morto do processo. Então, considerando que o período de amostragem T seja 0,5s e por definição $z = e^{st}$, portanto, $e^{-6s} = z^{-12}$ que representa o tempo morto do processo. Dessa forma, o novo $G(z)$ é:

$$G'(z) = \frac{0,07151 \cdot z^{-12}}{(z-0,8767)} \quad (17)$$

Novamente utilizou-se a *toolbox* Simulink do *software* Matlab[®] para comparar os sistemas contínuo e discreto em regime permanente considerando o tempo morto em ambos. O bloco *Transport Delay* foi utilizado para representar o tempo morto do sistema contínuo. No sistema discreto utilizou-se o bloco *Delay*. O diagrama de blocos está ilustrado na Figura 18.

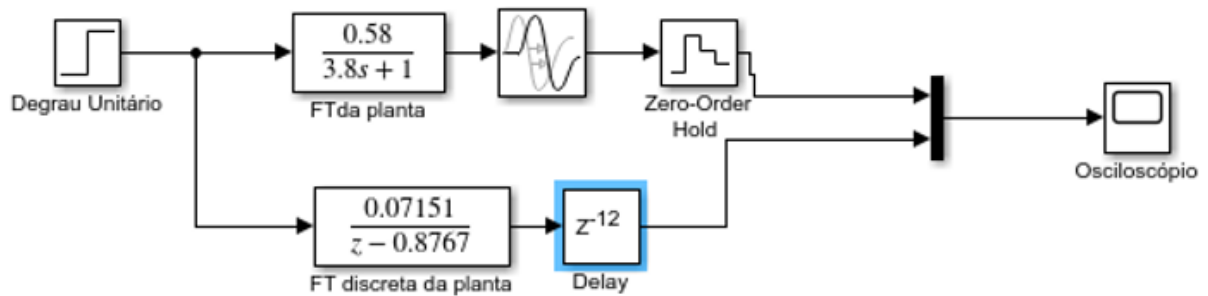


Figura 18–Diagrama de bloco no *software* Matlab® para comparação dos sistemas contínuo e discreto com tempo morto.

A resposta dos sistemas contínuo e discreto com tempo morto pode ser observada na Figura 19.

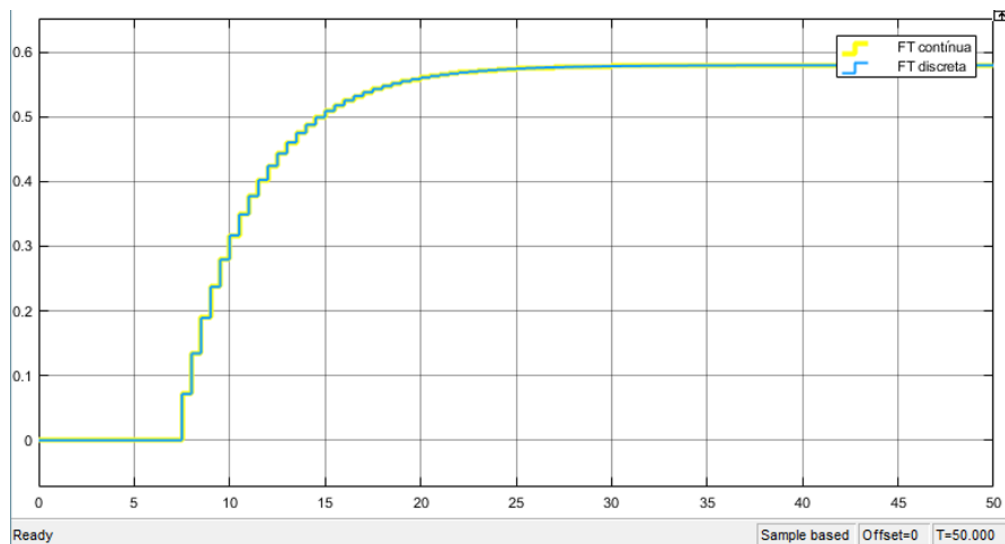


Figura 19 - Resposta da simulação dos sistemas contínuo e discreto com tempo morto.

Devido ao tempo morto, como era de se esperar, a variável de processo só começa a responder a aplicação do degrau unitário da entrada após 6 segundos. Observa-se na Figura 19 que novamente as duas ondas se confundem, pois são iguais, comprovando os resultados obtidos para o sistema com tempo morto. Com isso, pode-se proceder a simulação da planta no CLP. Esse procedimento será realizado na próxima seção, onde será implementada uma rotina para simular a planta no CLP. Antes disso, é necessário encontrar a equação de diferença da função de transferência do processo.

Sistemas dinâmicos são todos os sistemas que evoluem no tempo, qualquer que seja a sua natureza, isto é, sistemas físicos, biológicos, químicos, sociais, econômicos, etc. Esta

evolução pode ser descrita (modelada) por equações de diferenças, uma vez que esse tempo é muitas vezes medido em intervalos discretos. (Garanito, 2006).

Sendo a FT do processo dada pela Equação 18:

$$G(z) = \frac{0,07151}{(z-0,8767)} \quad (18)$$

Para obter a equação de diferença:

$$G(z) = \frac{Y(z)}{F(z)} = \frac{0,07151}{(z-0,8767)} = \frac{0,07151 \cdot z^{-1}}{1-0,8767 \cdot z^{-1}} \quad (19)$$

$$Y(z) \cdot [1 - 0,8767 \cdot z^{-1}] = F(z) \cdot [0,07151 \cdot z^{-1}] \quad (20)$$

$$Y(z) = 0,8767 \cdot z^{-1} \cdot Y(z) + 0,07151 \cdot z^{-1} \cdot F(z) \quad (21)$$

Na forma de equação de diferença:

$$y[n] = a \cdot y[n - 1] + b \cdot f[n - 1] \quad (22)$$

Portanto, substituindo as constantes que foram encontradas anteriormente, tem-se que:

$$a = 0,8767$$

$$b = 0,07151$$

A equação de diferença encontrada é dada pela Equação 23:

$$y[n] = 0,8767 \cdot y[n - 1] + 0,07151 \cdot f[n - 1] \quad (23)$$

A equação representa um sistema dinâmico discreto, onde $f[n]$ é a entrada e $y[n]$ é a saída conforme mostrado na Figura 20.

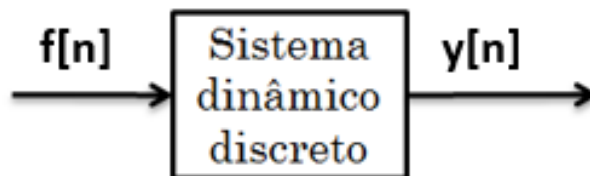


Figura 20 - Entrada e saída do sistema dinâmico discreto.

Para melhor identificação, a variável $y[n-1]$ será denominada de PV_AUX (variável de processo auxiliar – vazão de embebição anterior) que é o valor anterior ao valor atual da PV. A variável $y[n]$ será denominada de PV (variável de processo – vazão de embebição). A variável $f[n-1]$ correspondente à abertura anterior do elemento final de controle será denominada de

MV_AUX (variável manipulada auxiliar). Dessa forma, a equação encontrada para simular a planta no CLP está representada pela Equação 24.

$$PV = 0,8767 \cdot PV_AUX + 0,07151 \cdot MV_AUX \quad (24)$$

3.6 Simulação da Planta no CLP

O CLP é o responsável por analisar as entradas do sistema e a partir delas tomar as decisões. Utilizou-se a linguagem de programação *ladder* também chamada de Linguagem de Diagrama de Contatos, Diagrama de Relés ou Diagrama Escada, considerando-se ser a linguagem *ladder* a principal forma de programação aplicada nas indústrias para controladores lógicos programáveis e especificada pela IEC 61131-3 (1992). Além disso, optou-se por essa linguagem pelo fato de ser uma linguagem gráfica ao qual utiliza símbolos tais como contatos e bobinas, o que faz com que, independentemente do fabricante de CLP, as representações das instruções sejam semelhantes tornando a rotina genérica para os demais fabricantes.

3.6.1 Softwares utilizados para criação da rotina de simulação

Utilizou-se o *software* SoftLogix Chassis Monitor® para criar o controlador lógico programável. Esta ferramenta possibilita o computador industrial reconhecer e colocar em funcionamento qualquer lógica elaborada em outro programa em linguagem *ladder*, diagrama funcional de blocos, texto estruturado e SFC (Rockwell Automation, 2018). A tela principal permite configurar e monitorar os componentes do sistema. Nele é possível instalar dispositivos virtuais para representar o controlador e os cartões no sistema. Essa interface permite adicionar e configurar controladores, monitorar o controlador e o status do módulo.

O ambiente de desenvolvimento utilizado é o RSLogix5000® que é um *software* de programação da lógica *ladder* fabricado pela Rockwell Automation. O termo controlador Logix5000 refere-se a qualquer controlador que seja baseado no sistema operacional Logix, como: Controladores CompactLogix, ControlLogix, FlexLogix e SoftLogix (Rockwell Automation, 2018). Por meio do RSLogix 5000® pode-se editar o diagrama *ladder* referente à uma malha de controle, além de detectar erros de programação. A escolha deste *software* se dá também pelo fato deste ser compatível com programas criados por qualquer outro *software* da

Rockwell Automation, será necessário comunicar o CLP com o *software* FactoryTalk® para criação do controlador do tipo *fuzzy*.

3.6.2 Rotina de simulação

A criação da rotina que simule a planta foi estabelecida. Essa rotina é responsável por simular um processo real para posteriormente testar o controle por meio de técnicas inteligentes que levem em consideração a experiência do operador.

Detalhes do passo-a-passo para programação da rotina encontra-se no Apêndice B.

A Figura 21 representa uma imagem principal da rotina que realiza a simulação da planta. Nesta rotina existem duas linhas verticais denominadas de linhas de alimentação.

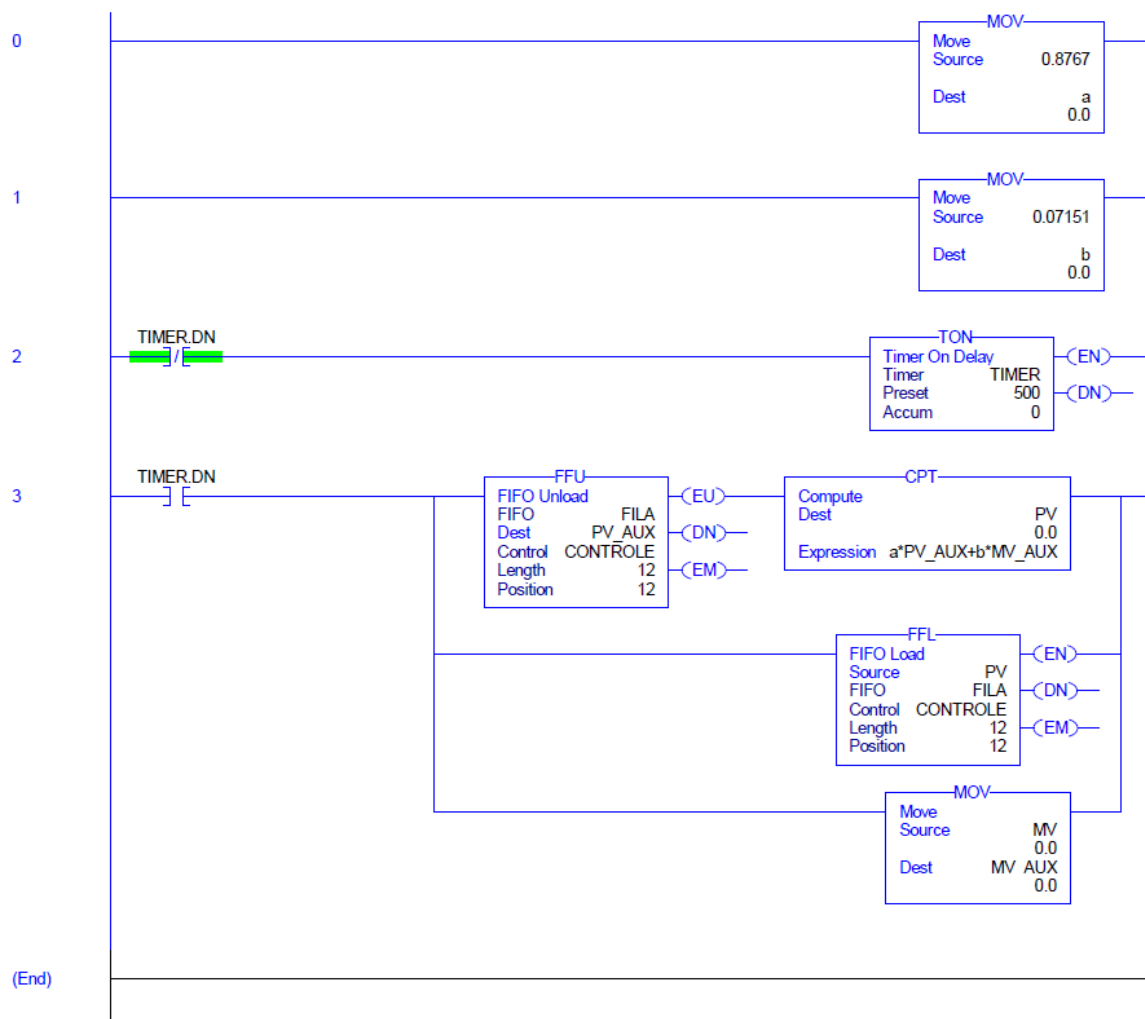


Figura 21 - Rotina em linguagem *ladder* responsável em simular a planta.

A instrução MOV na primeira e segunda linha da rotina da Figura 21 é utilizada para declarar as constantes da equação de diferença. Essa instrução é utilizada para transferir o conteúdo do endereço origem para o endereço destino. Neste caso está sendo transferidos 0.8767 e 0.07151 para as variáveis a e b. A instrução MOV compondo a terceira linha é a responsável em atualizar a variável MV_AUX que é responsável por armazenar o valor anterior do elemento final de controle.

A instrução TON (*Timer On-Delay*) é um temporizador e foi utilizado para executar o bloco CPT de 0,5 em 0,5 segundo atualizando os valores de simulação da planta. O bloco CPT é uma instrução matemática que ao ter a condição de entrada verdadeira realiza a operação matemática, executa e envia o resultado ao destino. Verifica-se na rotina que a PV é a variável correspondente a $y[n]$ decorrente da equação de diferença calculada. O valor anterior da variável de processo foi declarado como PV_AUX. O mesmo foi feito para o elemento final de controle, sendo declarado MV e MV_AUX para o valor atual e anterior, respectivamente.

Na lógica de controle implementada na Figura 21, ao energizar a entrada do temporizador, é iniciada a contagem em milissegundos e o bit EN (*enable*) é energizado para indicar que o temporizador está ativo. O valor acumulado indica o tempo corrido desde o início da contagem e o valor de “*preset*” indica o valor que o temporizador vai contar em milissegundos. Observa-se que o *preset* foi ajustado para 500 milissegundos, ou seja, 0,5 segundo. O bit DN (*done* – tempo atingido) permanece desenergizado até que o valor acumulado alcance o valor de *preset*. Quando a contagem alcança o valor de *preset* o bit DN é setado. O bit TIMER.DN permanece ativado até que o temporizador seja desativado (neste momento reseta o temporizador apagando o valor acumulado), ou seja, isso ocorre imediatamente já que o bit DN ao passar para nível lógico alto abre a linha de alimentação do temporizador levando o bit DN e EN novamente para nível lógico baixo e zera a contagem. O contato normalmente fechado na linha 2 da rotina apresenta continuidade lógica quando o bit DN apresenta nível lógico zero. Isso faz com que a entrada do temporizador seja energizada novamente e uma nova contagem de 0,5 segundo é iniciada. A Figura 22 ilustra o funcionamento em forma de gráficos.

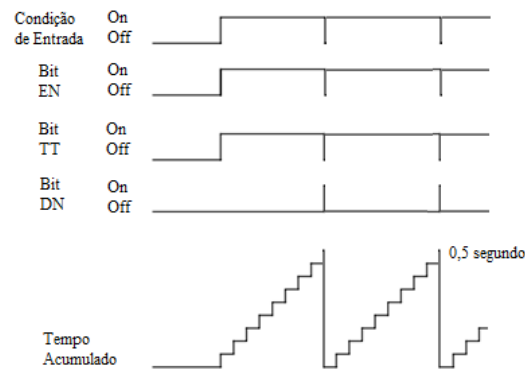


Figura 22 - Funcionamento do temporizador.

Dessa forma, de maneira reduzida, a instrução TON executará o bloco CPT toda vez que o temporizador atingir 0,5 segundo, recalculando periodicamente a equação de diferença do processo e atualizando a variável correspondente a vazão anterior. A Tabela I apresenta um resumo dos estados lógicos dos bits do temporizador para as condições de entrada.

Tabela I - Princípio de funcionamento do temporizador.

Condições de Entrada	Estado lógico
Entrada em nível lógico baixo	TT = 0
	DN = 0
	EN = 0
	ACC = 0
Entrada em nível lógico alto	TT = 1 (até atingir <i>preset</i>)
	EN = 1
	DN = 1 (após atingir <i>preset</i>)
	ACC = incrementado até atingir o <i>preset</i>

As instruções FFL e FFU foram utilizadas para simular o tempo morto do processo (z^{-12}) tendo como princípio o deslocamento assíncrono dos elementos na pilha. De 0,5 em 0,5 segundo o bit DN do temporizador TIMER vai para nível lógico alto alimentando a instrução FFL. A cada execução dessa instrução os elementos vão sendo transferidos da origem para o destino. O operando de lógica “*Source*” da Instrução FFL foi preenchido com a variável PV. Sendo assim, o FFL carrega o valor da variável de processo (PV) a cada execução da instrução armazenando na pilha do tipo FIFO. O operando de lógica “*Dest*” foi definido com a variável PV_AUX que é a variável onde será armazenado o valor descarregado da última posição da pilha do tipo FIFO. Como o FFU foi configurado com 12 posições, e sendo que ele é executado

a cada 0,5 segundo, então ele simulará o tempo morto de 6 segundos, pois o valor da variável de processo entrará na pilha e atualizará PV_AUX apenas decorridos 6 segundos.

Alterou-se o modelo de operação no RSLogix5000® para *Run Mode*, conforme ilustrado na Figura 23.

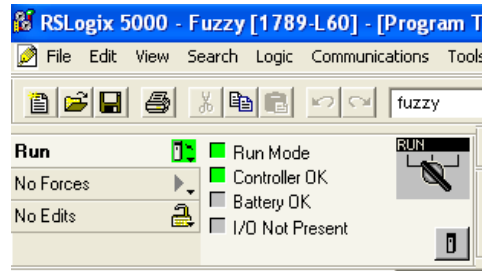


Figura 23 - Indicadores do modo de operação do CLP.

No modo *Run* a nova rotina será executada em modo de monitoração e todos os elementos de entrada e saída da rotina apresentará cores para representar o seu nível lógico conforme mostrado na Figura 24.

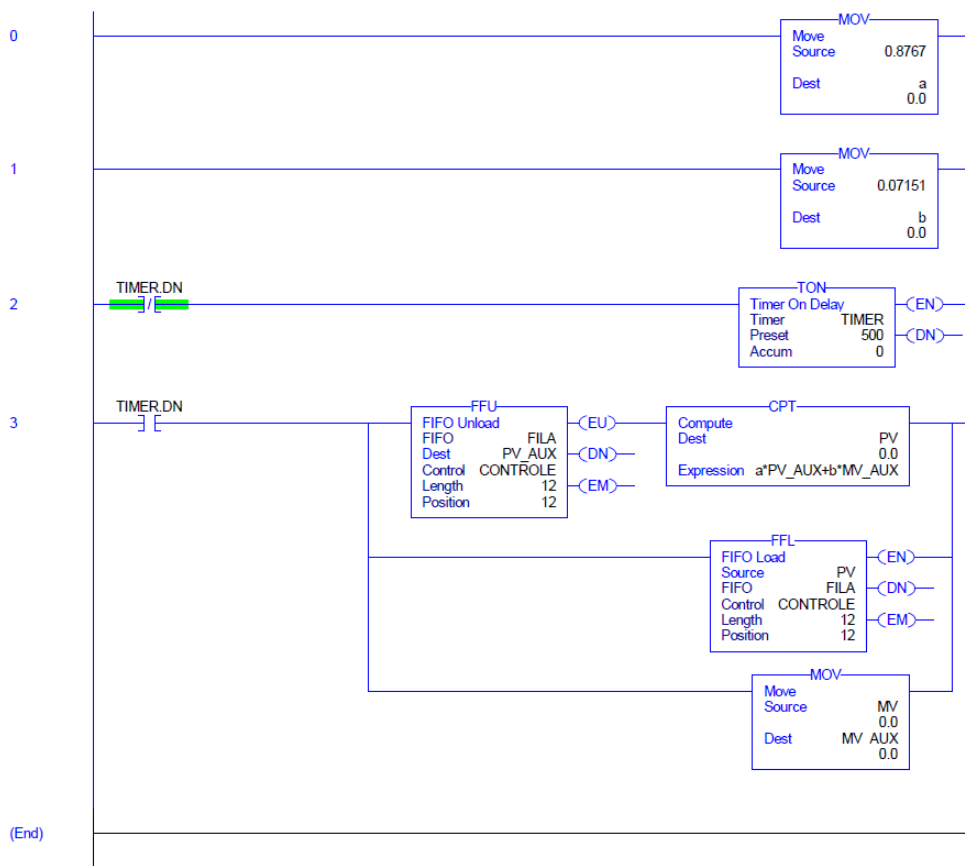


Figura 24 - Rotina ladder em execução.

O modo *Run* o RSLogix5000® permite monitorar as variáveis *on-line*. Ao clicar com o botão direito do mouse em *Controller Tags* é possível selecionar a opção *Monitor Tags* onde se podem monitorar todas as variáveis disponíveis para o uso do CLP conforme ilustrado na Figura 25.

Name	Value	Force Mask	Style	Data Type
a	0,8767		Float	REAL
b	0,07151		Float	REAL
CONTROLE	{...}	{...}		CONTROL
FILA	{...}	{...}	Float	REAL[12]
FILA[0]	0,104730345		Float	REAL
FILA[1]	0,104730345		Float	REAL
FILA[2]	0,104730345		Float	REAL
FILA[3]	0,104730345		Float	REAL
FILA[4]	0,104730345		Float	REAL
FILA[5]	0,104730345		Float	REAL
FILA[6]	0,104730345		Float	REAL
FILA[7]	0,104730345		Float	REAL
FILA[8]	0,104730345		Float	REAL
FILA[9]	0,104730345		Float	REAL
FILA[10]	0,1132701		Float	REAL
FILA[11]	0,1132701		Float	REAL
MV	0,3		Float	REAL
MV_AUX	0,3		Float	REAL
PV	0,1132701		Float	REAL
PV_AUX	0,104730345		Float	REAL
TIMER	{...}	{...}		TIMER

Figura 25 - Monitoração das tags.

Observa-se que a variável MV foi forçada a ter o valor de 0,3 para simular o processo com este sinal de entrada. A resposta do processo pode ser visualizada clicando-se com o botão direito do mouse na *tag* desejada e escolher a opção *Trend "PV"* conforme mostrado na Figura 26.

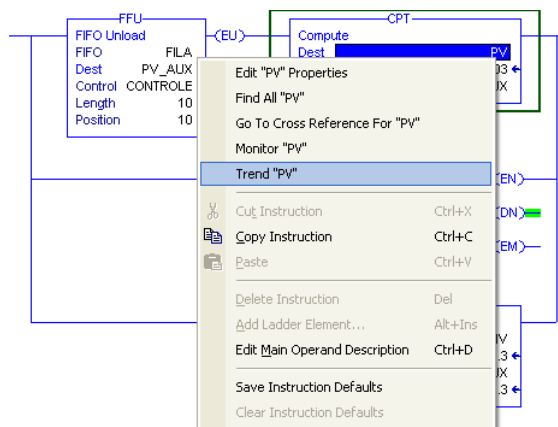


Figura 26 - Opção para monitorar uma *tag* por meio de gráfico.

A resposta da planta devido ao degrau de 0,3 na entrada pode ser visualizada na Figura 27. Verifica-se que há variações no valor da PV a cada 6 segundos, o que era de se esperar já que a planta adotada apresenta um tempo morto de 6 segundos.

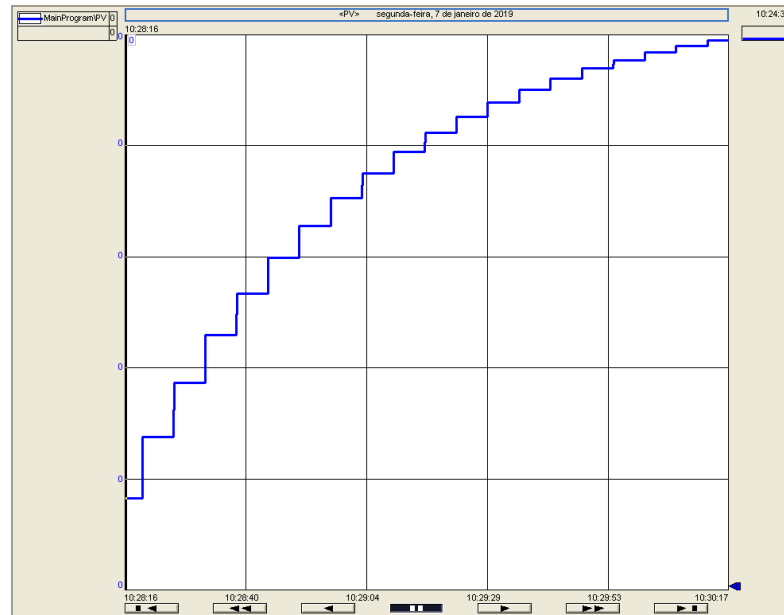


Figura 27 - Resposta da planta para $MV = 0.3$.

3.7 Conclusão

Neste capítulo foi descrito detalhadamente a criação da rotina de simulação da planta no CLP, destacando as etapas necessárias para criar o CLP virtual por meio do *software* SoftLogix Chassis Monitor[®], a implementação em linguagem *ladder* da rotina por meio do *software* RSLogix5000[®] e posteriormente, o download da rotina no CLP para testá-la.

De acordo com a execução da rotina no CLP e monitoração das *tags* a rotina implementada é capaz de simular a planta considerando o tempo morto adotado.

Capítulo 4 - Projeto do Controlador *Fuzzy*

4.1 Introdução Geral

Este capítulo trata do embasamento teórico necessário ao trabalho. Uma descrição dos componentes da estrutura do controlador *Fuzzy* é realizada. Procura-se somente apresentar os princípios básicos direcionados às aplicações industriais dos controladores *Fuzzy*, deixando-se de lado os caracteres formais da lógica *Fuzzy*. Entretanto, torna-se necessário o conhecimento da estrutura do controlador e também das noções de conjuntos *Fuzzy* e variáveis linguísticas.

É detalhada a criação das regras de controle que é a parte mais importante na implementação do controlador. A criação da base de regras se dá por meio da análise da curva do sistema.

Utilizou-se o *software* Matlab[®] como suporte para o cálculo de parâmetros do controlador e simulação para análise da resposta do processo diante do controlador projetado.

4.2 Estrutura do Controlador *Fuzzy*

Segundo Simões e Shaw (2007), um Controlador *Fuzzy* é composto dos seguintes blocos funcionais: Fuzzyficação, Base de conhecimento, Lógica de tomada de decisões e Defuzzyficação, conforme mostrado na Figura 28.

Desse modo, o controlador *Fuzzy* possui uma série de etapas para retornar um valor em sua saída. A primeira se constitui na fuzificação, passando em seguida por uma composição por meio de uma máquina de inferência e, por fim, terminando da defuzificação da variável. Além disso, existe a base de conhecimento do controlador que é composta pela base de regras e base de dados do processo.

O processo de fuzificação toma os valores numéricos (não *fuzzy*) das variáveis de entrada, faz um mapeamento para condicionar os valores de entrada a universos normalizados, e fuzifica-os, convertendo eles em conjuntos *Fuzzy* de modo que possam se tornar instâncias de variáveis linguísticas.

A base de regras é formada por um conjunto de regras linguísticas que definem a estratégia de controle do sistema. A base de dados constitui-se pelas definições dos conjuntos

fuzzy que descrevem os predicados de cada variável de entrada e saída com suas respectivas funções de pertinência, as quais podem ser discretas ou contínuas.

A máquina de inferência, obviamente, infere ações de controle empregando implicações *Fuzzy* e as regras de inferência da lógica *Fuzzy* além de simular tomada de decisões baseadas nos conceitos *Fuzzy*.

O processo de defuzificação torna a variável *Fuzzy* normalizada em uma variável não-*Fuzzy* na saída do controlador para que esta seja lida e entendida pelo processo a ser controlado.

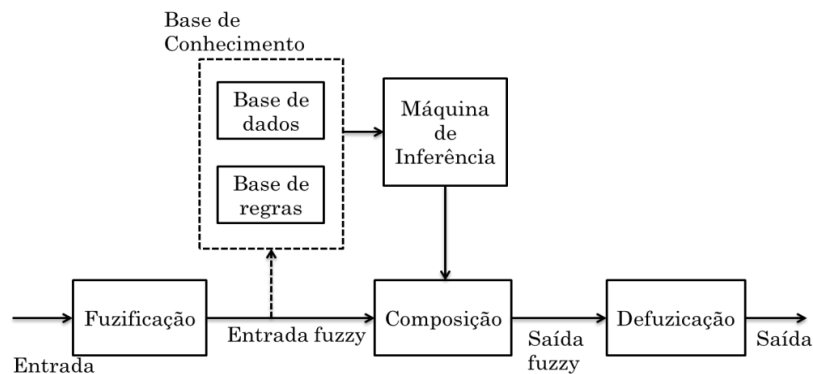


Figura 28 - Estrutura de um controlador *Fuzzy*. Fonte: Haroldo, 1999.

4.3 Conjuntos *Fuzzy*

Na teoria clássica de conjuntos, um elemento x_i pode pertencer ou não a um dado conjunto. Não há meios termos, isto é, não é possível que um elemento pertença um pouco a um conjunto e um pouco a outro conjunto. Entretanto, isso é perfeitamente possível na teoria dos conjuntos nebulosos. Nos sistemas *Fuzzy*, o intuito é que a decisão não seja apenas um sim ou um não, mas também tenha decisões abstratas do tipo próximo, em torno de ou muito alto.

Um conjunto *Fuzzy* é descrito por uma função que designa graus de pertinência entre zero e um aos seus membros. Um elemento que tenha grau de pertinência igual a zero não pertence ao conjunto. Grau de pertinência igual a um, indica que o elemento pertence totalmente ao conjunto. Graus de pertinência entre zero e um significam que o elemento pertence parcialmente ao conjunto.

Para exemplificação, considere o controle de vazão de embebição cuja vazão pode ser classificada como BAIXA, MÉDIA e ALTA. Classificando-se as vazões com pesos que variam entre 0 e 1 que definem o grau de pertinência dos elementos aos conjuntos, tem-se que uma

possível classificação de vazão poderia ser representada conforme na Tabela II que mostra o grau de pertinência do conjunto referente a vazão BAIXA, MÉDIA E ALTA.

Tabela II - Grau de pertinência dos elementos aos conjuntos.

Conjunto	Pertinência					
BAIXA	1	0.5	0.3	0	0	0
MÉDIA	0	0.3	1	0.4	0	0
ALTA	0	0	0.3	0.4	0.8	1
Vazão (m³/h)	0	100	200	300	400	500

O conjunto de valores de vazão (em m³/h), de um modo geral, é conhecido como universo de discurso.

Analisando-se o conjunto MÉDIA na Tabela II, 0 m³/h para a aplicação desejada é considerado uma vazão baixa, uma vez que seu grau de pertinência referente a vazão média foi zero. Já para um valor de 200 m³/h, tem-se grau de pertinência igual a um, indicando que está com uma vazão média. De modo análogo ao primeiro caso, 500 m³/h pode ser considerado vazão alta e retorna grau de pertinência nulo.

Costuma-se também representar os conjuntos *Fuzzy* da seguinte forma:

$$BAIXA = \left\{ \frac{1}{0}; \frac{0.5}{100}; \frac{0.3}{200}; \frac{0}{300}; \frac{0}{400}; \frac{0}{500} \right\}$$

$$MÉDIA = \left\{ \frac{0}{0}; \frac{0.3}{100}; \frac{1}{200}; \frac{0.4}{300}; \frac{0}{400}; \frac{0}{500} \right\}$$

$$ALTA = \left\{ \frac{0}{0}; \frac{0}{100}; \frac{0.3}{200}; \frac{0.4}{300}; \frac{0.8}{400}; \frac{1}{500} \right\}$$

A representação mostrada trata-se somente de uma representação, não tendo o traço horizontal o significado de fração ou divisão.

Os conjuntos *Fuzzy* podem também ser representados por funções contínuas e ser mostrados de forma gráfica conforme na Figura 29.

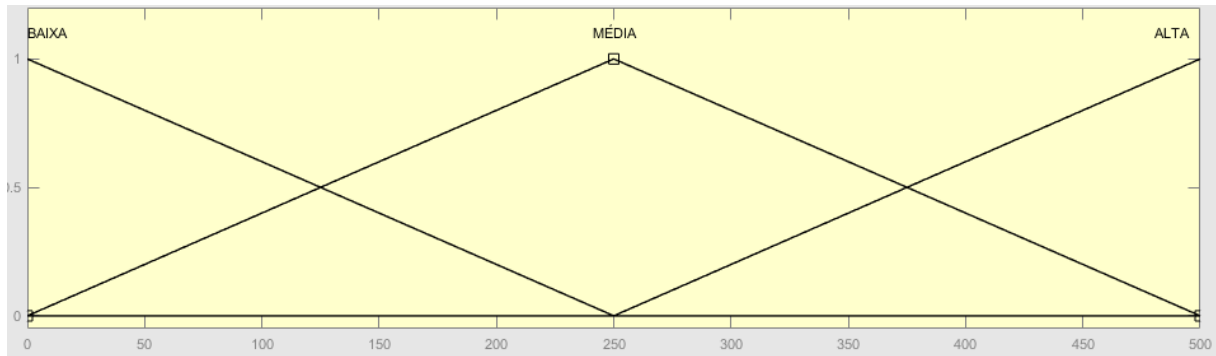


Figura 29- Representação dos conjuntos *Fuzzy* por meio de funções contínuas.

4.5 Variáveis Linguísticas

Variável linguística é uma variável em que os seus valores são conjuntos *Fuzzy*. As variáveis linguísticas são representadas por uma quádrupla $(x, T(x), G, U)$ (HAROLDO, 1999).

Onde:

- x é o nome da variável;
- $T(x)$ são os predicados que identificam linguisticamente diferentes regiões do sistema;
- G é a função de pertinência para cada conjunto *Fuzzy* designado por um predicado.
- U é o universo de discurso.

Foi adotada a topologia para o controlador *Fuzzy* que utiliza os valores do erro de controle e a variação do erro de controle como entradas. O erro em uma malha de controle pode ser calculado por: $\text{Erro} = SP - PV$. Considerando-se o erro como sendo uma variável linguística, tem-se que ele pode assumir valor de erro positivo quando vazão está abaixo do valor desejado, pode assumir valor de erro negativo quando a vazão está acima do valor desejado e pode assumir um erro zero quando a vazão de embebição é igual ao valor desejado. A mesma análise é realizada para a variação do erro que é calculada por $\frac{\text{Erro atual} - \text{Erro anterior}}{T}$ (em que T é o período de amostragem) e pode ter valores negativo, zero e positivo.

As duas entradas do controlador, erro e variação do erro de controle, e também a saída do controlador, são interpretados como variáveis linguísticas. Para a variável linguística “erro”

adotou-se três predicados *Fuzzy*: erro é negativo; erro é zero; erro é positivo. O conjunto de termos $T(\text{erro})$ pode ser:

$$T(x) = T(\text{erro}) = \{\text{NEGATIVO}, \text{ZERO}, \text{POSITIVO}\}$$

Cada termo de $T(x)$ é representado por um conjunto *Fuzzy* no universo U .

A variável erro é então representada pela quádrupla que se segue:

- $x = \text{erro}$;
- $T(x) = \{\text{NEGATIVO}, \text{ZERO}, \text{POSITIVO}\}$
- G é a função de pertinência para cada conjunto *Fuzzy* designado por um predicado.
- $U = -1$ a 1 (erro de -100% a 100%).

Os três conjuntos *Fuzzy* NEGATIVO, ZERO e POSITIVO foram criados para representar a variável linguística erro posto que o erro de controle pode ser negativo, zero ou positivo. Observa-se que se decidiu por classificar os conjuntos em NEGATIVO, ZERO e POSITIVO. Entretanto, se for conveniente, pode-se aumentar essa classificação, como por exemplo, muito negativo, negativo, zero, positivo e muito positivo. Denomina-se grau de cardinalidade ao número de classes nas quais se classificam as grandezas. Sendo assim, adotaram-se as grandezas como tendo cardinalidade igual a três.

Procedeu-se então a escolha da função de pertinência para os três conjuntos *Fuzzy*. Além da função triangular apresentada na Figura 29, tem-se vários outros tipos de funções de pertinência tais como: função trapezoidal, gaussiana, função sigmoide entre outras. Na prática, o tipo e a forma da função de pertinência deve refletir completamente a natureza do fenômeno que deseja modelar. A seguinte pergunta deve ser realizada: Qual a faixa de valores de erro que poderia qualificar um erro como negativo? A natureza desse conceito é tal que não se pode utilizar um simples número, pois a transição entre a noção de erro negativo e erro zero não é abrupta. Por exemplo, para representação de uma das entradas do controlador nebuloso, foi considerado que um erro igual a -1 apresenta grau de pertinência igual a 1 para o conjunto NEGATIVO. Vale ressaltar que um único valor numérico de erro pode cair em múltiplas categorias. Por exemplo, pode ser parcialmente zero e parcialmente positivo.

A função de pertinência que representará cada termo de $T(x)$ foi escolhida como sendo a função triangular. Isso porque erro = -100% apresentam grau de pertinência igual a 1 no conjunto NEGATIVO e o grau de pertinência deste conjunto decresce à medida que o erro aumenta. O erro igual a zero ocorre quando a variável de processo (vazão de embebição) é igual

ao valor desejado, neste caso tem-se que o erro é totalmente pertinente ao conjunto ZERO conforme evidenciado na Figura 31. O range utilizado para o erro foi de -1 a 1, ou seja, representando o intervalo de -100 a +100% de erro de controle. A função de pertinência possui amplitude unitária por estar trabalhando com graus de pertinência normalizados. A Figura 30, Figura 31 e Figura 32 representam a função de pertinência para cada conjunto *Fuzzy* designado por um predicado.

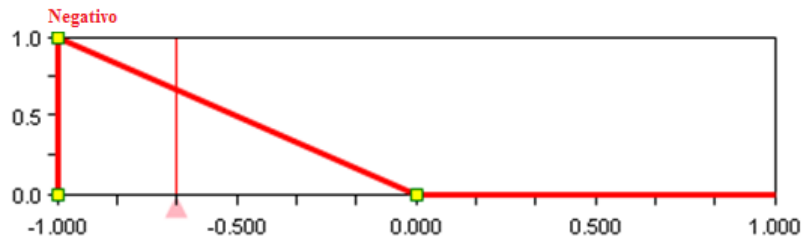


Figura 30 - Função de pertinência do conjunto NEGATIVO.

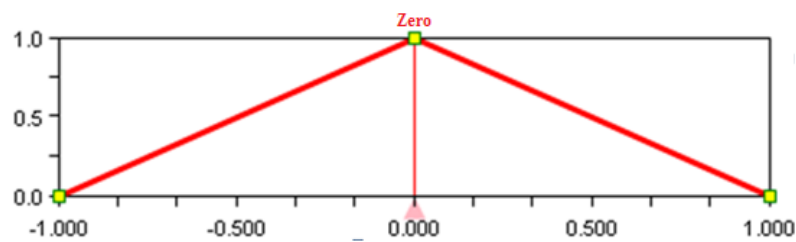


Figura 31 - Função de pertinência do conjunto ZERO.

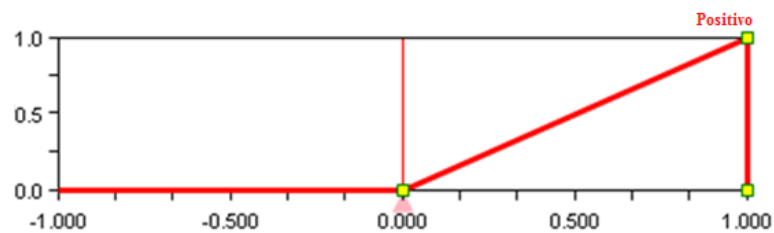


Figura 32 - Função de pertinência do conjunto POSITIVO.

As três funções de pertinência da variável linguística erro (E) estão representadas no mesmo gráfico na Figura 33. Observa-se que existe o “*crossover*” entre dois conjuntos *Fuzzy*, ou seja, o grau de pertinência no ponto de cruzamento entre dois conjuntos *Fuzzy* é 0,5.

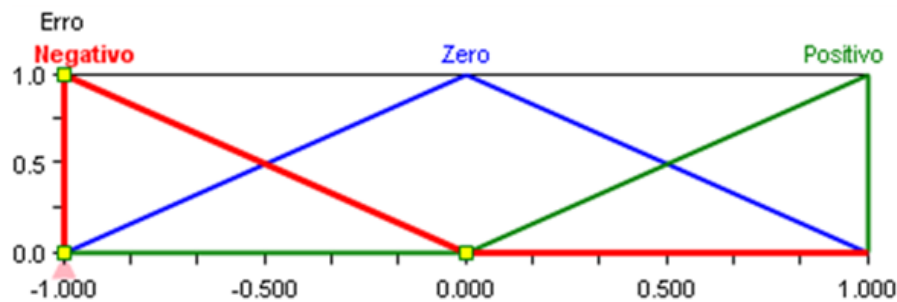


Figura 33 - Funções de pertinência para a variável ERRO.

A mesma quádrupla do erro foi utilizada para representar a outra entrada do controlador que é a variação do erro de controle (DE). Da mesma forma, esta quádrupla também foi utilizada para a variável linguística que representa a saída do controlador e que será chamada de SAÍDA.

4.4 Criação da Base de Regras

Uma regra *fuzzy* ou afirmação *fuzzy* relaciona variáveis *fuzzy*, cada uma delas associada a um dos seus predicados. Essas variáveis são conectadas por meio de operadores lógicos como: e, ou, então, outro. Um algoritmo de controle é constituído por um conjunto de regras *fuzzy* que são declarações condicionais ou incondicionais, envolvendo relações entre entradas e saídas (HAROLDO, 1999).

A tomada de decisão que um especialista usa na supervisão do sistema de controle pode ser expressa como um conjunto de regras *Fuzzy* do tipo se-então. Geralmente, um especialista que pode ser um engenheiro de controle, um operador ou pessoa da manutenção, conhece as ações a serem tomadas no momento de instabilidade. Temos muitas situações onde o processo está funcionando no modo manual ou um controlador PID mesmo bem sintonizado não fornece a resposta desejada. Uma alternativa é construir regras do tipo se-então para descrever o conhecimento dos especialistas e obter respostas satisfatórias.

O comportamento controle de vazão de embebição que foi adotado pode ser caracterizado por meio de regras *fuzzy*. Os operadores da planta conhecem, pela prática, regras tais como:

Se entrada 1 (E_1) for baixa E entrada 2 (E_2) for baixa ENTÃO a saída deve ser baixa (S_b).

Se a entrada 1 (E_1) for baixa E a entrada 2 (E_2) for média ENTÃO a saída deve ser baixa (S_b).

Se a entrada 1 (E_1) for grande E a entrada (E_2) for baixa ENTÃO a saída deve ser média (S_M).

Neste trabalho as regras *Fuzzy* levarão em consideração as duas variáveis linguísticas erro e a taxa de variação do erro que são as entradas do controlador nebuloso e variável linguística saída que é a saída do controlador que atuará no elemento final de controle. As regras são do tipo:

Se ERRO é POSITIVO e VARIAÇÃO DO ERRO é ZERO então SAÍDA é POSITIVA. Ou seja, esta regra determina que para estas condições de entrada deve-se enviar mais sinal para a válvula. Observa-se que as variáveis de entrada do controlador e a saída são relacionadas por meio do conectivo E e ENTÃO.

A proposta adotada para o controlador *Fuzzy* tem duas entradas e uma saída. Na seção anterior, definiram-se três predicados para cada entrada (negativo, zero e positivo) e a quantidade máxima de regras de controle a serem utilizadas está relacionada ao número de predicados das variáveis linguísticas. Então, o número máximo de regras será $3 \times 3 = 9$ regras.

Procedeu-se à criação das regras de controle que é a parte mais importante na implementação do controlador *fuzzy*. Essas regras podem vir a partir da experiência de um operador que trabalha no processo ou por meio da análise de uma curva do sistema como mostrada na Figura 34. Optou-se pela análise da curva da Figura 34 para criação da base de regras. Esta curva representa o pior caso possível de um controlador *fuzzy*, uma vez que existe um elevado “overshoot” (ultrapassagem do nível de referência) e um tempo de acomodação muito alto sendo uma curva de característica sub-amortecida. Este método de criação de regras *Fuzzy* é um método heurístico em que é realizada uma análise das características do processo. Na criação da base de regra devem ser tratadas todas as situações possíveis elaborando-se regras para todas as possibilidades.

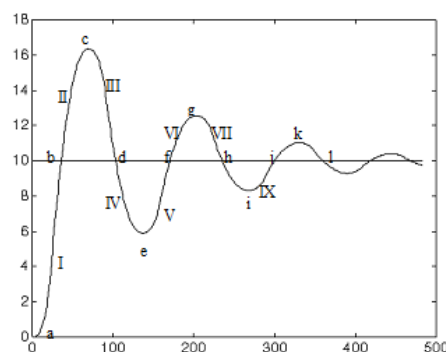


Figura 34 - Curva de um processo genérico. Fonte: HAROLDO, 1999.

Antes de analisar as regiões da curva da resposta foi necessário determinar o modo de controle do processo. Foi adotado para o processo de vazão de embebição que o elemento final de controle é do tipo normalmente fechado (NF). Deste modo é possível inferir qual é o modo de controle do processo. A Figura 35 representa de modo simplificado o processo a ser controlado.

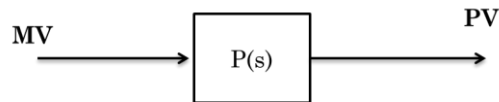


Figura 35 - Processo simplificado.

Na Figura 35, MV representa a Variável Manipulada, que é a válvula normalmente fechada, P(s) representa o processo e, por fim, PV é a Variável do Processo (vazão de embebição).

Caso a vazão de embebição aumente e ultrapasse o *Set-Point*, a válvula de controle deverá ser fechada para que a variável de processo alcance o valor desejado. Sendo a válvula do tipo NF, para fechar a válvula deve-se retirar sinal dela para que ocorra uma diminuição na PV até alcançar SP. Sendo o aumento de PV corrigido por uma diminuição no elemento final de controle, tem-se que esse tipo de comportamento observado para esta planta é característico do sistema de controle reverso, o qual possui o erro dado por $SP - PV$ (em que SP é o “*set-point*” ou valor de referência). Isto é de suma importância na análise do erro do sistema, uma vez que é a partir desse erro que se construirá a base de regras a ser embarcada no controlador *Fuzzy*.

Com o modo de controle definido procedeu-se a análise da curva da Figura 34 para criação da base de regras. Sabendo que $Erro = SP - PV$ e que a derivada do erro é $\frac{E_{atual} - E_{anterior}}{T}$ (em que T é o período de amostragem) pode-se analisar a curva para as suas diversas regiões e analisar o seu efeito sobre o processo. As variáveis de entrada a serem analisadas são o erro (E) a taxa de variação do erro (DE). Já a saída (definida como variável linguística SAÍDA) é o incremento ou decremento a ser dado na variável de ação que vai atuar na planta.

Segue a análise das regiões da curva da Figura 34.

- **Região I** – tomando dois pontos dentro desta região observa-se que o erro (erro atual – erro anterior) é positivo, indicando um valor de PV abaixo do SP, e sua derivada é

negativa. Porém, após uma análise mais rígida, percebe-se que a regra para este caso é manter o sinal enviado para a MV. Isto porque a PV está encaminhando para o SP, não havendo necessidade de modificar a abertura da válvula. O detalhe da análise da região I está representado na Figura 36.

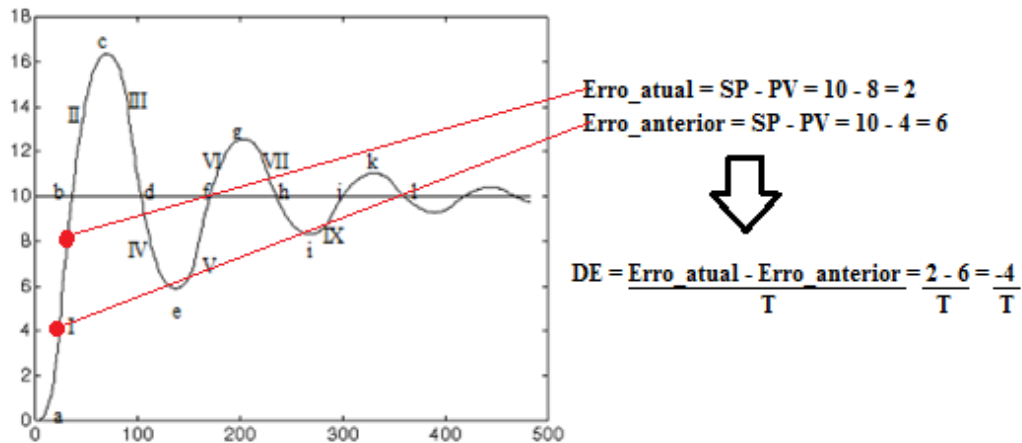


Figura 36 - Análise da região I da curva genérica do processo.

Pela análise da região I realizada na Figura 36, pode-se observar que as regiões que apresentam valor menor que o valor desejado ou valor de referência apresentam erro positivo. Se o valor for maior que o valor de referência o erro é negativo. A taxa de variação do erro é negativa quando o nível é crescente, e positiva quando é decrescente.

- Região II** – de modo análogo, tomando dois pontos dentro desta região na Figura 36 observa-se que o erro é negativo, indicando um valor acima do SP, e sua derivada é negativa. Deste modo, é necessário fechar a válvula para que o sistema tenda a alcançar o SP. Como a válvula é NF, para diminuir a vazão basta “retirar” sinal, ou seja, a variável linguística SAÍDA é negativa. Nota-se que a saída é representada aqui por sinal negativo apenas por caráter de diferenciação do sinal positivo representando então um decremento a ser dado no elemento final de controle para atuar na planta, não havendo como enviar um sinal negativo à válvula.

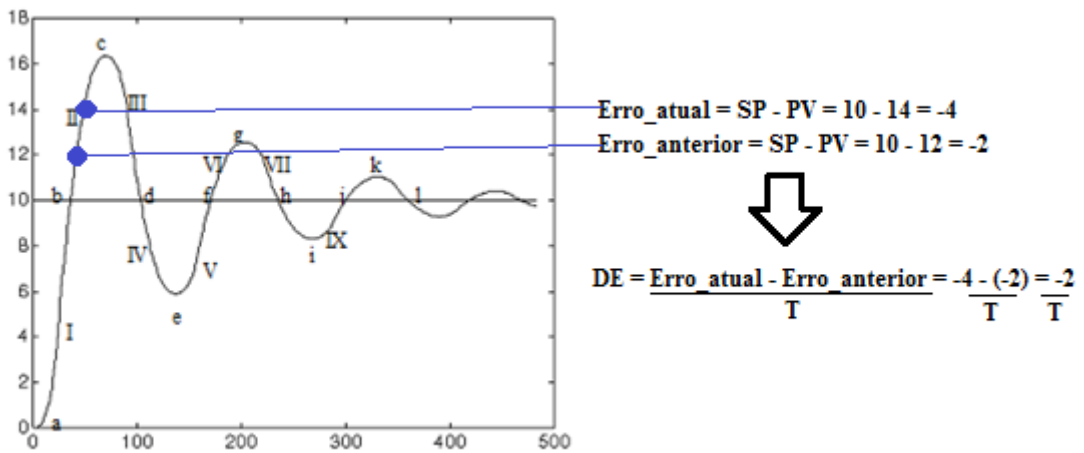


Figura 37 - Análise da região II da curva genérica do processo.

- **Região III** – o erro é negativo, indicando um valor acima do SP, e sua derivada é positiva. Este caso é semelhante à região I, uma vez que o sistema está se encaminhando para o SP.
- **Região IV** – o erro é positivo, indicando um valor abaixo do SP, e sua derivada é positiva. Neste caso é necessária a abertura da válvula para que a vazão aumente até o SP. Deste modo, envia-se sinal positivo à MV, uma vez que a mesma é NF.

De modo análogo, analisam-se todas as regiões desta curva. Como existem três estados para o erro e três para a taxa de variação do erro (Zero, Positivo e Negativo), têm-se nove possibilidades diferentes e, conseqüentemente, nove regras a serem analisadas. A Tabela III resume estas possibilidades. Conclui-se que a seguinte base de regras seria aceitável:

Tabela III - Resumo das regras Fuzzy.

		Derivada do Erro (DE)		
		Negativo	Zero	Positivo
Erro (E)	Negativo	N	N	Z
	Zero	N	Z	P
	Positivo	Z	P	P

Onde: N (negativa), Z (zero) e P (positiva) significam decrementar o sinal a ser enviado ao elemento final de controle, manter mesmo sinal para válvula e incrementar o sinal a ser enviado para a válvula de injeção de fluido, respectivamente.

A situação que representa a estabilidade do sistema é o erro e sua derivada com valores nulos e, deste modo, mantém-se o sinal de atuação na válvula para que ela permaneça como está com o objetivo de manter-se a estabilidade.

Outro resumo, agora de modo escrito, é mostrado na Tabela IV.

Tabela IV - Resumo das regras Fuzzy.

Regra ₁ : se (<i>E</i> é negativo e <i>DE</i> é negativa)	então <i>SAÍDA</i> é negativa
Regra ₂ : se (<i>E</i> é negativo e <i>DE</i> é zero)	então <i>SAÍDA</i> é negativa
Regra ₃ : se (<i>E</i> é negativo e <i>DE</i> é positiva)	então <i>SAÍDA</i> é zero
Regra ₄ : se (<i>E</i> é zero e <i>DE</i> é negativa)	então <i>SAÍDA</i> é negativa
Regra ₅ : se (<i>E</i> é zero e <i>DE</i> é zero)	então <i>SAÍDA</i> é zero
Regra ₆ : se (<i>E</i> é zero e <i>DE</i> é positiva)	então <i>SAÍDA</i> é positiva
Regra ₇ : se (<i>E</i> é positivo e <i>DE</i> é negativa)	então <i>SAÍDA</i> é zero
Regra ₈ : se (<i>E</i> é positivo e <i>DE</i> é zero)	então <i>SAÍDA</i> é positiva
Regra ₉ : se (<i>E</i> é positivo e <i>DE</i> é positiva)	então <i>SAÍDA</i> é positiva

Finalizada a criação da base de regras, a próxima seção utilizará o ambiente de simulação SIMULINK do *software* Matlab[®] para análise da resposta do processo diante do controlador projetado.

4.5 Simulação do Controlador *Fuzzy*

O passo-a-passo da criação do Controlador *fuzzy* por meio da *toolbox Fuzzy* do *software* Matlab[®] é tratado no Apêndice C.

Após a criação do Controlador *fuzzy* procedeu-se a simulação utilizando-se a *toolbox Simulink* do *software* Matlab[®]. Neste software, foi feito um posicionamento dos blocos de forma a simular e comparar a malha de realimentação negativa com controlador *Fuzzy*, malha com controlador PID e malha aberta. O diagrama de blocos está ilustrado na Figura 38.

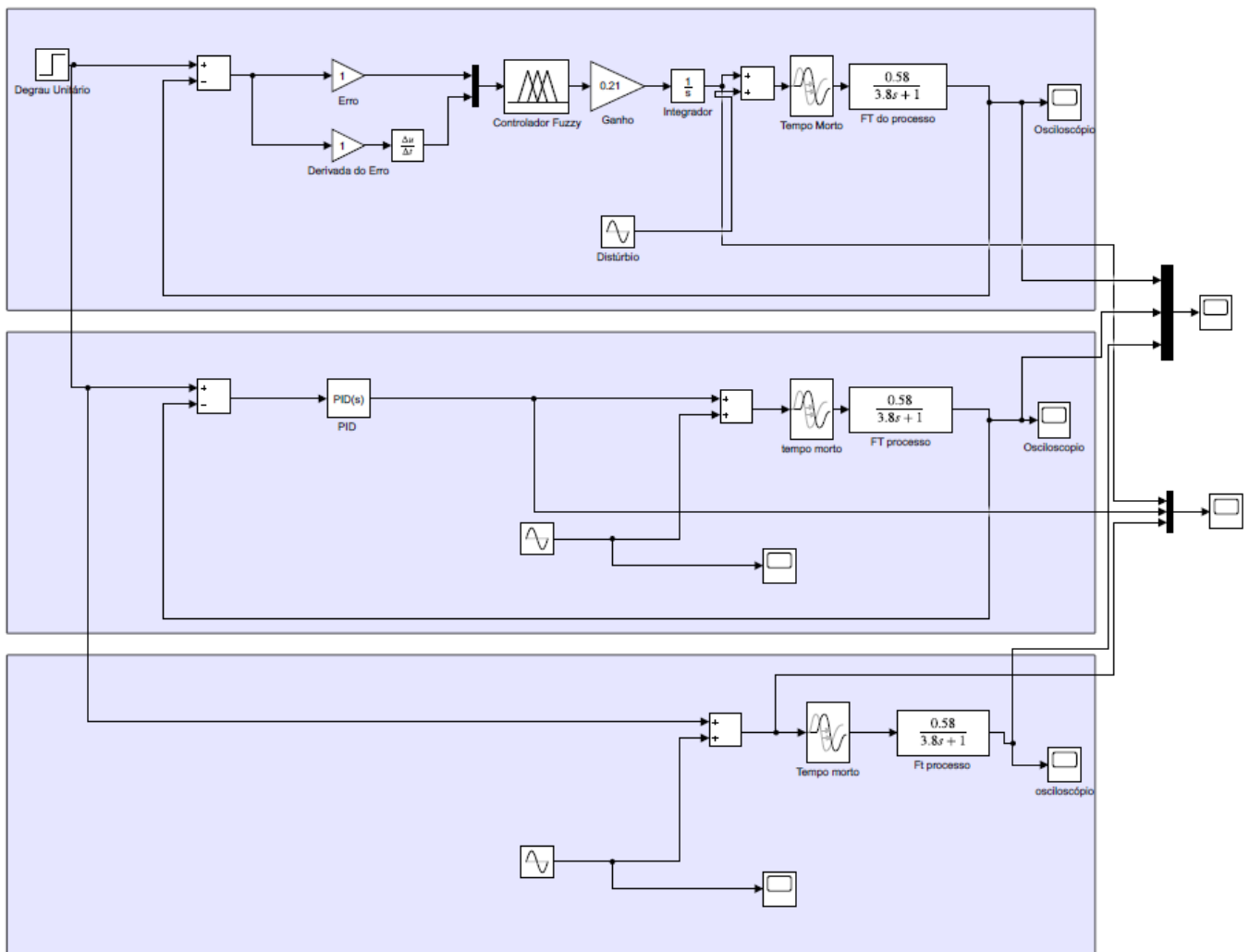


Figura 38 - Diagrama de bloco de uma planta de vazão de embebição controlada por um controlador *Fuzzy*, controlador PID e em malha aberta.

O controlador Fuzzy usará a estrutura apresentada na *seção 2.2* sendo as entradas E e DE, ou seja, as grandezas correspondentes ao erro e a derivada do erro, respectivamente. Como forma de simular a derivada do erro inseriu-se o bloco que fará a subtração entre o erro atual e o erro anterior dividido pelo período de amostragem.

Para se ativar o bloco do controlador *Fuzzy*, deve-se clicar no bloco e fornecer o mesmo nome que foi salvo, na sua criação, por meio da opção *Save To Workspace*.

Observe que o ganho na saída do controlador *Fuzzy* na Figura 38 tem o efeito de uma ação proporcional dependente. Alterando-se o seu valor, varia-se a velocidade de atuação do controlador *fuzzy*. O ajuste do ganho se dá também pelo fato de o controlador *fuzzy* ter uma ação muito abrupta o que ocasiona em frequentes pancadas na válvula levando a diminuição do tempo útil de vida e conseqüente aumento da frequência de manutenção. Observa-se também

que a função de transferência da planta não possui um integrador intrínseco, havendo a necessidade de incluir um integrador em série com o ganho na malha direta.

Foi inserida nas malhas fechadas (controlador *Fuzzy* e PID realimentados) e na malha aberta a função de transferência da planta da vazão de embebição que foi identificada utilizando-se o algoritmo AMCT (*Automatic Model Capture Technology*) do software PlantTriage® na *Seção 3.3*. Inseriu-se também o bloco *Delay* responsável em representar o tempo morto da planta.

Para comparar o desempenho do controlador PID com o controlador *Fuzzy*, realizou-se o cálculo dos parâmetros de sintonia do controlador PID utilizando-se o método de síntese direta. Os parâmetros calculados foram: $K_p = 0,727$ e $T_i = 5,22s$, sem ação derivativa.

Para verificar a resposta do sistema e realizar um comparativo entre a malha de realimentação negativa com controlador *Fuzzy*, malha com controlador PID e malha aberta, aplicou-se um degrau unitário no *Set-Point* da malha. O bloco *Scope* emula um osciloscópio plotando um gráfico do sinal de entrada. O gráfico gerado na simulação está representado na Figura 39 indicando a resposta da variável de processo (vazão de embebição).

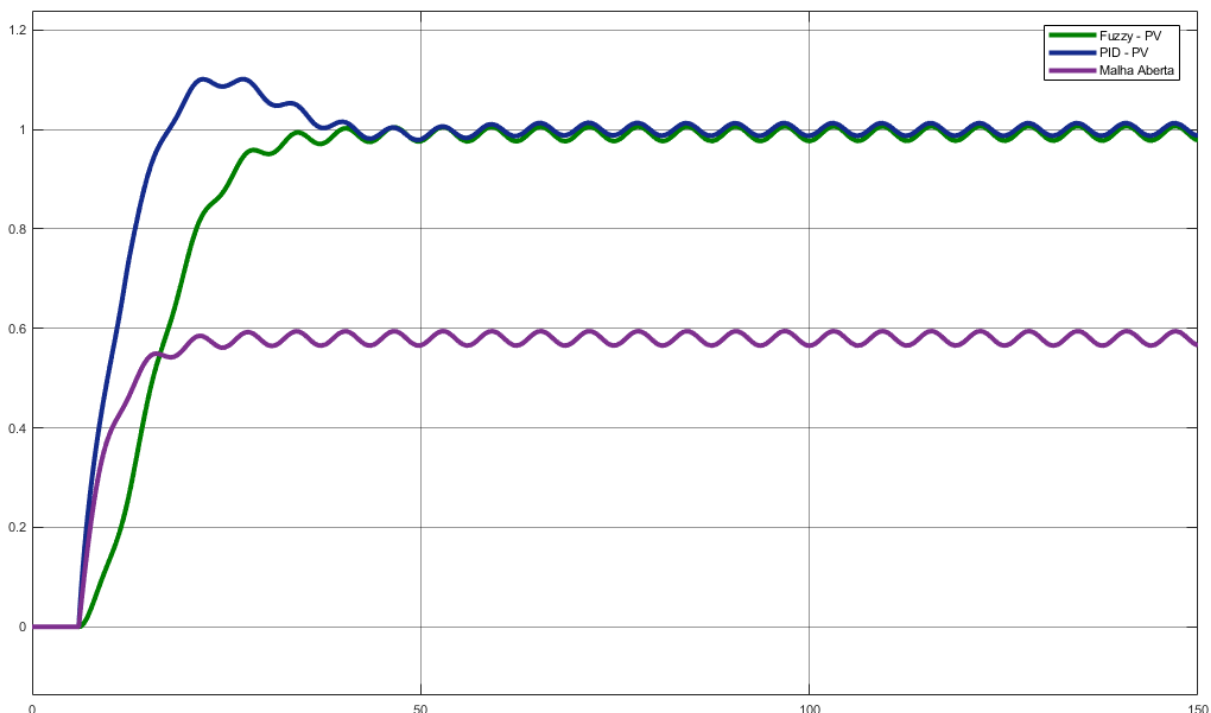


Figura 39 - Resposta da malha de controle com o controlador *Fuzzy*.

Como pode ser observado na Figura 39, o desempenho da malha com o controlador *Fuzzy* foi satisfatório, apresenta eliminação de offset e estabilidade em malha fechada.

4.6 Conclusão

Este capítulo apresentou os princípios básicos direcionados às aplicações industriais dos controladores *Fuzzy* detalhando sua estrutura e os conceitos de conjuntos *Fuzzy* e variáveis linguísticas.

Também foi descrito detalhadamente o ajuste e a simulação do controlador *Fuzzy*, apresentando os procedimentos necessários para simular a malha de controle utilizando-se as *toolbox Fuzzy* e *Simulink*. Na simulação foi verificada a necessidade de se inserir um integrador na saída do controlador devido à planta não possuir integrador intrínseco. Além disso, foi necessário adicionar um ganho em série com o integrador com o objetivo de uma ação proporcional dependente para variar a velocidade de atuação do controlador. De acordo com a análise da resposta da simulação conclui-se que a controlador implementado é capaz de controlar a planta que foi simulada. Pode-se, então, implementar esta estratégia de controle no CLP.

Capítulo 5 - Implementação do Controlador *Fuzzy* na Rotina do CLP

5.1 Introdução Geral

Este capítulo aborda os procedimentos necessários para a criação de uma rotina de controle inteligente em linguagem *ladder* por meio de ferramentas da Rockwell Automation. O controlador *Fuzzy* foi criado por meio do *software* FuzzyDesigner® e exportado com todas as configurações em forma de um *Add-On Instruction*. A instrução foi importada para o *software* RSLogix5000® em qual foi implementado na rotina *ladder*. Foi detalhado como se integra no RSLogix5000® o controlador *Fuzzy* criado por meio do *software* FuzzyDesigner®. Isso foi feito por meio de imagens e descrições detalhadas dos procedimentos necessários para construir sistemas *Fuzzy* por meio de diagrama de bloco contendo componentes da biblioteca da ferramenta. Foi detalhado como exportar o controlador para o formato XML. O objetivo é detalhar o passo-a-passo da utilização das ferramentas da Rockwell Automation para auxiliar usuários que venham a ter o primeiro contato com o controle inteligente e também contribuir para disseminar a utilização deste tipo de estratégia de controle por parte de alunos de graduação e pós graduação.

Por fim, após finalizar a implementação da rotina foi realizado o *download* do programa no CLP. Foi alterado o CLP para o modo *Run* para que a rotina seja executada no modo de monitoração realizando-se a análise dos resultados.

Espera-se ao final deste capítulo que se obtenha um *how-to* de utilização das ferramentas da Rockwell Automation com o foco na criação de rotinas de controle inteligente.

5.2 Parametrização do Controlador Fuzzy e Implementação na Rotina CLP

Foi utilizada a ferramenta FuzzyDesigner®, criada pela Rockwell Automation, para criação dos controladores do tipo *Fuzzy*. Esta ferramenta permite que o controlador *Fuzzy* possa ser exportado e utilizado nas mais diversas lógicas e rotinas de controle. Já o *software* RSLogix5000®, que já foi bastante utilizado no Capítulo 3, será novamente utilizado para dar continuidade na implementação da rotina de controle.

A justificativa em se adotar esta ferramenta é que o FuzzyDesigner[®] é projetado para trabalhar com a família de controladores Logix5000 da Rockwell Automation.

Na interface do *software* é inserido o conhecimento de um especialista referente a um determinado sistema, e a ferramenta é então capaz de gerar um arquivo do tipo XML chamado *AOI (Add-on instruction)*. A integração entre as ferramentas é feita pela geração deste *Add-On Instruction* que pode ser importado no RSLogix5000[®] e efetuada na lógica *ladder*.

O detalhe da montagem e parametrização do controlador *fuzzy* é descrito no Apêndice D. A criação do controlador é feita por meio de blocos de montagem que são configurados e interligados.

Após a criação do controlador realizou-se a importação no RSLogix[®] e o mesmo aparece como qualquer outro bloco de construção *Ladder*, especificamente na aba *Add-On*, assim como demonstrado na Figura 40.

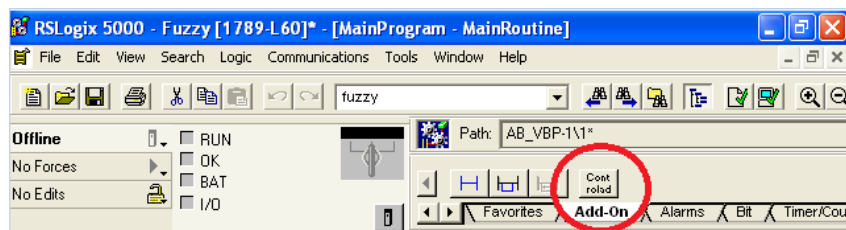


Figura 40 - Bloco de instrução do controlador *Fuzzy* na aba *Add-On*.

Para utilizá-lo, primeiramente, é necessário criar na lista de *TAGS* um objeto do mesmo tipo do controlador, ou seja, *Data Type* do tipo *FuzzyDesign*, assim como demonstrado na Figura 41. O *Data Type* deve ser definido com o mesmo nome do arquivo *AOI* importado na rotina.

	Name	Alias For	Base Tag	Data Type
	Fuzzy			ControladorFuzzy

Figura 41 - Criação da TAG do controlador *Fuzzy*.

Com isso, procedeu-se a implementação do controlador *Fuzzy* na rotina *ladder*. Na Figura 42 é demonstrada a rotina de controle após a sua montagem, e os seus componentes serão devidamente descritos de modo a tornar mais clara a necessidade de utilizar cada um.

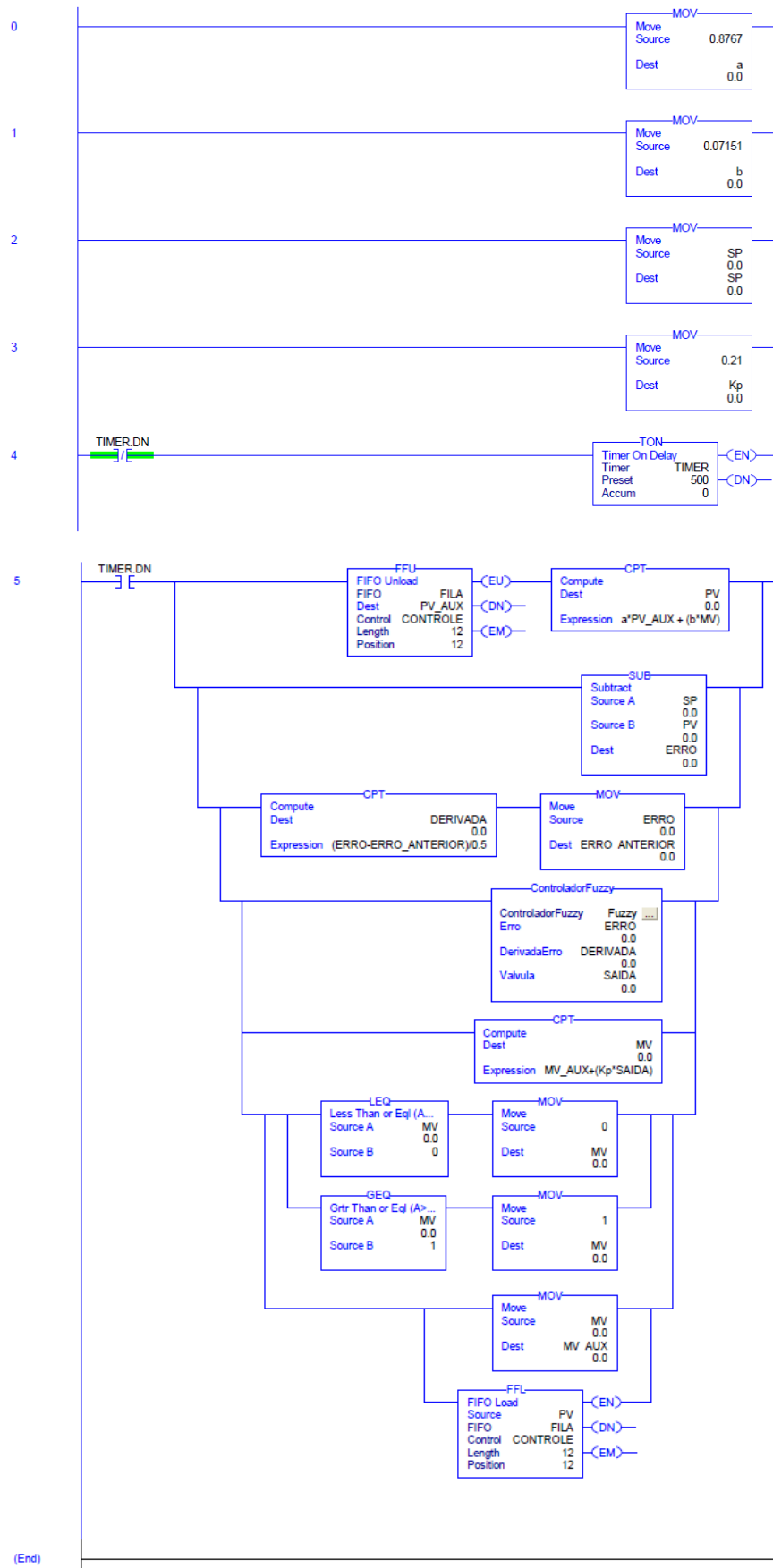


Figura 42 - Rotina ladder contendo o controlador Fuzzy.

Inseriu-se na linha 2 da rotina um bloco responsável por fazer alterações de *Set-Point* de acordo com as necessidades do usuário já que o bloco do controlador não dispõe dessa variável pré-definida conforme existe no PID. Na linha 3 inseriu-se um bloco responsável por possibilitar alterações no parâmetro K_p , utilizado para permitir a aplicação de um ganho proporcional na saída do controlador. Na linha 4, conforme comentado na *seção 3.6*, foi inserido um bloco temporizador que é responsável por manter a rotina executando indefinidamente para simular a planta. A linha de número 5 é a linha onde é calculada a ação do controlador *Fuzzy* atuando na planta escolhida, e diversos blocos foram inseridos para permitir estes cálculos. Foi mencionado que para este exemplo criado, o controlador *Fuzzy* estaria trabalhando com dados de erro e derivada do erros alimentados em sua entrada, e para isso, ambas as variáveis precisam ser devidamente calculadas. Sendo assim, o bloco SUB é o responsável em calcular o erro subtraindo-se o valor de SP – PV, já que o modo de controle é reverso. A variável *ERRO_ANTERIOR* foi utilizada para receber o valor de erro da iteração anterior, já que a derivada do erro por definição é calculada por $\frac{E_{atual} - E_{anterior}}{T}$ em que o período adotado foi de 0,5s. Adicionou-se também o bloco referente ao controlador *Fuzzy* contendo a base de regra gerada por meio do *software* FuzzyDesigner®.

Desta forma, ambas as variáveis de entrada do controlador são devidamente calculadas, de modo que a saída do controlador nomeada de SAIDA retorna um valor entre -1 e 1, de acordo com a parametrização feita no FuzzyDesigner®. Este valor é então utilizado para a definição da saída final de controle, que necessita de ser transformada em um valor de 0 a 100%, de modo que será o próprio valor usado para determinação da abertura da válvula de controle, por exemplo. A saída final do controlador foi então determinada $MV = MV_AUX + K_p * SAIDA$, de modo que a saída do controlador *Fuzzy* gera incrementos ou decrementos na saída geral para a válvula, de acordo com o seu valor atual, isso ainda se considerando a intensidade dos passos, de acordo com o valor do parâmetro K_p .

Dois blocos de comparação e de movimentação foram utilizados para transformar a saída em 0 ou 1, caso os valores em um determinado momento se encontrassem negativos ou mesmo maiores que 1. Esse procedimento é necessário para limitar o sinal de atuação na válvula entre 0 e 100%.

Os demais blocos da rotina da Figura 42, tais como FFU, FFL, TIMER, já foram comentados na *seção 3.6* e suas funções na rotina permanecem as mesmas.

Finalizada a implementação da rotina foi realizada a verificação da lógica da rotina a fim de verificar erros de programação. Este procedimento foi realizado selecionando a opção

Logic → *Verify* → *Routine*. Conforme mostrado na Figura 43 não foram encontrados erros na rotina de programação.

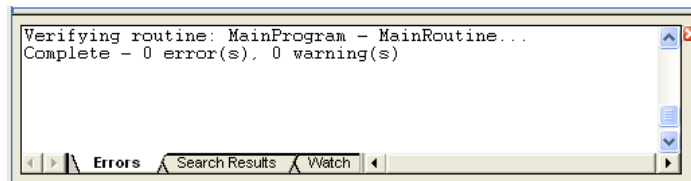


Figura 43 - Janela de visualização de erros para a rotina completa.

Após programar e verificar a rotina foi feito o *download* para enviá-la para o controlador. Após o *download* da rotina, se o CLP estiver no modo *Remote Run* haverá mudança para o modo *Program* suspendendo a execução da rotina carregada anteriormente e substituindo pela nova rotina. Depois de carregado, foi necessário alterar o modo de operação mudando a posição da chave seletora do controlador por meio do *software* SoftLogix Chassis Monitor[®]. Também foi necessário alterar o modo de operação no RSLogix5000[®] para *Run Mode*. No modo *Run* a nova rotina será executada em modo de monitoração e todos os elementos de entrada e saída da rotina irão apresentar cores para representar o seu nível lógico conforme mostrado na Figura 44 e Figura 45.

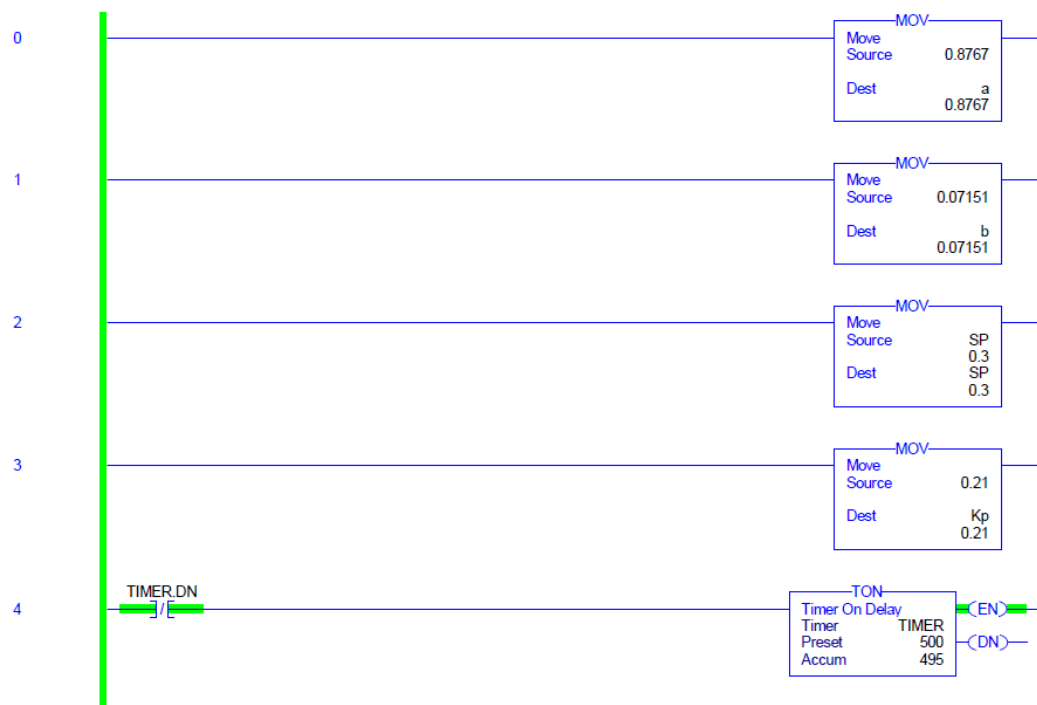


Figura 44 - Rotina *ladder* em execução – parte I.

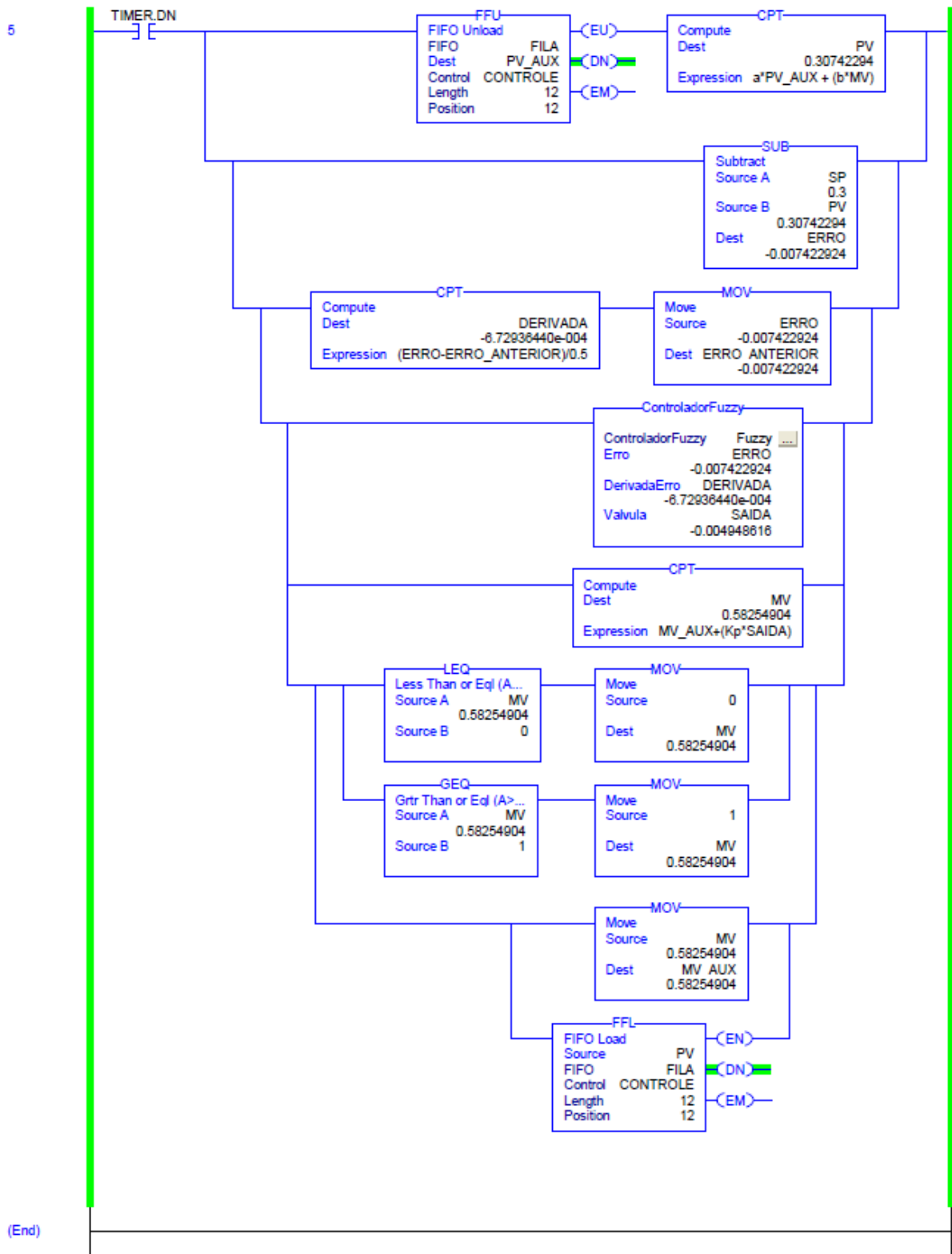


Figura 45 - Rotina ladder em execução – parte II.

No modo *Run* o RSLogix5000® permite monitorar as variáveis *on-line*. Ao clicar com o botão direito do mouse em *Controller Tags* é possível selecionar a opção *Monitor Tags* onde

se podem monitorar todas as variáveis disponíveis para o uso do CLP conforme ilustrado na Figura 46.

Name	Value	Force Mask	Style	Data Type
a	0.995		Float	REAL
b	0.005		Float	REAL
± CONTROLE	{...}	{...}		CONTROL
DERIVADA	1.73389912...		Float	REAL
DERIVADA_ANTERIOR	0.0		Float	REAL
ERRO	-6.2699616...		Float	REAL
ERRO_ANTERIOR	-6.2699616...		Float	REAL
± FILA	{...}	{...}	Float	REAL[10]
± Fuzzy	{...}	{...}		ControladorFuzzy
Kp	0.2		Float	REAL
MV	0.108149916		Float	REAL
MV_AUX	0.108149916		Float	REAL
PV	0.30626997		Float	REAL
PV_AUX	0.30726147		Float	REAL
SÁIDA	-4.0636765...		Float	REAL
SP	0.3		Float	REAL
± TIMER	{...}	{...}		TIMER

Figura 46 - Monitoração on-line de todas as TAGs.

É possível observar na rotina em execução o valor da variável de processo se aproximando do valor de 0.3 (30%) escolhido anteriormente como *Set-Point*.

Para se verificar o comportamento da planta em um determinado intervalo de tempo qualquer, a resposta do processo pode ser visualizada clicando-se com o botão direito do mouse na *tag* desejada e escolher a opção *Trend "PV"*. Com isso, é possível se criar um gráfico personalizável. A resposta da variável de processo está ilustrada na Figura 47.

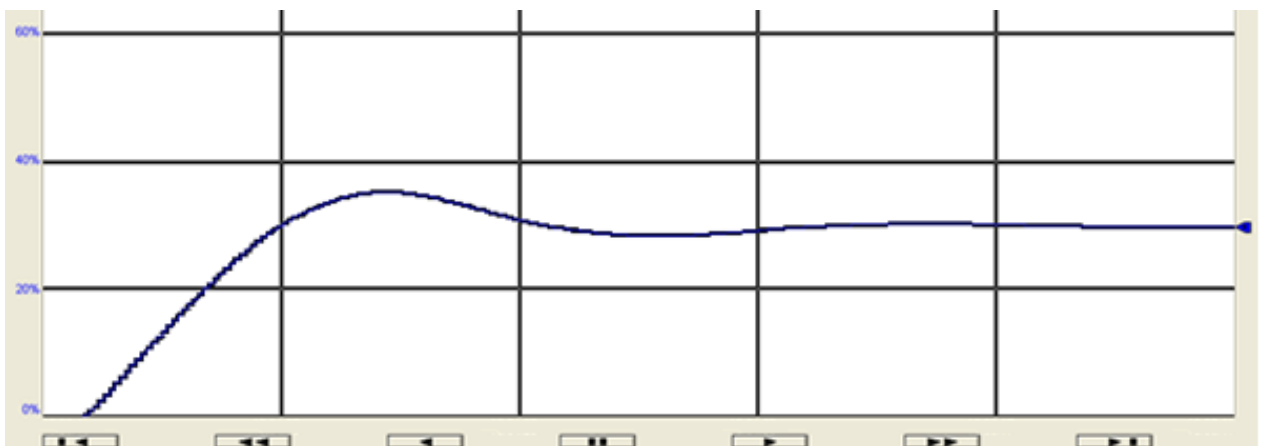


Figura 47 - Resposta da variável de processo.

5.3 Conclusão

Este capítulo abordou os procedimentos necessários para implementar o controlador *Fuzzy* em uma rotina *ladder* utilizando-se ferramentas da Rockwell Automation. Foi necessário criar o controlador e sua base de regras por meio do *software* FuzzyDesigner[®] exportando-o na forma de um *Add-On Instruction*. Dando sequencia, esta instrução foi importada para o *software* RSLogix5000[®] onde foi implementado na rotina *ladder*. Ao final do capítulo foi possível executar a rotina e verificar por meio de monitoração das variáveis que a variável de processo se aproximou do *Set-Point* e que o erro de regime estacionário foi eliminado.

Capítulo 6 - Desenvolvimento do Sistema Supervisório

6.1 Introdução Geral

Em se tratando de um sistema automatizado, surge a necessidade de se ter uma interface amigável para facilitar o trabalho do operador, sendo que é complexo analisar diretamente do CLP. Esta interface amigável é também chamada de supervisório.

A confecção de tela foi desenvolvida por meio do *software* Factory Talk View Studio[®] cuja aplicação é voltada à criação de telas gráficas de interação com CLPs tendo como função proporcionar uma visão geral e intuitiva do processo.

Na *seção* 6.2 são detalhados por meio de figuras, os procedimentos executados para desenvolver uma tela.

6.2 Criação do Sistema Supervisório

Para criar o projeto, ao abrir o *software* Factory Talk View Studio[®] selecionou-se a opção *Site Edition Network* conforme ilustrado na Figura 48.

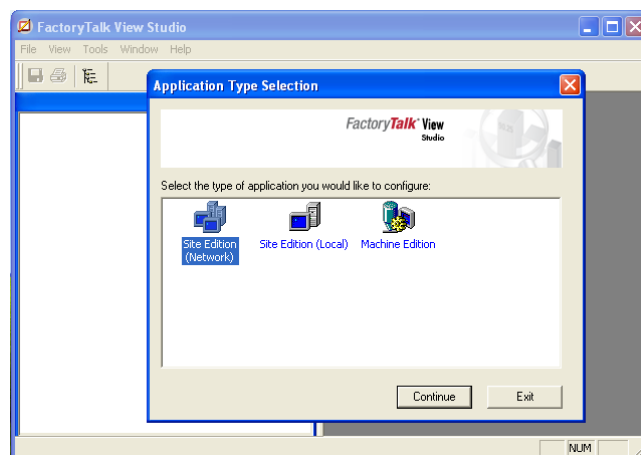


Figura 48 - Criando-se um projeto no *software* Factory Talk[®].

Clicou-se em *New* → *Open Site Edition Network* onde foi inserido o nome do projeto conforme ilustrado na Figura 49.

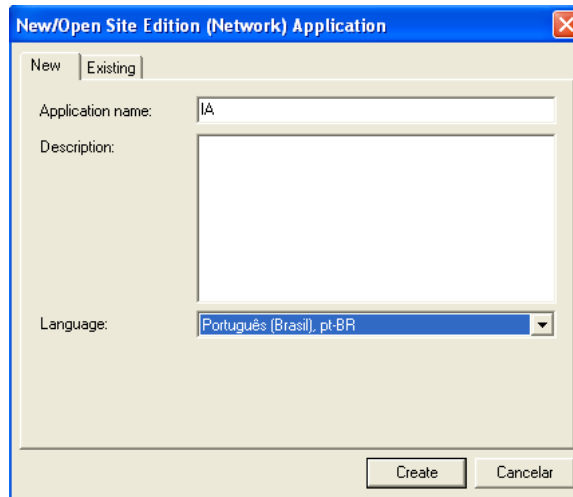


Figura 49 - Nomeando o projeto.

Após criar o projeto aparece a tela conforme apresentado na Figura 50.

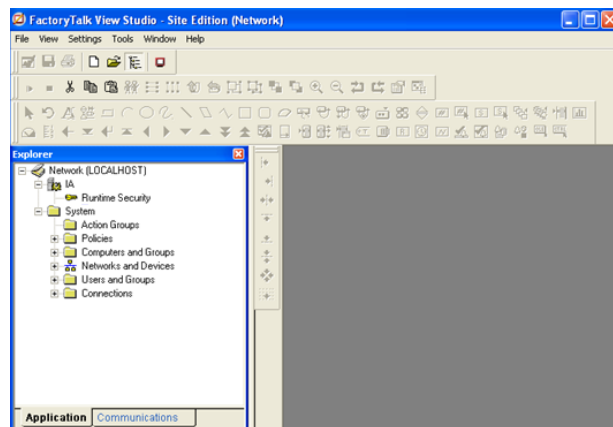


Figura 50 - Tela principal do software Factory Talk®.

Foi necessário criar uma nova área, clicando-se com o botão direito no ícone com o nome do projeto e escolhendo a opção *New Area* conforme apresentado na Figura 51.

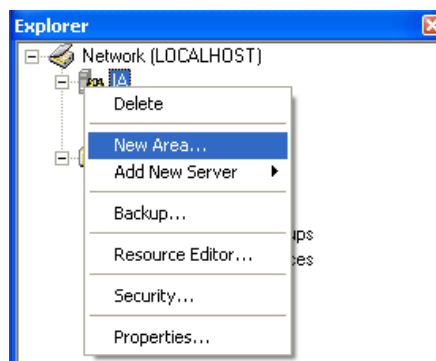


Figura 51 - Criação de uma nova área para criação do supervisório.

Nomeou-se a nova área de “supervisório”. Uma pasta com o nome supervisório é criada na janela *explorer*. Clicou-se com o botão direito sobre ele selecionando *Add New Server* → *HMI Server* conforme ilustrado na Figura 52. O *HMI Server* foi nomeado de “HMI”.

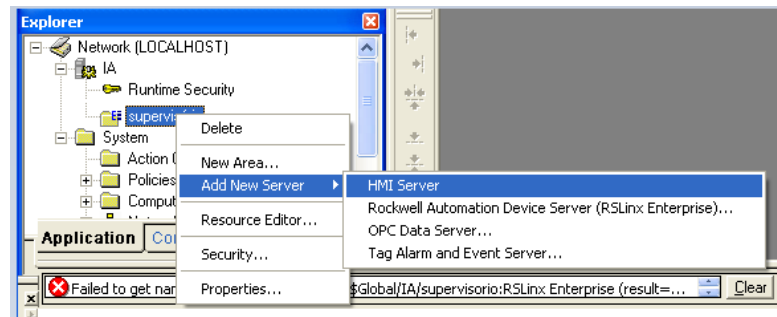


Figura 52 - Adicionando um HMI Server ao projeto.

Finalizado a criação do HMI, aparece a janela *Add Process Faceplates* onde se selecionou as opções ilustradas na Figura 53.

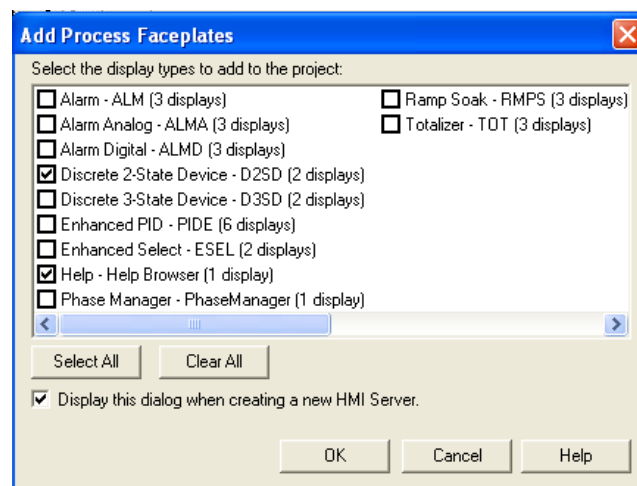


Figura 53 - Configurações do HMI Server.

Clicou-se com o botão direito na pasta principal selecionando-se a opção *Add New Server* → *Rockwell Automation Device Server* conforme ilustrado na Figura 54.

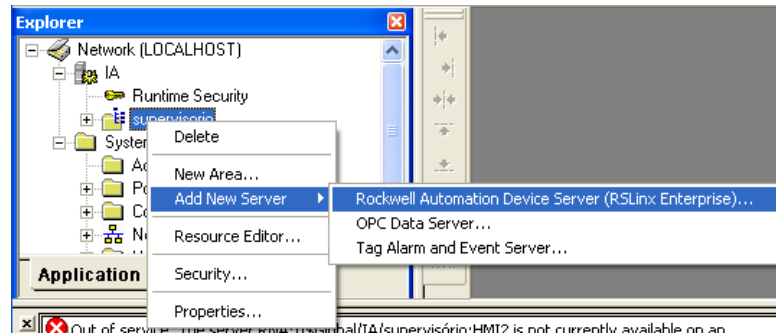


Figura 54 - Adicionando o dispositivo RSLinx Enterprise.

Neste momento, foi aberta uma tela de configuração de propriedades onde foram feitas modificações na aba de *Alarms and Events* conforme observado na Figura 55.

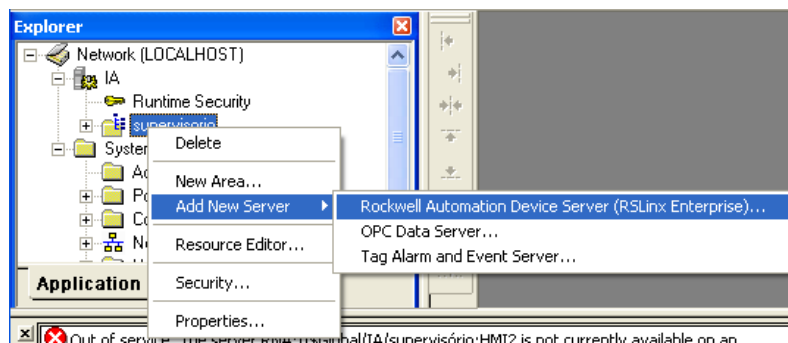


Figura 55 - Configurações do novo Server.

Realizados os procedimentos, tem-se agora no *explorer* do projeto o servidor RSLinx criado contendo a opção *Communication Setup*. Clicando-se nesta opção é exibida a tela ilustrada na Figura 56 a qual é realizada a configuração para comunicar o computador com o CLP.

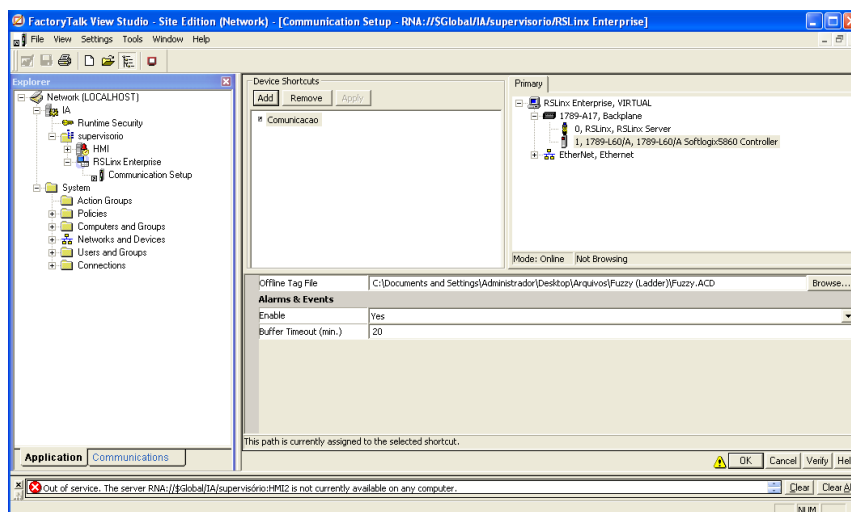


Figura 56 - Configurando a comunicação.

Os procedimentos realizados até aqui estão relacionados à conexão entre o CLP e o sistema supervisorio. Então, procedeu-se a criação da tela do supervisorio. A partir deste instante, o programa criado do RSLogix5000® deve estar no modo *Run*. Para criação da tela, retornou-se a janela *explorer* do projeto selecionando as opções *Displays* → *New* conforme na Figura 57.

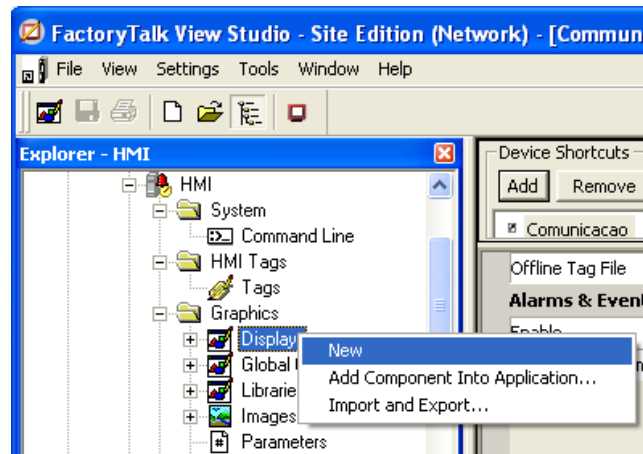


Figura 57 - Procedimento para criação da tela do supervisorio.

Para o desenvolvimento do supervisorio foram analisadas situações importantes para estruturação do sistema dentre as quais, tem-se:

- ✓ Definição dos dados que estarão presentes na tela do supervisorio;
- ✓ Definição de símbolos, cores e animações para objetos envolvidos no processo industrial que será simulado tendo clareza de entendimento e padronização;
- ✓ Layout e funcionalidade da janela de operação;

A ideia é desenvolver uma tela que permita a supervisão do processo de controle de embebição por meio de telas que possuam animações em função das informações recebidas pelo CLP. Para isso, utilizou-se para representação do sistema de vazão símbolos convencionais do próprio *software*, conforme pode ser visto pela Figura 58, que podem ser de fácil entendimento, tais como tanque e válvula.

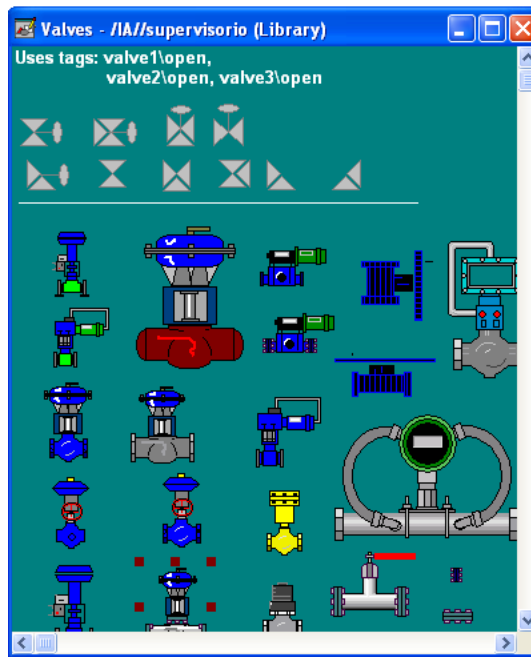


Figura 58 - Símbolos convencionais de válvulas oferecidos pelo software Factory Talk®.

Para desenvolver a tela de supervisorio foi necessário adicionar alguns *displays* como, por exemplo, um *display* para alterar o valor de *Set-Point*. O campo de *input* utilizado para inserir *Set-Point* foi criado selecionando-se as opções *Objects* → *Numeric Input* conforme na Figura 59.

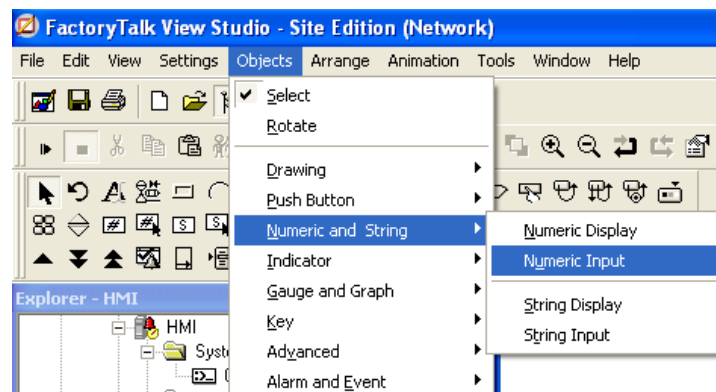


Figura 59 - Inserindo um campo de input.

Nas propriedades do campo *input* que foi criado é possível conectá-lo com qualquer TAG criada na rotina *ladder*. Este procedimento é realizado selecionando a opção *Tag* na aba *Connections* conforme demonstrado na Figura 60.

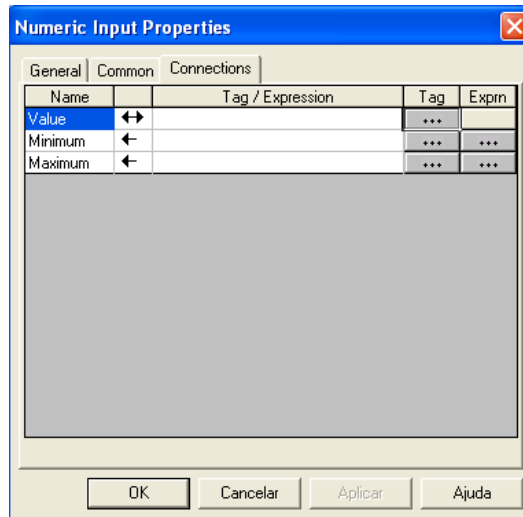


Figura 60 - Procedimento para vincular o Numeric Input a uma Tag declarada na rotina ladder.

Em Tag, selecionou-se a tag da rotina ladder referente ao Set-Point ou qualquer outra variável que se deseja representar no *display* do Factory Talk®. A tela onde foi selecionado a tag da rotina ladder está ilustrada na Figura 61.

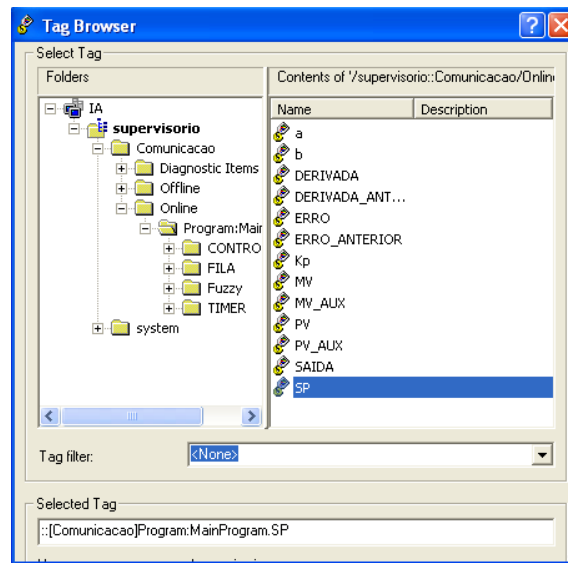


Figura 61 - Seleção das variáveis declaradas na rotina ladder.

Na Figura 62, é demonstrado o sistema supervisorio em funcionamento. Realizou-se um teste escrevendo 30% de Set-point e por meio do gráfico da vazão na tela de supervisorio foi possível observar a curva de resposta da vazão com atuação do controle fuzzy no elemento final de controle.

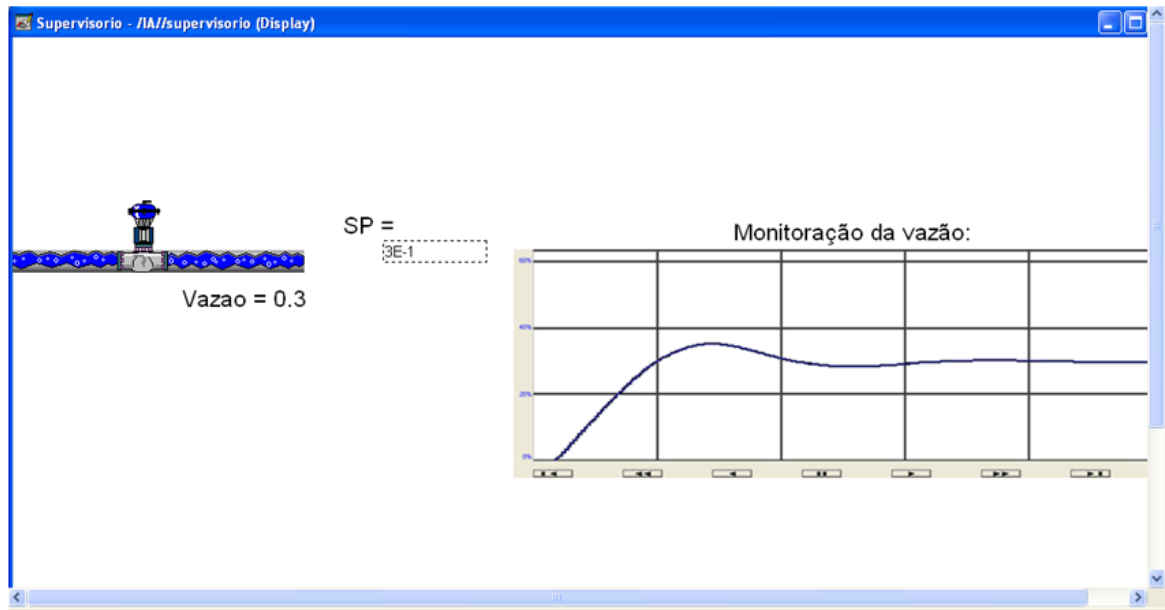


Figura 62 - Tela do sistema supervisorio do controle de vazão de embebição.

6.3 Conclusão

Neste capítulo foram descritos os procedimentos necessários para desenvolver uma tela de sistema supervisorio por meio do *software* Factory Talk[®]. Foi possível desenvolver um sistema supervisorio capaz de se tornar uma importante ferramenta para análise e testes de estratégias de controle inteligente que poderiam ser impossíveis em um ambiente industrial. O sistema supervisorio desenvolvido permite que o sistema de simulação do controle inteligente da vazão de embebição no CLP opere de forma que o usuário possa utilizá-lo diretamente de um sistema supervisorio facilitando a análise do processo.

Capítulo 7 - Conclusões Gerais

Foi possível desenvolver a proposta deste trabalho que era viabilizar a criação de um ambiente de simulação de controle inteligente de processos por meio da implementação de uma rotina de controle *Fuzzy* em linguagem *ladder* no *software* RSLogix 5000[®]. Também foi possível desenvolver um guia completo de como se utilizar as ferramentas da Rockwell Automation, desde a construção do controlador *Fuzzy* até os passos da própria integração do FuzzyDesigner[®] com o RSLogix 5000[®], tornando o uso mais claro para qualquer usuário que já tenha tido contato com programação em *ladder*.

Foi possível integrar a rotina a um sistema supervisório, com simulação da planta do processo na própria ferramenta, tornando possível que o usuário teste e faça simulações mesmo sem ter acesso a plantas e sistemas físicos reais. Este sistema supervisório é capaz de se tornar uma importante ferramenta para análise e testes de estratégias de controle que poderiam ser impossíveis em um ambiente industrial. O *software* FactoryTalk[®] apresentou recursos suficientes sendo possível utilizar este *software* para tornar o sistema mais próximo de uma aplicação real, facilitando a operação.

A implementação do controlador Fuzzy apresenta como vantagem não necessitar conhecer a função de transferência da planta, ao contrário do PID, que necessitou do modelo para realizar a sintonia dos parâmetros do controlador. Além disso, a situação de controle de vazão de embebição onde o sistema de controle seja baseado em regras empíricas, na prática pode ser mais eficaz, além de poder atuar em regiões de não linearidades.

Com base na teoria de sistemas dinâmicos lineares e na área de controle, o controlador *Fuzzy* pode ser também atraente para plantas as quais o seu modelo matemático seja difícil de estimar ou plantas mais complexas e com alto grau de dificuldade para análise.

Observou-se a simplicidade do FuzzyDesigner[®] e que o projeto de um controlador *Fuzzy* é simples não exigindo grandes conhecimentos matemáticos do sistema, mas sim, conhecimento do funcionamento do sistema para criação da base de regras, ou seja, a maior dificuldade é conceitual de controle e não da utilização das ferramentas.

Portanto, o ambiente de simulação permite que a simulação de processos possa ser desenvolvida em ambiente industrial por pessoal não especializado em técnicas de simulação. Desta forma, o técnico ou o engenheiro de controle pode realizar experiências de simulação antes de implantar as estratégias de controle no processo real.

Referências

- DABNEY, James B. e HARMAN Thomas L., “*Mastering Simulink*”, Ed. Pearson Education, 2003.
- FONSECA, M. O.; SILVA, C.; TORRES, B. S. **Avaliação de desempenho e auditoria de malhas de controle**. Revista InTech, n. 63, p. 32-37, 2004.
- GARANITO, Rafael Domingos. **Equações de diferenças e aplicações**. Dissertação de mestrado apresentada ao Departamento de Matemática e Engenharias da Universidade da Madeira. Agosto, 2006.
- HAROLDO, Rodrigues Azevedo. **Conjuntos Fuzzy aplicados ao controle de processos**. Apostila, 1999. Disponível em: <pt.scribd.com/doc/53559102/Apostila-Fuzzy-Prof-Haroldo>. Acesso em: 12/01/2019.
- HOLLAND, J. H. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- IEC - International Electrotechnical Commission. Standard 61131-3: **A Programmable Logic Controllers, Part 3: Languages**. 1992.
- JERRY M. Mendel, “*Fuzzy Logic Systems for Engineering: A Tutorial*”, Proc. of the IEEE”, vol 83, No. 3, 1995.
- MAMDANI, E. H. e Assilian S. (1975) **An experiment in linguistic synthesis with a fuzzy logic controller**. Int. J. Man-MachineStudies, vol. 7,1973.
- MANELLA, José Nazareno Anzanello. **Glossário de termos técnicos usados na indústria açucareira e alcooleira**. Maracaí: Markgraf, 2012.
- MELLO, F. M. Controle nebuloso (“Fuzzy”) aplicado à combustão em caldeiras a bagaço de cana. Dissertação de Mestrado em Engenharia Química – Universidade Federal de São Carlos, 2018.
- NETO, A. Z; **Análise do Impacto da Comunicação em Redes Foundation Fieldbus no Desempenho de Sistemas de Controle**. Dissertação de Mestrado em Engenharia Elétrica – Universidade Federal do Rio Grande do Sul, 2007.
- NISE, Norman S.. **Engenharia de Sistemas de Controle**. 5ª Edição. Rio de Janeiro: LTC, 2009.
- OLIVEIRA, D. N. e BRAGA, A. P. S. e ALMEIDA, O. M. “**Fuzzy Implementado em ladder com Funções de Pertinência Descontínuas**”. Universidade Federal do Ceará, 2010. XVIII Congresso Brasileiro de Automática.

ORTEGA, N. R. S., “**Aplicação da Teoria de Conjuntos Fuzzy a Problemas da Biomedicina**”. Dissertação de doutorado submetida ao Instituto de Física da USP. São Paulo, 2001.

PEDRYCZ, W, e GOMIDE, F..**An Introduction to Fuzzy Sets**.The MIT Press, 1998, ISBN 0-2620-16171-0.

REZNIK, L. “**Fuzzy Controllers. Newnes**”, Reino Unidos, 1996.

ROCKWELL AUTOMATION. **FactoryTalk** – Disponível em: <www.rockwellautomation.com/rockwellsoftware/factorytalk/activation/whatsnew.html> Acesso em: 01/12/2018.

ROCKWELL AUTOMATION. **RSLogix5000Fuzzy Designer – User Manual**. Disponível em:<literature.rockwellautomation.com/idc/groups/literature/documents/um/logix-um004_-en-p.pdf>. Acesso em: 01/12/2018.

ROCKWELL AUTOMATION. **RSLogix5000 Controllers General Instructions Reference Manual. 2012**. Disponível em:<literature.rockwellautomation.com/idc/groups/literature/documents/rm/1756-rm003_-en-p.pdf>. Acesso em: 01/12/2018.

ROCKWELL AUTOMATION. **SoftLogix 5800 System – User Manual. 2012**. Disponível em: <literature.rockwellautomation.com/idc/groups/literature/documents/um/1789-um002_-en-p.pdf>. Acesso em: 01/12/2018.

ROCKWELL AUTOMATION. **RSLogix 5000 FuzzyDesigner – User Manual. 2007**. Disponível em: <literature.rockwellautomation.com/idc/groups/literature/documents/um/logix-um004_-en-p.pdf>. Acesso em: 01/12/2018.

RUBAAI, A., M. J. C. SITIRICHE & A. R OFOLI (2008), 'Design and implementation of parallel fuzzy pid controller for high-performance brushless motor drives: An integrated environment for rapid control prototyping', IEEE Transactions on Industrial Electronics 44, 1090-1098.

SIMÕES, M.G.; SHAW, I.S. **Controle e Modelagem Fuzzy**. Edgard Blücher, 2007.

SOPRANI, L. R. **Controle Automático de Processos Industriais Utilizando Controlador Lógico Programável (PLC) e Sistema Supervisório para Controle de Temperatura de Líquidos em Tubos**. 2006. Monografia (Graduação em Engenharia de Controle e Automação), Escola de Minas, Universidade Federal de Ouro Preto -UFOP, Ouro Preto: UFOP, 2006.

SUGENO, M. & URAKAMI, K. “**An Experimental StudyonFuzzy Parking Controlusing a Model Car**”, Industrial Applications of Fuzzy Control, Sugeno, M. Ed. Amsterdam: North-Holland, 1985.

YEN, J. “**Logic Fuzzy: A Modern Perspective**. IEEE Transactions on Knowledge and Data Engineering” vol. 11, no. 1, pp. 153-165, 1999.

ZADEH, L. A. “**Fuzzy sets**”. Information and Control, vol. 8, 1965.

Apêndice A – Artigos Publicados e Prêmio

O artigo “Criação de um ambiente de simulação de controle avançado de processos em uma usina sucroenergética “ publicado e apresentado no II EDEPA 2018, ganhou o prêmio de Menção Honrosa conforme mostrado na Figura 63.



Figura 63 - Menção honrosa no II EDEPA Uniube 2018.

Criação de um ambiente de simulação de controle avançado de processos em uma usina sucroenergética

Pinheiro Neto L.R.¹, Teixeira E.P.²

^{1,2} Universidade de Uberaba, Programa de Pós Graduação em Engenharia Química – Mestrado Profissional;

* Edilberto: edilberto.teixeira@gmail.com; Tel.: +55(34) 99977-3704

* Luciano: luciano.pinheiro@tereos.com; Tel.: +55(34) 99995-1432

RESUMO: Com a evolução da capacidade computacional dos modernos controladores lógicos programáveis, tornou-se possível a implementação de estratégias avançadas de controle que tradicionalmente são realizadas em sistemas digitais de controle distribuído. Desta forma, os controladores lógicos programáveis estão cada vez mais competitivos na área de controle avançado de processos. De forma geral, este trabalho apresenta a criação de um ambiente de simulação de controle avançado de processos por meio de uma rotina de controle no software RSLogix5000®, com a utilização da ferramenta FuzzyDesigner® para a criação do controlador do tipo Fuzzy. Utilizou-se o software Matlab® como suporte para o cálculo de parâmetros do controlador e o software PlantTriage® para identificação da função de transferência do processo. Neste trabalho, foi simulado o controle de vazão de embebição utilizando-se um controlador Fuzzy cuja topologia tem-se como entradas o erro e a derivada do erro. Foi adotado o controlador SoftLogix5860®, no entanto, os conceitos que serão utilizados são aplicáveis à maioria dos controladores programáveis comuns. A planta do processo foi simulada no próprio CLP e o sistema opera de forma integrada com um supervisório desenvolvido através do software FactoryTalk®. Com o término do trabalho foi obtido como resultado uma rotina em ladder no CLP capaz de simular sistemas industriais a partir de modelos matemáticos, gerando gráficos de forma intuitiva permitindo que estratégias de controle sejam previamente testadas nas indústrias pelo engenheiro de automação, em um ambiente de simulação, antes de serem implantadas no processo industrial. Além disso, foi feito um how-to de utilização de ferramentas com o foco na criação de rotinas de controle inteligente, em alternativa ao controle tradicional PID, e foi possível se concluir a simplicidade de uso da ferramenta FuzzyDesigner® mesmo se considerando a complexidade do assunto que é a lógica Fuzzy.

Palavras-chave: Ladder, Supervisório, Controlador Fuzzy, Lógica Fuzzy, Controle de Processos.

1. Introdução

A competição global e a pressão dos acionistas fazem com que a indústria tenha que superar metas. Para permitir a continuidade da empresa no mercado que exige rápidas mudanças, este setor tem focado muito em melhoria contínua, em aumento da eficiência operacional, redução de custos, aumento da qualidade do seu produto final e investimento em projetos de otimização do seu processo industrial. Com toda essa dificuldade que o setor sucroenergético tem vivenciado, as linhas de processo de uma planta devem estar continuamente em melhoria, mantendo a sustentabilidade dos negócios.

O sistema de controle de processo automatiza ambos, rotinas e condições anormais de operação, e auxilia a operação a manter o controle de seu processo. O desempenho do processo, e, portanto, os resultados dos negócios aumentam através de operações aplicadas e conhecimento do processo, via sistemas de controle de processos. Infelizmente, muitos estudos nas indústrias têm mostrado que grande parte das malhas de controle não operam em modo normal, sofrem oscilação e aumentam a variabilidade do processo, reduzindo a eficiência da planta. O impacto direto no desempenho do processo é significativo, podendo citar: perda

de produção, problemas de qualidade, consumo excessivo de energia, incidentes com segurança e meio ambiente e paradas não planejadas e alto custo de manutenção.

Nesse contexto, este trabalho tem como objetivo primário a criação de um guia ou passo-a-passo de como se utilizar o software FuzzyDesigner® e de como integrar o controlador criado nesta ferramenta no RSLogix5000®. Neste trabalho foi adotado o controlador SoftLogix5860, no entanto, os conceitos que serão utilizados são aplicáveis à maioria dos controladores lógicos programáveis comuns. O sistema proposto neste trabalho viabilizará tanto a simulação de processo de vazão de embebição, como a implantação de estratégias de controle inteligente nos controladores lógicos programáveis. Tal sistema permitirá que estratégias de controle sejam previamente testadas nas indústrias, em um ambiente de simulação, antes de serem implantadas no processo industrial.

O foco em buscar aumento da eficiência industrial motivou a parceria empresa-universidade sem a qual este trabalho não seria possível. O autor deste trabalho é colaborador no grupo e possui acesso aos dados de sistema de supervisão, laboratório e aos equipamentos industriais.

2. Função de Transferência do Processo

Uma importante operação unitária na usina é a extração do caldo da cana-de-açúcar que é realizado pelas moendas ou por difusores. Este processo consiste em separar o máximo possível de caldo da cana-de-açúcar. É importante que o controle desta área esteja otimizado para extrair o máximo possível de caldo o que significa aumento de produtividade para a unidade industrial. Na extração do caldo da cana-de-açúcar, um dos controles de processo é conhecido como controle de vazão de embebição que tem como objetivo medir e controlar a vazão de água utilizada para a lavagem e “amolecimento” da cana-de-açúcar.

Segundo Manella (2012), o processo no qual a água ou caldo é aplicado ao bagaço que sai dos ternos das moendas, a fim de aumentar a extração de sacarose de um conjunto de moendas.

Utilizou-se para identificação do modelo do processo a funcionalidade AMCT (Automatic Model Capture Technology) do software PlantTriage®. O AMCT funciona em real-time procurando e reportando modelos de processos por meio da avaliação automática de todos os testes de resposta que ocorrem na malha de controle, tais como: mudanças de set point, rampas ou movimentos de válvula no modo manual. Ele filtra os dados e procura locais em que a variável de processo (PV) e a variável manipulada (MV) estejam estáveis, sofrem alteração e em seguida retornam novamente para o regime de estabilidade. Dessa forma, se o operador realizar uma mudança de Set-Point ou movimentar a válvula em manual, o AMCT irá avaliar os dados, desenvolver e validar o modelo. Para validar o modelo, os dados devem ser suaves, dentro do ruído normal de processo. A Figura 1 mostra os dez últimos modelos encontrados.

PID Tuning and Analysis for SJ-FIC_1301															Auto Scaled
Process Variables		AMCT Summary			Simulation and Analysis				Performance Evaluation						
AMCT Summary - SJ-FIC_1301															
Show AMCT models captured in the last 3 weeks															
	Start	P	I	D	F	Quality	Gain	DT	Lag1	Lag2	Intg	Stability	RRT	Notes	
<input type="checkbox"/>							%/%	sec	sec	sec			sec		
<input checked="" type="checkbox"/>	AMCT 03/10/2018 23:41	0,37	13	0	0	4	0,34	25,2	0	0	False	1,7	96		
<input checked="" type="checkbox"/>	AMCT 03/10/2018 19:58	0,39	50	0	0	2	0,37	27,3	0	0	False	6,1	100		
<input checked="" type="checkbox"/>	AMCT 03/10/2018 14:15	0,6	30	0	0	2	2,2	2	0	0	False	2,4	43		
<input checked="" type="checkbox"/>	AMCT 03/10/2018 13:14	0,13	3,6	0	0	3	1,4	4	0	0	False	1,3	16		
<input checked="" type="checkbox"/>	AMCT 03/10/2018 11:07	0,16	9,9	0	0	3	1,3	11	5,2	0	False	3	69		
<input checked="" type="checkbox"/>	AMCT 03/10/2018 10:44	0,043	39	0	0	2	3,5	70	0	0	False	43	200		
<input checked="" type="checkbox"/>	AMCT 02/10/2018 23:31	0,092	5,9	0	0	4	1,8	7	2,7	0	False	3,1	40		
<input checked="" type="checkbox"/>	AMCT 02/10/2018 21:51	0,43	6,3	0	0	1	0,58	6	3,8	0	False	0,7	43		
<input checked="" type="checkbox"/>	AMCT 02/10/2018 02:25	0,52	4,8	0	0	2	0,27	10	0	0	False	0,44	27		

Figura 1 - Modelos identificados pelo AMCT do PlantTriage.

A coluna "Quality" indica a qualidade do modelo que é um número entre 1 e 7. Sendo que 1 é uma qualidade excelente e 7 uma qualidade muito baixa. Para este estudo escolheu-se a penúltima linha por ser um modelo de qualidade excelente. Ao clicar neste modelo, é mostrada a Figura 2, onde temos a janela de tempo onde o AMCT encontrou uma oportunidade de detecção do modelo.

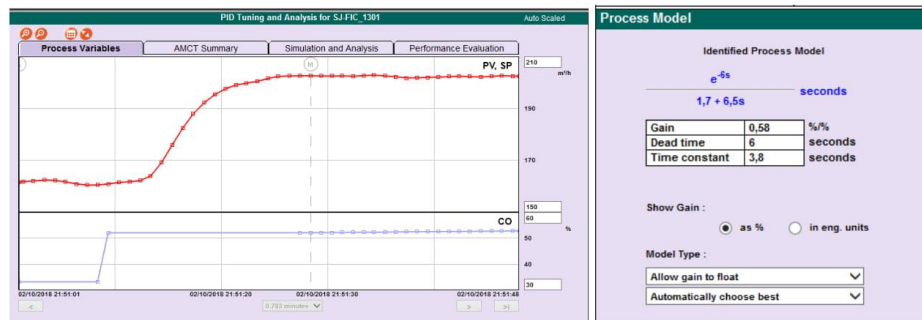


Figura 2 - Janela de tempo utilizada pelo AMCT para identificar o modelo de processo e modelo identificado.

Sendo assim, o modelo do processo, $P(s)$ é conhecido.

$$P(s) = \frac{0,58 \cdot e^{-6s}}{3,8s + 1} \quad (1)$$

Como o CLP trabalha de maneira digital, procedeu-se a modelagem matemática para determinar a função de transferência de dados amostrados, $G(z)$, considerando que o período de amostragem, T , seja 0,5s. Utilizou-se o software MATLAB para cálculo.

'MODELAGEM MATEMÁTICA'	%Exibe o título
T = 0.5;	%Define o período de amostragem.
numeradorS=0.58	%Define o numerador de G1(s)
denominadorS=[3.8 1]	%Define o denominador de G1(s)
'G1(s)'	%Exibe o título
G1=tf(numeradorS,denominadorS)	%Cria G1(s) e exibe



```
G =
    0.07151
    -----
    z - 0.8767

Sample time: 0.5 seconds
Discrete-time transfer function.
```

Considerando que o período de amostragem, T , seja 0,5s e por definição $z = e^{st}$, então $e^{-6s} = z^{-12}$. Dessa forma, o novo $G(z)$ é:

$$G'(z) = \frac{0,07151 \cdot z^{-12}}{z - 0,8767}$$

Segundo GARANITO, 2006, sistemas dinâmicos são todos os sistemas que evoluem no tempo, qualquer que seja a sua natureza, isto é, sistemas físicos, biológicos, químicos, sociais, econômicos, etc. Esta evolução pode ser descrita (modelada) por equações de diferenças, uma vez que esse tempo é muitas vezes medido em *intervalos* discretos.

Para simular o processo no CLP foi necessário encontrar a equação de diferença da função do processo vazão de embebição. Sendo a FT do processo dada por:

$$G(z) = \frac{0,07151}{z - 0,8767}$$

Para obter a equação de diferença:

$$G(z) = \frac{Y(z)}{F(z)} = \frac{0,07151}{z - 0,8767} = \frac{0,07151 \cdot z^{-1}}{1 - 0,8767 \cdot z^{-1}}$$

$$Y(z) \cdot [1 - 0,8767 \cdot z^{-1}] = F(z) \cdot [0,07151 \cdot z^{-1}]$$

$$Y(z) = 0,8767 \cdot z^{-1} \cdot Y(z) + 0,07151 \cdot z^{-1} \cdot F(z)$$

Na forma de equação de diferença:

$$y[n] = 0,8767 \cdot y[n-1] + 0,07151 \cdot f[n-1]$$

3. Resultados

Foi simulado o controlador fuzzy usando-se o Toolbox Simulink do software Matlab. Neste software, foi feito um posicionamento dos blocos de forma a simular e comparar a malha de realimentação negativa com controlador Fuzzy, malha com controlador PID e malha aberta. O diagrama de blocos está ilustrado na Figura 3.

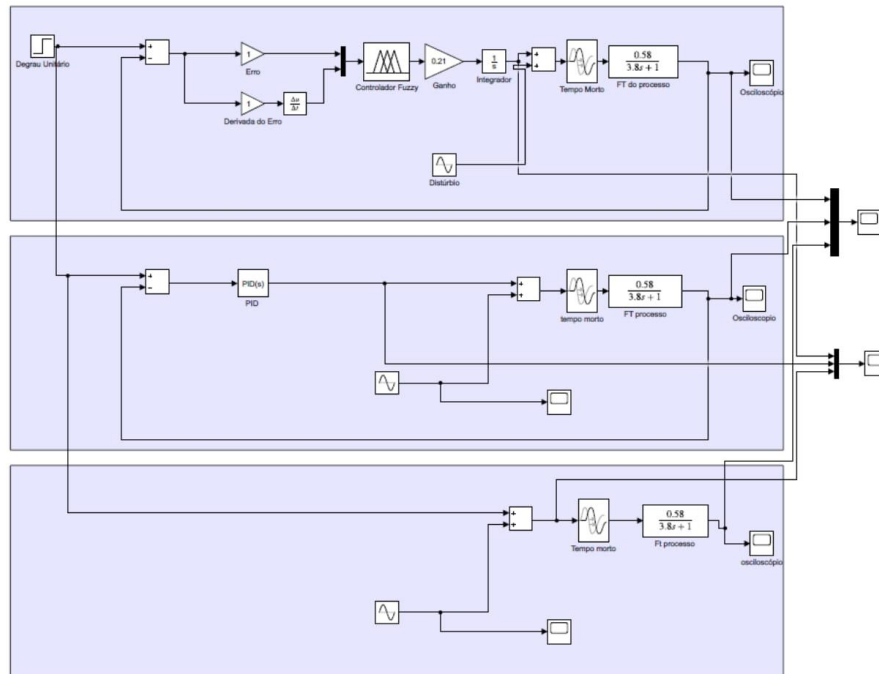


Figura 3 - Diagrama de bloco de uma planta de vazão de embebição controlada por um controlador Fuzzy, controlador PID e em malha aberta.

Observe que o ganho na malha do controlador Fuzzy tem o efeito de uma ação proporcional dependente. Alterando-se o seu valor, varia-se a velocidade de atuação do controlador fuzzy. Observa-se também que como a função de transferência da planta não possui um integrador intrínseco, há a necessidade de incluir um integrador em série na malha direta. A resposta da simulação está representada na Figura 4.

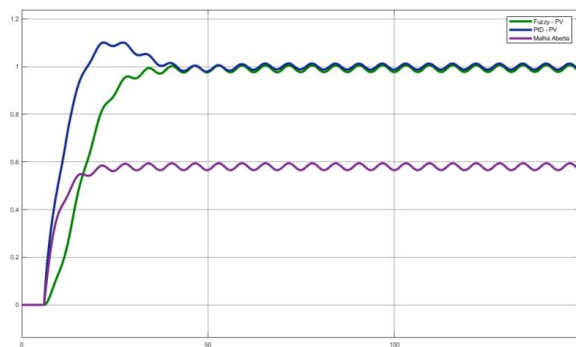


Figura 4 - Resposta da malha de controle.

A rotina do CLP com o controlador Fuzzy foi implementada em linguagem ladder. A

Figura 5 representa uma imagem principal da rotina.

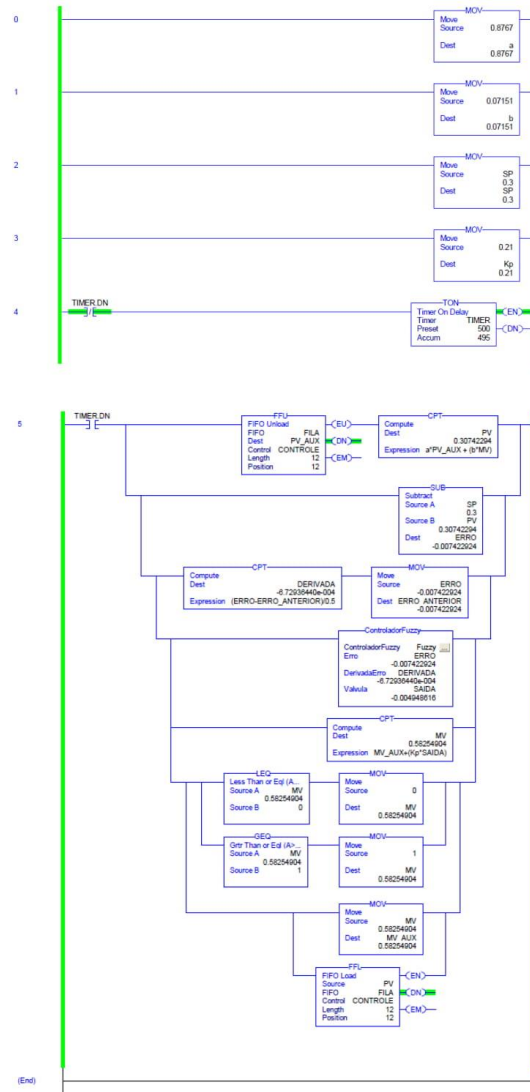


Figura 5 - Rotina no CLP.

A confecção de tela foi desenvolvida por meio do software Factory Talk View Studio® cuja aplicação é voltada à criação de telas gráficas de interação com controladores lógicos programáveis tendo como função proporcionar uma visão mais geral e intuitiva do processo. A tela pode ser vista na Figura 6.

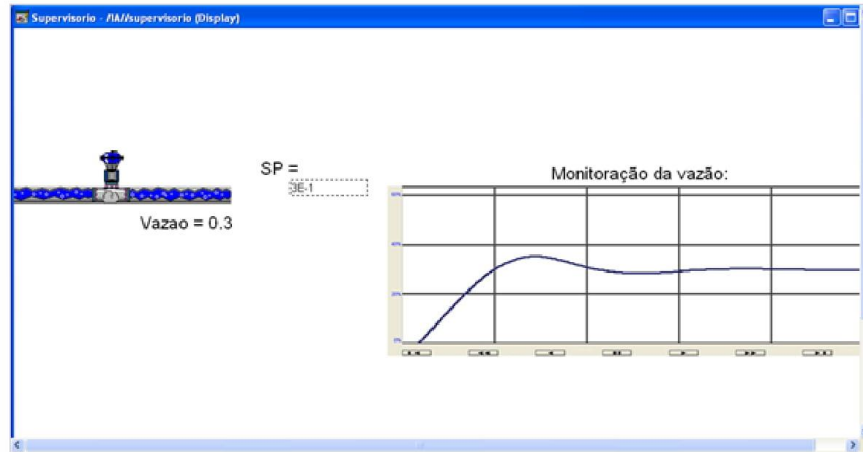


Figura 6 - Rotina implementada no CLP.

4. Conclusões

Com o término do trabalho foi possível desenvolver a proposta deste trabalho que era viabilizar a criação de um ambiente de simulação de controle inteligente de processos por meio da implementação de uma rotina de controle Fuzzy em linguagem ladder no software RSLogix 5000®. Também foi possível desenvolver um guia completo de como se utilizar as ferramentas da Rockwell Automation, desde a construção do controlador Fuzzy até os passos da própria integração do FuzzyDesigner® com o RSLogix 5000®, tornando tudo mais claro para qualquer usuário que já tenha tido contato com programação em ladder.

Foi possível integrar a rotina a um sistema supervisor, com simulação da planta do processo na própria ferramenta, tornando possível que o usuário teste e faça simulações mesmo sem ter acesso a plantas e sistemas físicos reais. Este sistema supervisor é capaz de se tornar uma importante ferramenta para análise e testes de estratégias de controle que poderiam ser impossíveis em um ambiente industrial. Verificou-se que o software FactoryTalk® foi satisfatório e com recursos suficientes sendo possível utilizar este software para tornar o sistema mais próximo de uma aplicação real, facilitando a operação.

Com base na teoria de sistemas dinâmicos lineares e na área de controle, a experiência revela que o controlador Fuzzy apresenta resultados melhores do que os obtidos com os controles convencionais, principalmente, quando se trata de um processo complexo ou com regiões não lineares. Este tipo de estratégia de controle pode ser também atraente para plantas onde o seu modelo matemático seja difícil de estimar ou plantas mais complexas e com alto grau de dificuldade para análise.

Observou-se neste trabalho a simplicidade do FuzzyDesigner® e que o projeto de um controlador Fuzzy é simples não exigindo grandes conhecimentos matemáticos do sistema, mas sim, conhecimento do funcionamento do sistema para criação da base de regras, ou seja, a maior dificuldade é conceitual de controle e não da utilização das ferramentas. Existe também a dificuldade de se encontrar referências para os primeiros passos na montagem de uma lógica Fuzzy.

Portanto, o trabalho desenvolvido permite que a simulação de processos possa ser desenvolvida em ambiente industrial por pessoal não especializado em técnicas de simulação. Desta forma, o técnico ou o engenheiro de controle poderá realizar experiências de simulação

antes de implantar as estratégias de controle no processo real.

5. Referências

GARANITO, Rafael Domingos. **Equações de diferenças e aplicações**. Dissertação de mestrado apresentada ao Departamento de Matemática e Engenharias da Universidade da Madeira. Agosto, 2006.

MANELLA, José Nazareno Anzanello. **Glossário de termos técnicos usados na indústria açucareira e alcooleira**. Maracá: Markgraf, 2012.

Utilização de ferramenta de control loop performance monitoring e Matlab para sintonia de malhas de controle

Neto L.R.P¹, Teixeira E.P²

¹ Universidade de Uberaba, Departamento de Engenharia Elétrica;

² Universidade de Uberaba, Departamento de Engenharia de Computação;

* Edilberto: edilberto.teixeira@gmail.com; Tel.: +55(34) 99977-3704

* Luciano: lpinheiro@tereos.com; Tel.: +55(34) 99995-1432

RESUMO: A competição global e as preocupações dos acionistas pressionam as plantas a superar suas metas tanto quanto reduzir custos, aumentar produção, melhorar qualidade, segurança e confiabilidade. Neste cenário, uma das formas de reduzir custos variáveis é melhorar o controle regulatório. Um dos fatores que influenciam a performance das malhas de controle é a sintonia inadequada pois as malhas sofrem deterioração no seu desempenho devido a alterações normais na dinâmica do processo. A utilização de softwares para simulação e sintonia de malhas de controle tem se destacado nas indústrias uma vez que permite aumentar a estabilidade da planta. Neste contexto têm-se as justificativas necessárias para se pesquisar e aplicar ferramentas de sintonia de malhas de controle nas indústrias. Este trabalho apresenta um estudo de caso de sintonia da malha de controle de embebição realizado em uma unidade industrial da Tereos Brasil. A hipótese defendida neste trabalho é avaliar se as ferramentas control loop performance monitoring e Simulink do Matlab® são eficazes para sintonia de malhas de controle. O resultado mostrou que as ferramentas utilizadas são práticas e eficazes, pois, permitiram a sintonia para parâmetros ótimos o que resultou na redução na variabilidade da vazão de embebição (67,8% menor) e também na atividade da válvula (79,5% menor).

Palavras-chave: Controle de processo, Sintonia de Malhas de Controle, Variabilidade de processos, Vazão de Embebição, Moendas.

1. Introdução

A competição global e a pressão dos acionistas fazem com que a indústria tenha que superar metas. Para permitir a continuidade da empresa no mercado que exige rápidas mudanças, este setor tem focado muito em melhoria contínua, em aumento da eficiência operacional, redução de custos, aumento da qualidade do seu produto final e investimento em projetos de otimização do seu processo industrial. Com toda essa dificuldade que o setor sucroenergético tem vivenciado, as linhas de processo de uma planta devem estar continuamente em melhoria, mantendo a sustentabilidade dos negócios.

O sistema de controle de processo automatiza ambos, rotinas e condições anormais de operação, e auxilia a operação a manter o controle de seu processo. O desempenho do processo, e, portanto, os resultados dos negócios aumentam através de operações aplicadas e conhecimento do processo, via sistemas de controle de processos. Infelizmente, muitos estudos nas indústrias têm mostrado que grande parte das malhas de controle não operam em modo normal, sofrem oscilação e aumentam a variabilidade do processo, reduzindo a eficiência da planta. O impacto direto no desempenho do processo é significativo, podendo citar: perda de produção, problemas de qualidade, consumo excessivo de energia, incidentes com segurança e meio ambiente e paradas não planejadas e alto custo de manutenção.

Segundo Fonseca (2004), uma malha de controle com desempenho ruim implica em perda de energia, qualidade e produção, já uma malha de controle com bom desempenho permite aumentar a qualidade e produtividade no processo. Uma das causas desse baixo desempenho é sintonia inadequada.

Objetiva-se neste estudo de caso otimizar o desempenho da extração do caldo-de-açúcar por meio da sintonia da malha de controle de vazão de embebição. Escolheu-se a malha de embebição que é uma malha de extrema importância no processo de extração da cana. Quanto mais caldo for extraído da cana mais se tem para produção e menor a perda no bagaço.

Foi utilizado uma ferramenta de control loop performance monitoring para auxiliar na identificação do modelo do processo e a toolbok Simulink do Matlab® para simular os parâmetros calculados. O foco em buscar aumento da eficiência industrial motivou a parceria empresa-universidade sem a qual este trabalho não seria possível. Os dados utilizados neste trabalho foram obtidos em uma usina pertencente ao grupo Tereos. A sintonia da malha foi realizada no sistema PCS7 da Siemens, com consentimento dos departamentos ligados à operação. O autor deste trabalho é colaborador no grupo e possui acesso aos dados de sistema de supervisão, laboratório e aos equipamentos industriais.

2. Controle de Vazão de Embebição

Uma importante operação unitária na usina é a extração do caldo da cana-de-açúcar que é realizado pelas moendas ou por difusores. Este processo consiste em separar o máximo possível de caldo da cana-de-açúcar. É importante que o controle desta área esteja otimizado para extrair o máximo possível de caldo o que significa aumento de produtividade para a unidade industrial. Na extração do caldo da cana-de-açúcar, um dos controles de processo é conhecido como controle de vazão de embebição que tem como objetivo medir e controlar a vazão de água utilizada para a lavagem e “amolecimento” da cana-de-açúcar.

Segundo Manella (2012), o processo no qual a água ou caldo é aplicado ao bagaço que sai dos ternos das moendas, a fim de aumentar a extração de sacarose de um conjunto de moendas. A embebição pode ser simples quando é aplicada somente água como agente de embebição. A embebição pode ser composta, quando a água é aplicada somente antes do último terno de moenda, sendo o caldo extraído pelo último terno aplicado como embebição, no antepenúltimo terno assim sucessivamente até a saída do bagaço do primeiro terno de moenda.

Submetendo o bagaço a pressões consideráveis e repetidas, nunca é possível extrair a totalidade de seu caldo. Visando extrair a maior quantidade possível do açúcar contido no bagaço, como não é possível diminuir esta umidade, procura-se substituir o caldo pela água. Este artifício é conhecido como embebição. Molhando o bagaço nas moendas, esta água se alastra no bagaço dilui o caldo nele contido. A moenda ficará com um bagaço com sua umidade limite. Porém, esta umidade não será mais constituída pelo caldo absoluto, mas pelo caldo diluído. Portanto, a quantidade de caldo ainda no bagaço continua a mesma, mas agora é constituída pelo caldo diluído em vez do caldo absoluto, assim extrai-se mais açúcar e esta operação é repetida.

Na unidade industrial da Tereos, onde realizou-se este estudo de caso, existe um sistema de RTO (Real-Time Optimization). Este sistema é baseado no modelo matemático da planta para verificar cenários próximos da operação atual e determinar qual a melhor condição de operação definindo um valor de set-point mais adequado para atingir o objetivo. Este sistema é capaz de prever o comportamento fluidodinâmico e termodinâmico do processo industrial e atuar para que o processo trabalhe no seu ponto ótimo, ou seja, baseado em equacionamento matemático, princípios de engenharia, dados e restrições de processo, este sistema determina a melhor condição de operação para o processo industrial no momento atual.

O controle de vazão de embebição pode ter um set-point local para manter uma vazão fixa ou ter um sistema de otimização em tempo real gerando set-point remoto. Uma das atuações do RTO, nesta unidade onde foi realizado o estudo de caso, é justamente definir o set-point remoto da malha de embebição que é foco deste trabalho. Para isso, ele calcula um balanço de massa e energia online e atua enviando set-point para a malha de controle de embebição com o objetivo da constância da embebição % fibra da usina e por consequência o aumento da extração da moenda. O RTO analisa a variação da fibra da cana processada, POL e umidade do bagaço, e seus impactos no controle da vazão de embebição. O ajuste da quantidade ideal de água garante a melhor extração da moenda, sem excesso de água, que afeta umidade do bagaço que está sendo queimado na caldeira.

Definido o set-point ótimo para esta malha por meio de um sistema de real-time-optimization, o objetivo deste estudo de caso é diminuir a variabilidade da malha de controle por meio da utilização dos softwares PlantTriage® e Matlab® para sintonia de malha de controle, ou seja, o RTO irá definir o ponto ótimo de atuação da malha que é o set-point ótimo e a sintonia da malha irá reduzir a variabilidade da variável de processo. A redução da variabilidade se reflete em aumento da extração do caldo-da-cana e consequente aumento de produção além de redução de consumo de vapor nas válvulas redutoras e sobra direta de bagaço para cogeração em períodos que a planta não está operando, como paradas programadas, paradas por chuva e entressafra.

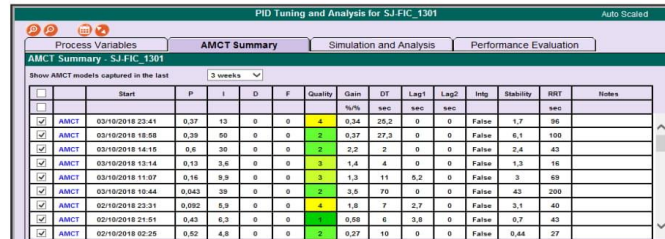
3. Materiais e Métodos

Segundo Farenzena (2008), devido ao grande número de controladores em uma planta industrial, avaliar o desempenho de todos os controladores manualmente em tempo real é impossível. Isto explica o interesse crescente no meio acadêmico e na indústria, em metodologias automáticas para avaliar desempenho das malhas de controle.

Utilizou-se neste trabalho a ferramenta PlantTriage® que é uma ferramenta de Control Loop Performance Monitoring (CLPM) desenvolvido pela empresa norte-americana ExpertTune. Este software opera monitorando continuamente e avaliando as malhas de controle e informa quando o desempenho começa a se deteriorar, disponibilizando indicadores de desempenho das malhas de controle.

Detectadas as malhas que estão com baixa performance, utilizou-se para identificação do modelo do processo a funcionalidade AMCT (Automatic Model Capture Technology) do software PlantTriage®. O AMCT funciona em real-time procurando e reportando modelos de processos por meio da avaliação automática de todos os testes de resposta que ocorrem na malha de controle, tais como: mudanças de set point, rampas ou movimentos de válvula no modo manual. Ele filtra os dados e procura locais em que a variável de processo (PV) e a variável manipulada (MV) estejam estáveis, sofrem alteração e em seguida retornam novamente para o regime de estabilidade. Dessa forma, se o operador realizar uma mudança de Set-Point ou movimentar a válvula em manual, o AMCT irá avaliar os dados, desenvolver e validar o modelo. Para validar o modelo, os dados devem ser suaves, dentro do ruído normal de processo. Uma sobrecarga de carga não se qualifica como fornecendo dados bons para modelagem.

Conforme mostrado na Figura 1, todos os modelos encontrados pelo AMCT, sendo que ele monitora continuamente um local em que pode-se encontrar um bom modelo de processo. Nesta figura, é mostrado os dez últimos modelos encontrados.



Start	P	I	D	F	Quality	Gain	DT	Lag1	Lag2	Intg	Stability	BRT	Notes
						%%	sec	sec	sec			sec	
AMCT 03/10/2018 23:41	0,37	13	0	0	4	0,34	25,2	0	0	False	1,7	95	
AMCT 03/10/2018 18:58	0,39	00	0	0	2	0,37	27,3	0	0	False	5,1	100	
AMCT 03/10/2018 14:15	0,6	30	0	0	2	2,2	2	0	0	False	2,4	43	
AMCT 03/10/2018 13:14	0,13	3,6	0	0	3	1,4	4	0	0	False	1,3	16	
AMCT 03/10/2018 11:07	0,16	0,8	0	0	3	1,3	11	0,2	0	False	3	69	
AMCT 03/10/2018 10:44	0,043	39	0	0	2	3,5	70	9	0	False	43	300	
AMCT 02/10/2018 23:31	0,092	5,9	0	0	4	1,8	7	2,7	0	False	3,1	40	
AMCT 02/10/2018 21:51	0,43	6,3	0	0	5	0,58	6	3,8	0	False	0,7	43	
AMCT 02/10/2018 02:25	0,52	4,8	0	0	2	0,27	10	0	0	False	0,44	27	

Figura 1 - Modelos identificados pelo AMCT do PlantTriage.

A coluna “Quality” indica a qualidade do modelo que é um número entre 1 e 7. Sendo que 1 é uma qualidade excelente e 7 uma qualidade muito baixa. Para este estudo escolheu-se a penúltima linha por ser um modelo de qualidade excelente. Ao clicar neste modelo, é mostrada a Figura 2, onde temos a janela de tempo onde o AMCT encontrou uma oportunidade de detecção do modelo.

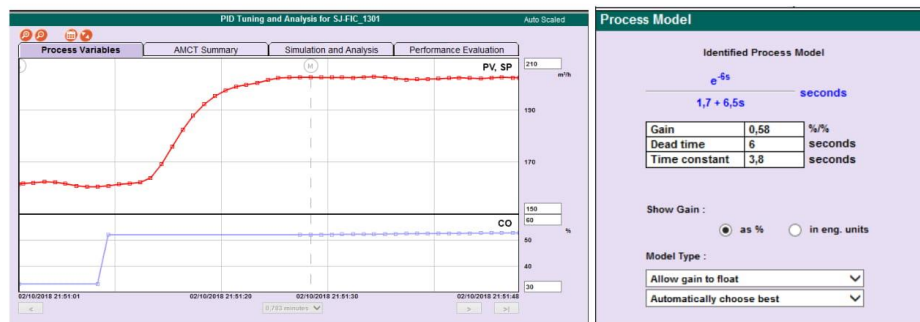


Figura 2 - Janela de tempo utilizada pelo AMCT para identificar o modelo de processo e modelo identificado.

Sendo assim, o modelo do processo, $P(s)$ é conhecido.

$$P(s) = \frac{0,58 \cdot e^{-6s}}{3,8s + 1} \quad (1)$$

Conhecido o modelo do processo, procedeu-se ao cálculo dos parâmetros de sintonia. Na literatura existem vários métodos de sintonia de controladores PID. Neste trabalho, adotou-se para sintonia do controlador o método de síntese direta. Se o modelo do processo, $P(s)$, for conhecido e τ_c determinado logo é possível calcular $C(s)$. Como a função do processo é conhecida, e determinando $\tau_c = 3s$, ou seja, a malha irá estabilizar em 50 unidades de tempo: $(5 \times \tau_c) + \theta = (5 \times 3) + 6 = 21u.t.$ Sendo que, $\frac{\tau_c}{\theta} = \frac{3,82}{6} < 2$ então o controlador PID funcionará bem. Dessa forma, procedeu-se ao projeto do controlador por meio de síntese direta.

$$C(s) = \frac{e^{-\theta \cdot s}}{P(s) \cdot (\tau_c + \theta_c) \cdot s} = \frac{e^{-6s}}{\left(\frac{0,58 \cdot e^{-6s}}{3,8s + 1}\right) \cdot (3+6) \cdot s} = \frac{3,8s + 1}{5,22s} = 0,727 + \frac{1}{5,22s}$$

Portanto, temos que $K_p = 0,727$ e $T_i = 5,22s$. Dando sequência, procedeu-se a simulação do controlador PID utilizando-se a toolbox Simulink do software Matlab®. O diagrama de blocos está ilustrado na Figura 3.

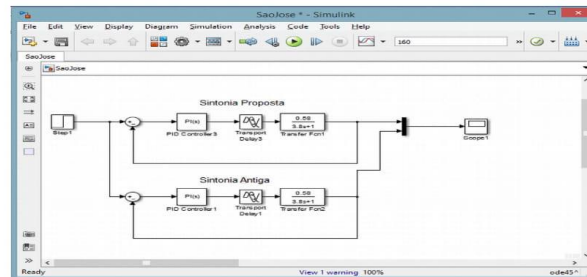


Figura 3 - Simulação do controlador PID.

Para realizar o acompanhamento da variável de processo, variável manipulada e setpoint da malha de controle de vazão de embebição, foi utilizado o sistema PI® (Plant Information) da OSIsoft. Trata-se de um PIMS (Plant Information Management System), utilizado para gestão de informação da planta, que coleta dados de processo residentes em fontes distintas e faz o armazenamento em um banco de dados único.

4. Resultados

A simulação no Simulink descrita no tópico anterior, mostra que para um set-point unitário, o desempenho da malha foi melhor com os parâmetros de sintonia calculados (linha azul). O resultado da simulação está representado pela Figura 4.

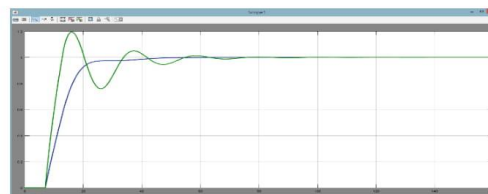


Figura 4 - Resposta da malha de controle.

Após ajustar o controlador PID com os parâmetros da simulação, os resultados foram satisfatórios. Conforme demonstrado na Figura 5, a variabilidade da vazão de embebição foi menor no período pós sintonia e a válvula atuou menos aumentando a vida útil da mesma.

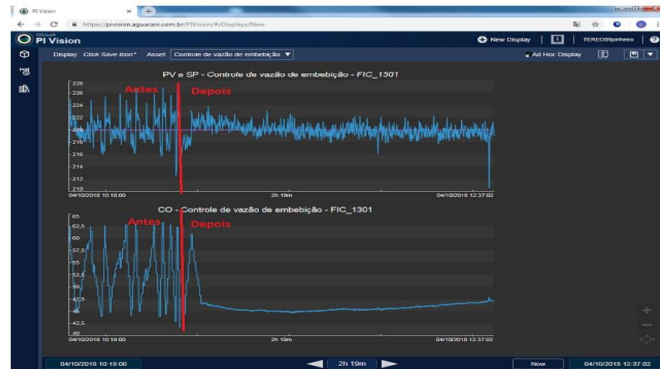


Figura 5 - Comparação antes e depois da sintonia.

Analisando-se Tabela 1, no período antes da sintonia, o valor médio da PV foi de 219,84, com desvio padrão de 2,49. No período após a sintonia o valor médio da PV foi de 219,87 com desvio padrão de 0,8, ou seja, o período com sintonia apresentou uma dispersão menor dos dados em relação à média e a amostra é mais homogênea. Portanto, a sintonia contribuiu para diminuir a variação da vazão de embebição em 67,8%. Ao observar os valores da MV, o ajuste do controlador PID também obteve resultados satisfatórios. No período antes da sintonia, o valor médio da MV foi de 2,89, com desvio padrão de 4,43. No período após a sintonia o valor médio da MV foi de 45,78 com desvio padrão de 0,907, ou seja, o período com sintonia apresentou uma dispersão menor dos dados em relação à média e a amostra é mais homogênea com uma redução de 79,5% no desvio padrão. Portanto, a sintonia contribuiu deixar a válvula atuando menos e com isso aumentar a vida útil da mesma.

Tabela 1 - Dados antes e após a sintonia.

	Variável de Processo		Variável Manipulada	
	Antes da sintonia	Após a sintonia	Antes da sintonia	Após a sintonia
Média	219,8461778	219,8750688	52,89488693	45,78655885
Erro padrão	0,379721933	0,081283321	0,684157823	0,091706697
Mediana	219,5274107	219,8272957	52,42902374	45,48290634
Desvio padrão	2,490003232	0,800547867	4,43384945	0,907849985
Variação de amostra	6,200116095	0,640876888	19,65902095	0,824191596
Curtose	1,284859414	3,008259464	-0,914713768	18,99996701
Range	13,28841284	5,595265214	16,11893463	7,029327393
Mínimo	212,5163361	216,4273146	45,64484787	44,78342056
Máximo	225,8047489	222,0225798	61,7637825	51,81274796

6. Conclusões

Verificou-se com os resultados obtidos no estudo de caso de sintonia de malha de controle da Tereos que as ferramentas AMCT do PlantTriage® e Simulink do Matlab® são eficazes para

sintonia das malhas de controle pois foi possível comprovar melhoria de performance com a redução na variabilidade. Neste estudo de caso utilizou-se a malha de controle de vazão de embebição para avaliação do seu desempenho. Com o auxílio do AMCT identificou-se o modelo do processo e realizou-se o cálculo dos novos parâmetros do controlador. Como resultado obteve-se uma melhoria considerável na oscilação do controle. Além da inspeção visual, os ganhos puderam ser observados por meio da redução do desvio padrão da variável de processo e pela estabilização do elemento final de controle.

Sendo assim, a hipótese defendida neste trabalho é verdadeira, pois, com a utilização do PlantTriage® e do Matlab® foi possível identificar o modelo, analisar e simular a performance da malha de controle, ou seja, a eficácia dos parâmetros calculados foi comprovada com os resultados obtidos. Na malha de controle de vazão de embebição os principais resultados foram com a redução na variabilidade da vazão (67,8% menor) e também na atividade da válvula (79,5% menor). Esta redução de vazão de embebição se reflete em redução de consumo de vapor nas válvulas redutoras e sobra direta de bagaço para cogeração em períodos que a planta não está operando, como paradas programadas, paradas por chuva e entressafra.

6. Referências

FARENZENA, M. **Novel methodologies for assessment and diagnostics in control loop management**. 2008. Tese de Doutorado – Faculdade de Engenharia Química, Universidade Federal do Rio Grande do Sul – UFRGS, Porto Alegre, RS.

FONSECA, M. O.; SILVA, C.; TORRES, B. S. Avaliação de desempenho e auditoria de malhas de controle. **Revista InTech**, n. 63, p. 32-37, 2004.

MANELLA, José Nazareno Anzanello. **Glossário de termos técnicos usados na indústria açucareira e alcooleira**. Maracá: Markgraf, 2012.

Apêndice B – Passo-a-passo para criação da rotina *ladder* de simulação da planta

Para criação e configuração do controlador SoftLogix5860 utilizou-se o *software* SoftLogix Chassis Monitor®. Esse procedimento foi realizado selecionando a opção *Slot* → *Create Module* conforme ilustrado na Figura 64.

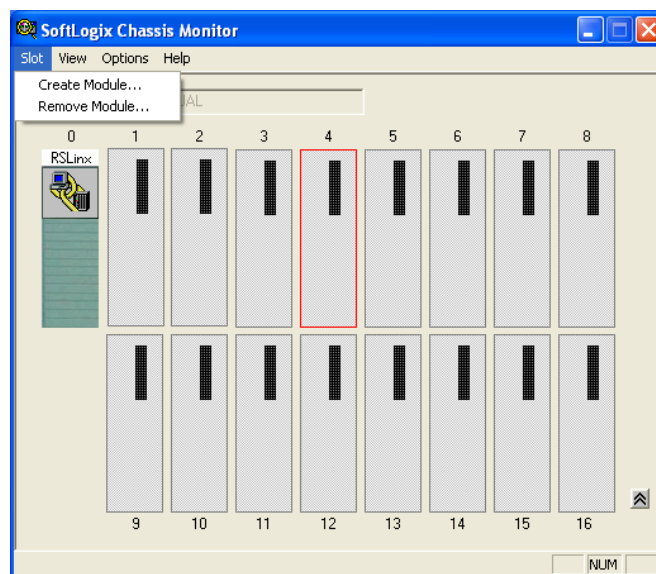


Figura 64 - Criação e configuração do controlador SoftLogix 5860.

Selecionou-se o modelo do controlador e escolheu-se o slot 1 dos 16 disponíveis no *software* SoftLogix Chassis Monitor®. Foi adotado o controlador SoftLogix5860 conforme na Figura 65.

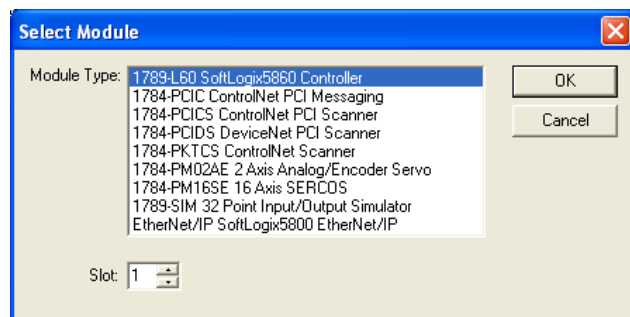


Figura 65 - Seleção do tipo de controlador a ser criado.

Dando sequencia, escolheu-se o modo de operação do controlador no *startup* conforme mostrado na Figura 66.

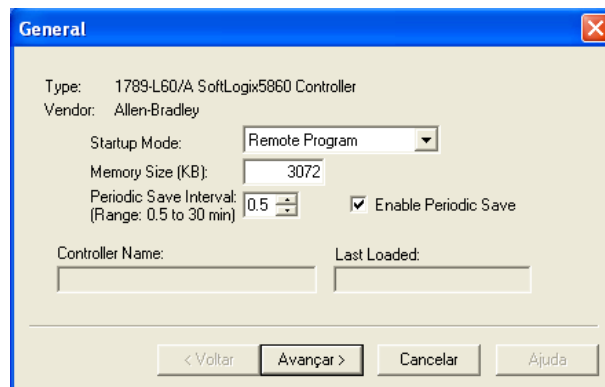


Figura 66 - Tela de configurações gerais do controlador virtual.

Escolheu-se o tamanho de memória e configurações gerais conforme mostrado na Figura 67.

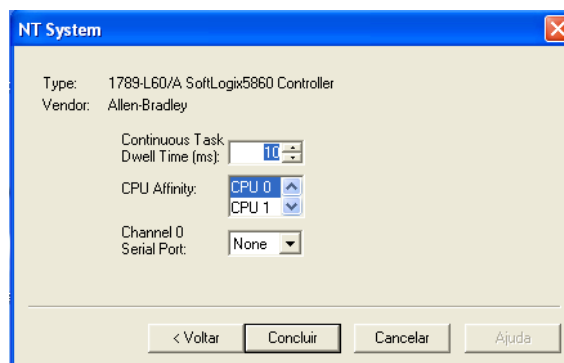


Figura 67 - Tela de configurações gerais do controlador virtual.

Por fim, é concluída a configuração para criação do controlador. A partir de agora o mesmo aparece na tela principal do *software* conforme visto na Figura 68.

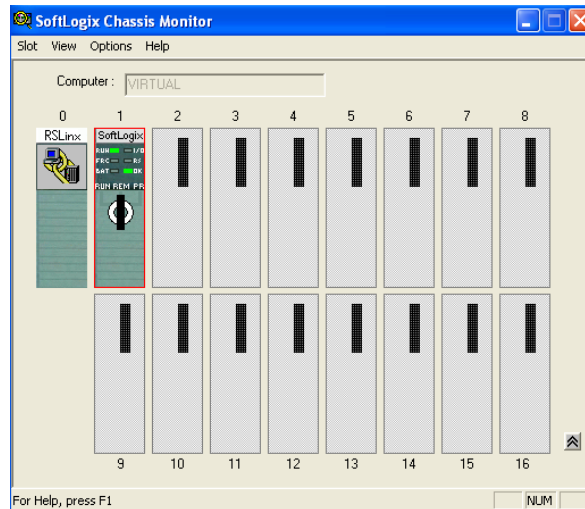


Figura 68 - Seleção dos modos de operação do SoftLogix 5860.

Ao criar o controlador virtual SoftLogix 5860 o próximo procedimento é implementar a rotina para carregá-la no CLP.

O ambiente de desenvolvimento utilizado é o RSLogix5000®. Para programar uma rotina, as seguintes etapas devem ser executadas: abertura da rotina; inserção das instruções *ladder*; inserção das instruções do bloco de função; atribuição de operandos e verificação da rotina (Rockwell Automation, 2018).

Inicialmente criou-se um projeto no RSLogix 5000® selecionando a opção *File*→*New*. Preencheram-se as informações necessárias para criação conforme na Figura 69.

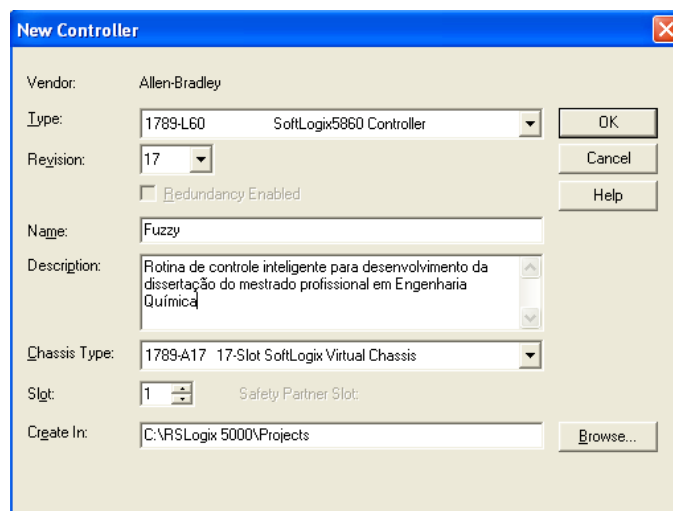


Figura 69 - Criação do projeto no RSLogix 5000®.

Após criar o projeto aparece a tela conforme apresentado na Figura 70.

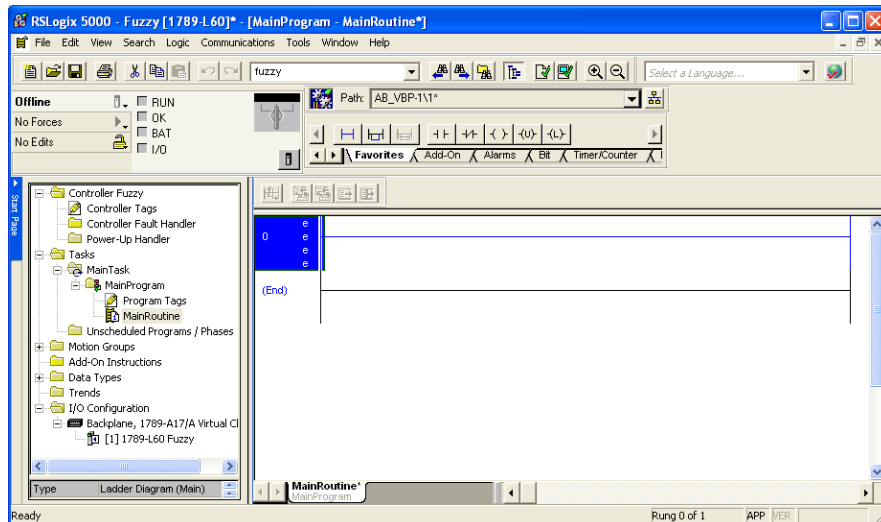


Figura 70 - Tela principal do *software RSLogix 5000*[®].

No subdiretório *MainProgram* há dois campos: *ProgramTags* e *MainRoutine*. A opção *MainRoutine* é utilizado para abrir a edição de programação em diagrama *ladder* conforme representado na Figura 70.

Utilizou-se o *ProgramTags* para facilitar a programação referenciando os dados na memória do CLP ao invés de endereços, ou seja, os dados são armazenados em *tags* ao invés de ter que referenciar o endereço numericamente. Outra vantagem é que a sua utilização melhora a organização dos dados e do código em *ladder*. A Figura 71 mostra a tela de monitoração de *tags* e seus tipos.

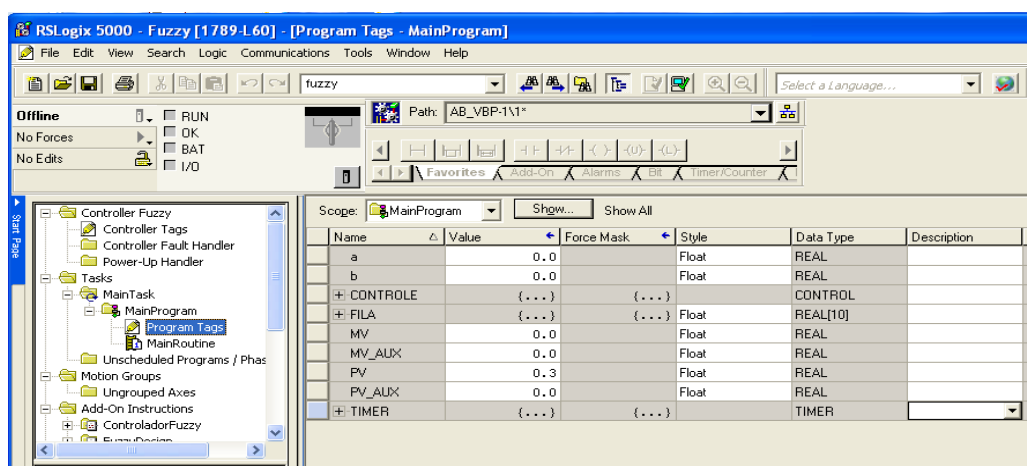


Figura 71 - Tela de monitoração de *tags* e seus tipos.

A variável PV é a variável de entrada do CLP que simulará o sinal externo pertencente ao processo como, por exemplo, o sinal advindo do transmissor de vazão de embebição que será recebido pelo controlador. O sinal lido pelo sensor ou transmissor de vazão informará ao

controlador de vazão para compará-lo com o valor de *set-point* e atuar na válvula de injeção de fluido. A variável MV é a variável de saída que representará o sinal enviado pelo controlador ao elemento final de controle para ter intervenção no processo. Os valores de PV e MV serão transmitidos a um sistema supervisorio em seção posterior deste trabalho. As variáveis foram declaradas do tipo *float*. A variável “TIMER” e “FILA” foram declaradas para definir, respectivamente, o nome do temporizador e da fila do tipo *FIFO*. A tag FILA contém 12 elementos do tipo de dados real. A variável CONTROLE do tipo *control* foi declarado para definir o endereço de arquivo de controle usado para armazenar os *bits* de status, a duração da pilha e o valor da posição para as instruções de deslocamento.

Após declarar as variáveis, procedeu-se ao desenvolvimento da rotina *ladder*. A programação se dá inserindo elementos controlados (bobina de relé) e um conjunto de condições para o controle desse elemento (lógica de contatos) nas linhas horizontais. Cada contato pode assumir dois estados (energizado ou desenergizado) representando assim uma variável booleana, ou seja, uma variável que assume os valores de verdadeiro ou falso (Soprani, 2006).

Ao inserir o temporizador no diagrama *ladder*, clicou-se com o botão direito do mouse escolhendo a opção "New Tag" para definir o nome do temporizador. Escolheu-se o valor base para o tipo de endereçamento e "TIMER" para o tipo de dado conforme representado na Figura 72. A variável “TIMER” foi declarada anteriormente. O mesmo procedimento foi realizado para vincular as tags FILA e CONTROLE aos blocos FFU e FFL.

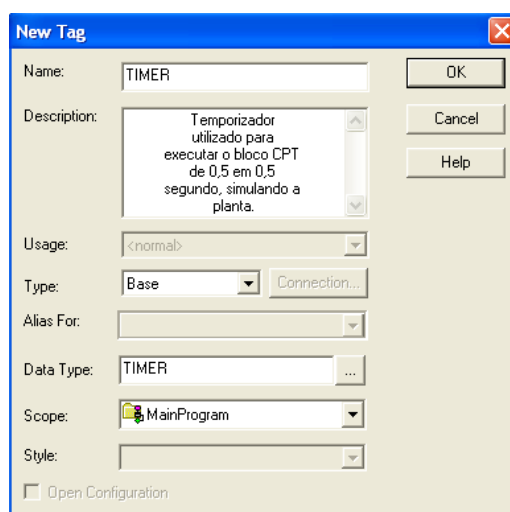


Figura 72 - Tela para vincular uma tag ao temporizador.

Finalizando a implementação da rotina de simulação do processo de vazão de embebição foi realizada a verificação da lógica da rotina a fim de verificar erros de programação. Este procedimento foi realizado selecionando a opção *Logic* → *Verify* → *Routine*. Conforme mostrado na Figura 73 não foram encontrados erros na rotina de programação.

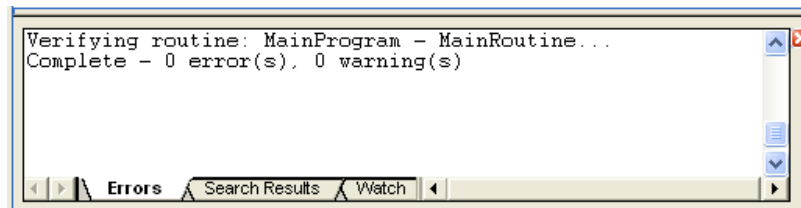


Figura 73 - Janela de visualização de erros.

Após programar e verificar a rotina é necessário enviá-la para o controlador. O *download* do programa é feito selecionando a opção *Who Active* conforme na Figura 74.

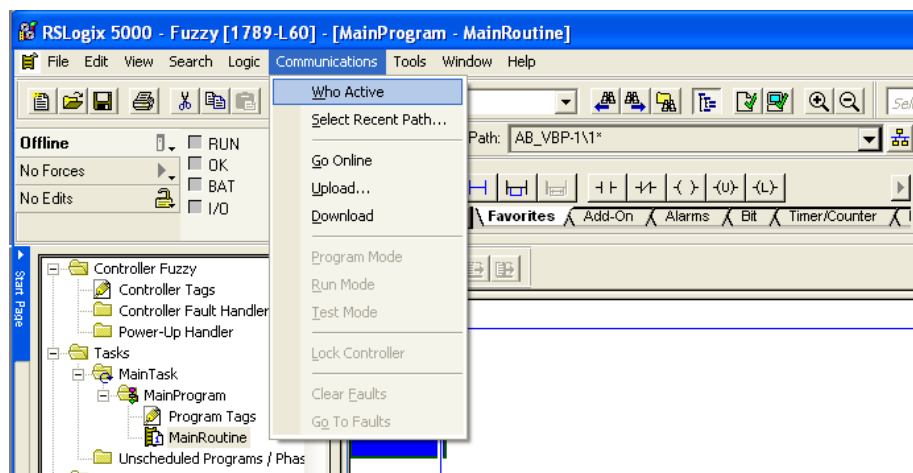


Figura 74 - Download do programa no controlador.

Na opção *Who Active*, deve-se selecionar o controlador ao qual se deseja enviar o programa e clicar em *download* conforme mostra a Figura 75.

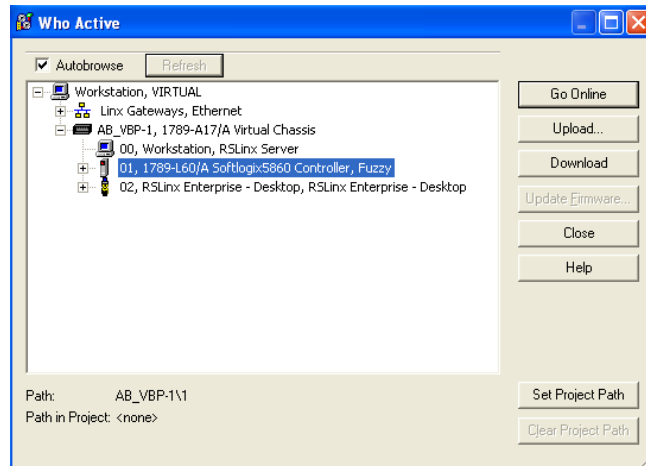


Figura 75 - Procedimento para enviar o programa ao controlador.

Ao clicar em *Download* irá aparecer um aviso padrão de segurança solicitando confirmação da carga do programa conforme mostrado na Figura 76.

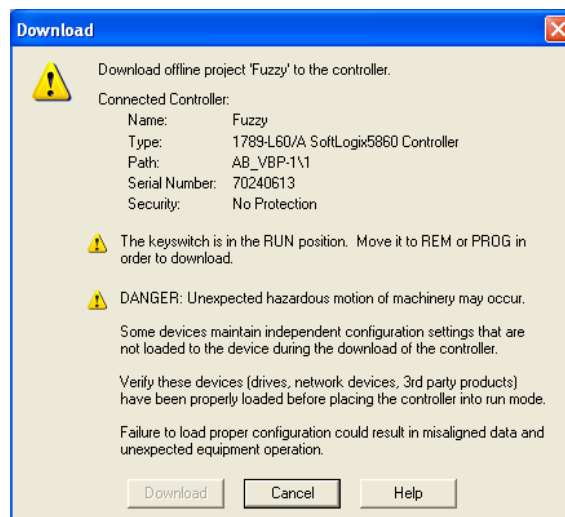


Figura 76 - Aviso de confirmação de *download* de programa para o CLP.

Os controladores da família Logix da Rockwell possui uma chave no seu painel onde é possível selecionar três posições: *Run*, *Program* ou *Remote*. O modo de operação *Run* executa a rotina *ladder* monitorando as entradas e atuando na saída. Esse modo permite a edição da rotina *on-line*. O modo de operação *Program* não executa a rotina, desenergiza as saídas e permite editar a rotina *on-line*. O modo *Remote* também permite a edição da rotina *on-line* evitando perda dos estados das saídas evitando um novo *startup*.

Após o *download* da rotina, se o CLP estiver no modo *Remote Run* haverá mudança para o modo *Program* suspendendo a execução da rotina carregada anteriormente e substituindo

pela nova rotina. Depois de carregado, é necessário alterar o modo de operação mudando a posição da chave seletora do controlador por meio do *software* SoftLogix Chassis Monitor[®] conforme ilustrado na Figura 77.

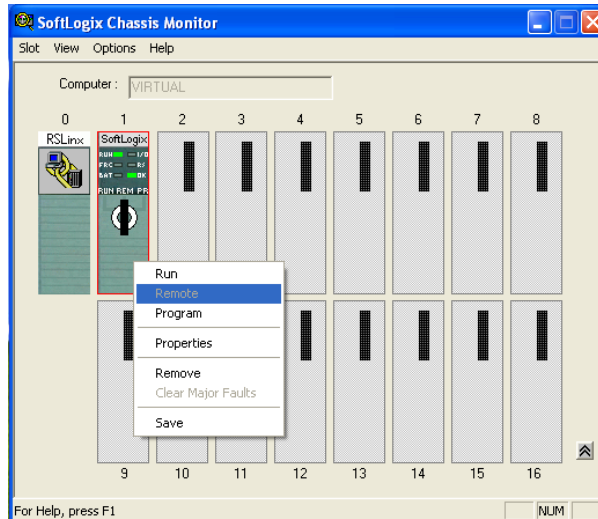


Figura 77 - Mudança do modo de operação por meio do *software* SoftLogix Chassis Monitor[®].

Apêndice C – Passo-a-passo para criação do Controlador *Fuzzy* utilizando-se o *software* Matlab

No *software* Matlab[®] utilizou-se uma interface para a simulação do controlador *Fuzzy* e sistemas nebulosos como um todo. Esta é denominada FIS (do inglês, *Fuzzy Inference Systems*) e é acionada pelo comando “*fuzzy*” no *workspace*. A tela inicial da *toolbox* está ilustrada na Figura 78.

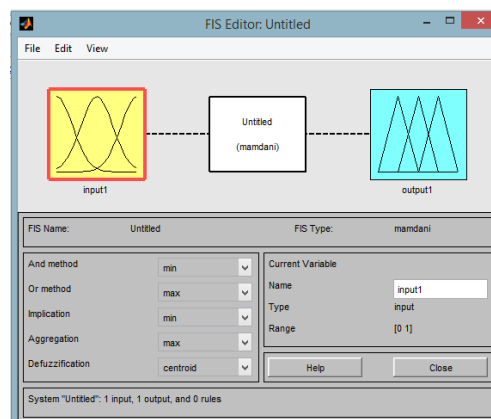


Figura 78 - Tela inicial da *toolbox* *Fuzzy* do *software* Matlab[®].

A proposta adotada para o controlador neste trabalho utiliza os valores do erro de controle e a variação do erro de controle como entradas para o controlador *Fuzzy*. Então, inseriu-se mais uma entrada clicando-se em *Edit* → *Add input*. Alteraram-se os nomes das entradas e da saída para sua respectiva variável linguística. A tela ficou configurada conforme representado na Figura 79.

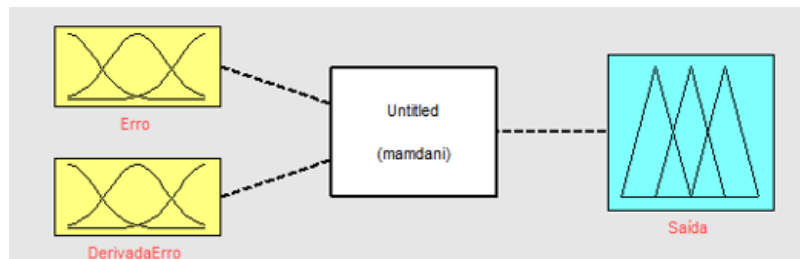


Figura 79 - Tela com o controlador com duas entradas.

Em seguida, adicionaram-se as funções de pertinência clicando-se na variável de entrada *Erro* → *View* → *Edit membership functions*. Em seguida clicou-se em *Edit* → *Add membership functions* e escolheu-se o tipo de função de pertinência *trimf* que corresponde à função triangular conforme Figura 80.

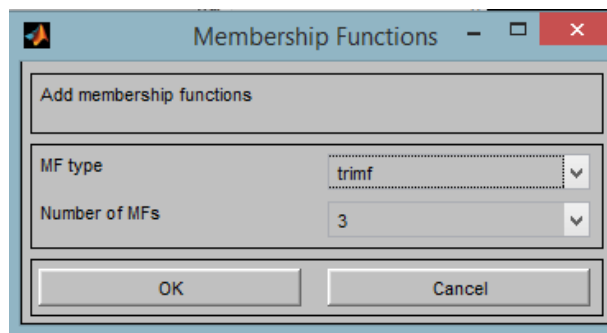


Figura 80 - Configurando o tipo de função de pertinência.

Selecionado o tipo de função de pertinência irá aparecer a função que pode ser modificada graficamente clicando-se nos pontos indicados nas linhas e arrastando-os. Dessa forma, clicou-se sobre a curva para configurar a faixa para o universo de discurso e também o nome dos predicados da variável linguística. Repetiu-se este procedimento para a outra entrada, e para a saída que corresponde a válvula de atuação na planta. A Figura 81 apresenta a tela com as funções já configuradas.

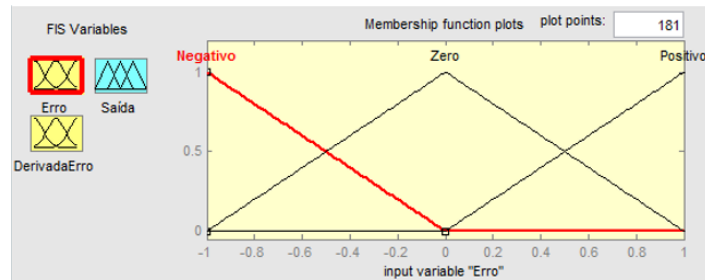


Figura 81 - Configuração das funções de pertinência.

O próximo passo é editar a base de regras. Para editá-las clicou-se em *Edit*→*Rules* e inseriram-se as nove regras. A tela com todas as regras inseridas está ilustrada na Figura 82.

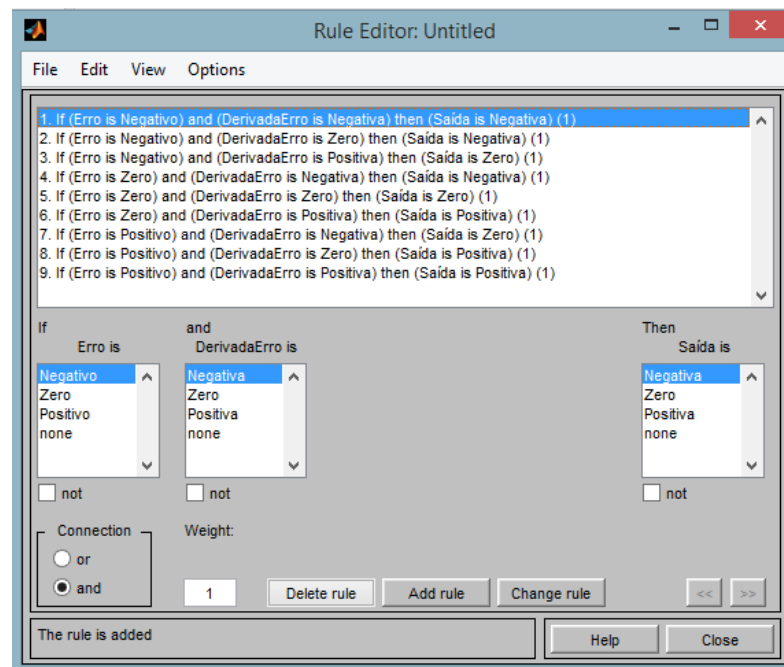


Figura 82 - Inserindo a base de regras.

Percebe-se a presença das nove regras que foram estudadas. As variáveis de entrada do controlador e a saída são relacionadas por meio do conectivo E e ENTÃO (em inglês, “*if*” e “*then*”). Clicando em qualquer uma dessas regras é possível editá-las, assim como excluir ou criar novas.

Em *View*→*View Rules*, é possível ter acesso a uma simulação em tempo real sobre o que está ocorrendo no sistema conforme ilustrado na Figura 83. Nesta figura o critério MIN-MAX está ilustrado graficamente. É possível observar que estão desenhadas todas as funções de pertinência. Os valores mínimos assumidos pelas variáveis de entrada são comparados com a função de saída, isto é, com a função de pertinência correspondente a saída do controlador.

Esses valores estão indicados graficamente, nos gráficos à direita da Figura 83. Finalmente, determina-se uma função que corresponde ao valor máximo entre essas três funções. Obtém-se então o gráfico mostrado embaixo e à direita da Figura 83. A obtenção do resultado final da válvula é realizada por meio de um processo denominado defuzificação, onde se obtém um valor numérico da grandeza de engenharia a partir da função de pertinência resultante. Há vários métodos de defuzificação, dentre os quais se citam:

- Método do centro de área: obtém-se o valor resultante tomando-se o valor do universo de discurso correspondente ao centro geométrico da área sob a curva.
- Método da média ponderada: obtém-se o valor resultante tomando-se a média ponderada entre os valores de pertinência e os valores do universo de discurso correspondente.
- Método da média dos máximos: nesse caso, tomam-se os valores máximos da curva resultante e obtém-se a média dos universos de discurso correspondentes.

Para analisar o sistema selecionou-se um erro igual a -0.422 e um valor de -0.282 para a derivada do erro. Por meio do método da média ponderada, obteve-se o valor resultante para a saída do controlador igual a -0.0883. Observa-se que o controlador vai decrementar o sinal enviado a válvula, ou seja, tende a se fechar. Isto está de acordo com a regra *fuzzy* 1.

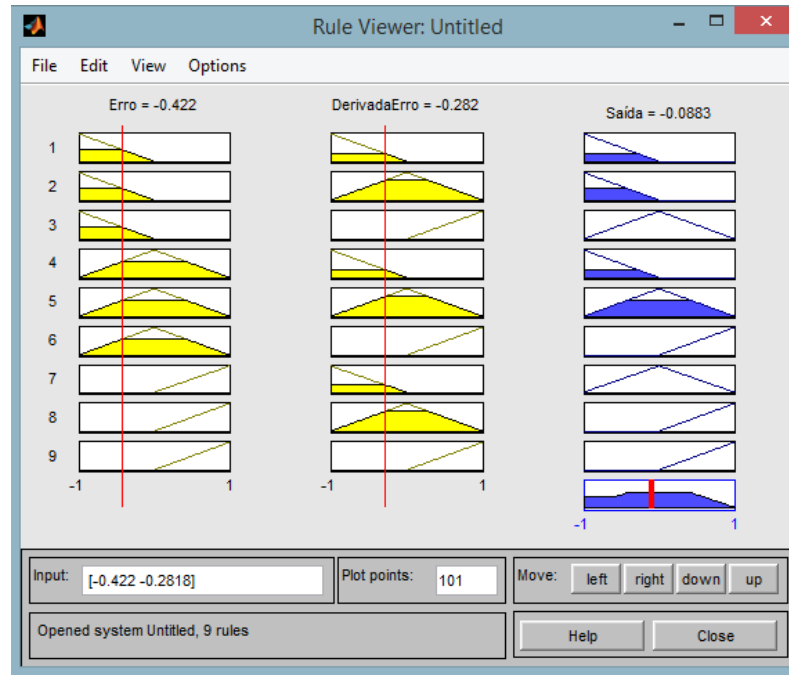


Figura 83 - Visualização da Aplicação do critério MIN-MAX na aplicação das regras.

A título de exemplificação, na Figura 84 percebe-se que o erro e a sua derivada são nulos, logo, a válvula recebe um valor praticamente nulo para manter-se a estabilidade do sistema. Isto está de acordo com a regra número 5 elaborada na seção anterior.

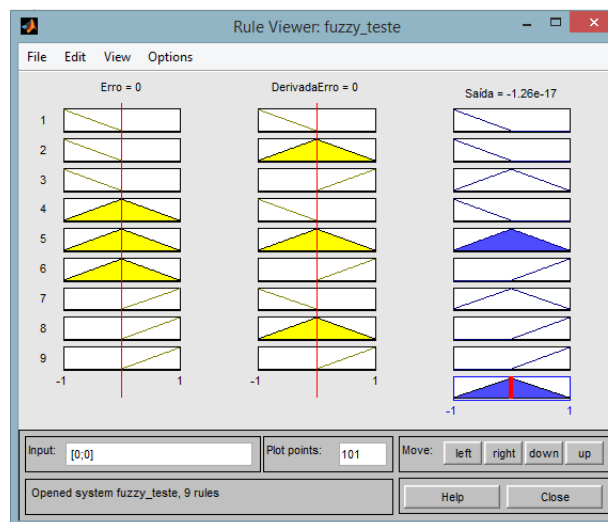


Figura 84 - Simulação em tempo real do FIS. Fonte: Autor

Pode-se também visualizar a aplicação das regras em um gráfico tridimensional clicando-se em *View* → *Surface*. Este gráfico apresenta dois eixos na horizontal com as entradas e um eixo na vertical com a saída do controlador conforme ilustrado na Figura 85.

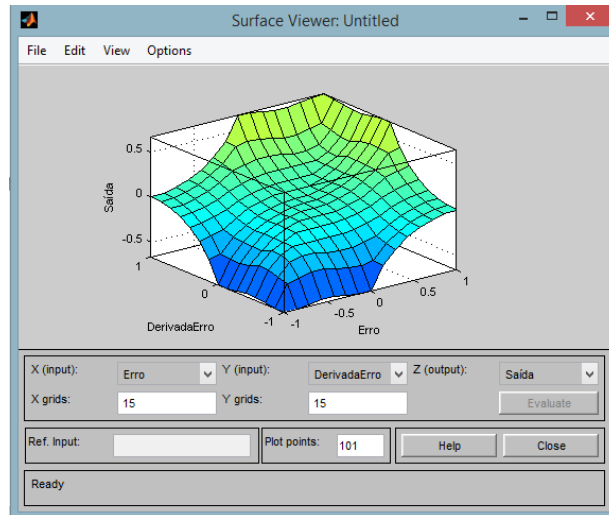


Figura 85 - Superfície que ilustra a atuação das regras fuzzy.

Em seguida, exportou-se o controlador criado na *toolbox Fuzzy* para o *workspace*. Com isso, será possível utilizá-lo na *toolbox Simulink*, também do *software Matlab®*, para simular a atuação do controlador em malha fechada com a função da planta simulada. Para exportar foi necessário clicar em *File* → *Export* → *To Workspace* conforme na Figura 86.

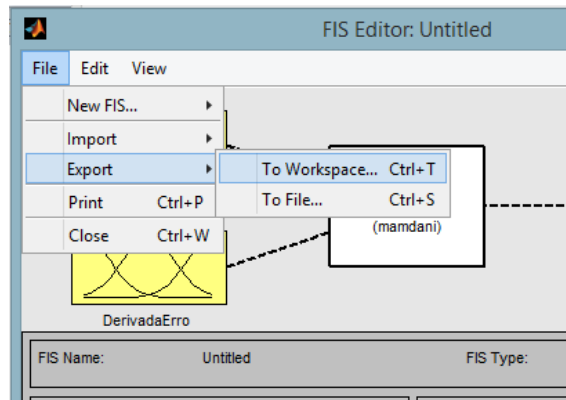


Figura 86 - Exportar o controlador Fuzzy para o Workspace do Matlab®.

Com este procedimento realizado, gerou-se um arquivo de extensão *.fis* e este arquivo contém a base de regras e é vinculado ao controlador *Fuzzy* no *Simulink*.

Apêndice D – Passo-a-passo para criação do Controlador *Fuzzy* utilizando-se o *software* FuzzyDesigner e posterior importação na rotina no RSLogix

A tela inicial do FuzzyDesigner® está ilustrada na Figura 87.

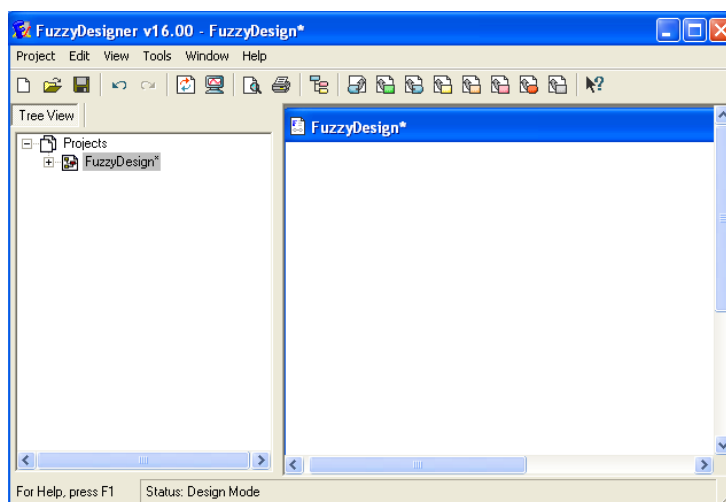


Figura 87 - Tela inicial do FuzzyDesigner®.

Na barra de ferramentas do *software*, ilustrada na Figura 88, temos todos os blocos de montagem que serão necessários para a construção do controlador. Os blocos são facilmente inseridos quando se clica em um dos seus respectivos ícones, sendo então mostrada de imediato uma janela de parametrização do bloco.



Figura 88 - Barra de ferramentas para inserção dos blocos de montagem.

Procedeu-se a inserção dos blocos para construção do controlador. Inicialmente, foi inserido o bloco *Input Port*. Este bloco é a entrada do controlador responsável por receber um valor real de *input* no sistema, semelhante a um bloco de entrada analógica encontrado em outras ferramentas. Ao clicar sobre o ícone para adicioná-lo abrirá a janela de parametrização onde foi necessário inserir o nome da entrada. Adicionou-se outro bloco *Input Port* para representar a entrada derivada do erro conforme mostrado na Figura 89.

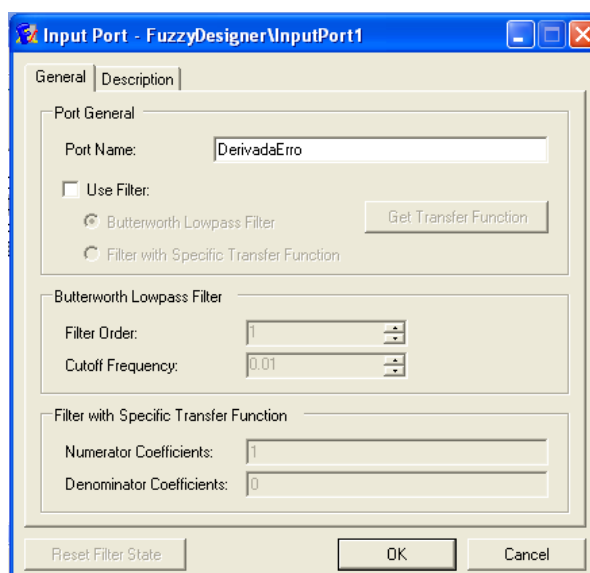


Figura 89 - Janela de parametrização dos blocos Input Port.

Após o término da configuração, os blocos *Input Port* foram automaticamente posicionados na tela conforme ilustrado na Figura 90.

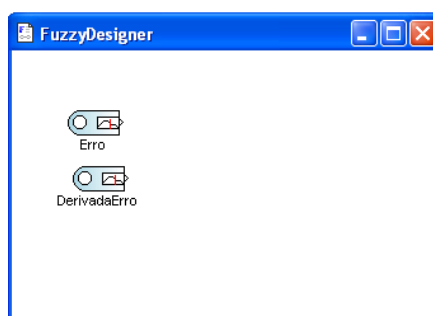


Figura 90 - Inserção dos blocos Input Port.

Adicionaram-se os blocos *Input Linguistic Variable* que são os blocos responsáveis pela *fuzificação* da entrada, ou seja, transforma valores reais de entrada para valores normalizados do universo *Fuzzy*. Neste bloco é possível se configurar o *range* de entrada (valores de limite mínimo e máximo), o tipo de entrada (o *software* possui diversos tipos pré-definidos, como corrente, temperatura, frequência, etc.), e o mais importante, a configuração da função de pertinência. Na tela de parametrização foi inserido o nome da variável e na opção *Input Link* foi selecionado o bloco que vai ser interligado à sua entrada conforme ilustrado na Figura 91, pois a topologia utilizar como entrada o erro e a derivada do erro.

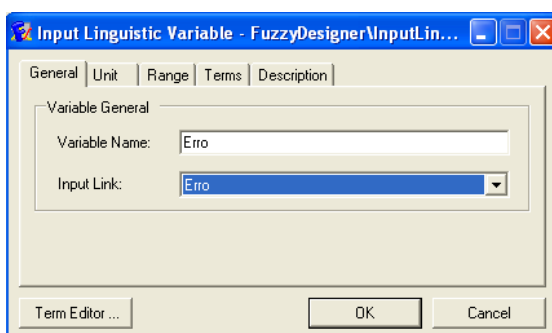


Figura 91 - Configurando o nome da variável e a interligação do bloco *Input Linguistic Variable*.

Na aba *Range* foi inserido os valores de limite inferior e superior da variável iguais a -1 e 1, respectivamente, conforme indicado na Figura 92.

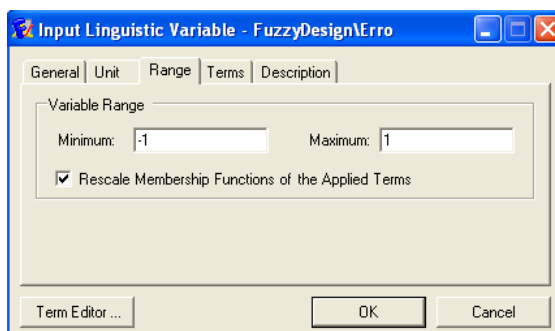


Figura 92 - Configurando os valores de limite inferior e superior no bloco *Input Linguistic Variable*.

Repetiu-se o mesmo procedimento para inserir a variável linguística de entrada referente à derivada do erro de controle. Após o término da configuração, os blocos *Input Linguistic Variable* foram automaticamente posicionados na tela conforme ilustrado na Figura 93.

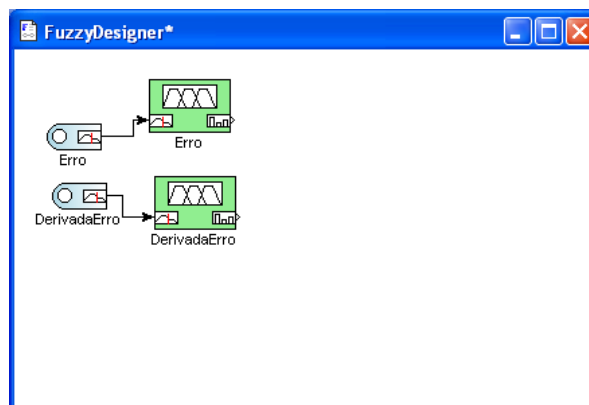


Figura 93 - Inserção dos blocos *Input Linguistic Variable*.

Nos blocos *Input Linguistic Variable* foi necessário configurar as funções de pertinência definidas na seção 4.5. A função pode ser modificada graficamente clicando-se nos pontos indicados nas linhas e arrastando-os até ficar no formato desejado. A Figura 94 apresenta a tela com as funções de pertinência já configuradas.

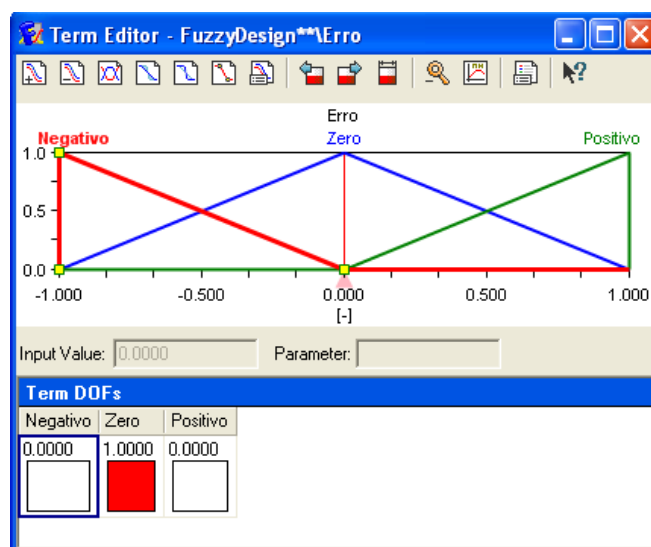


Figura 94 - Configuração da função de pertinência da variável linguística erro.

Repetiu-se o procedimento para configurar a função de pertinência da outra entrada. A Figura 95 apresenta a tela com as funções de pertinência já configuradas.

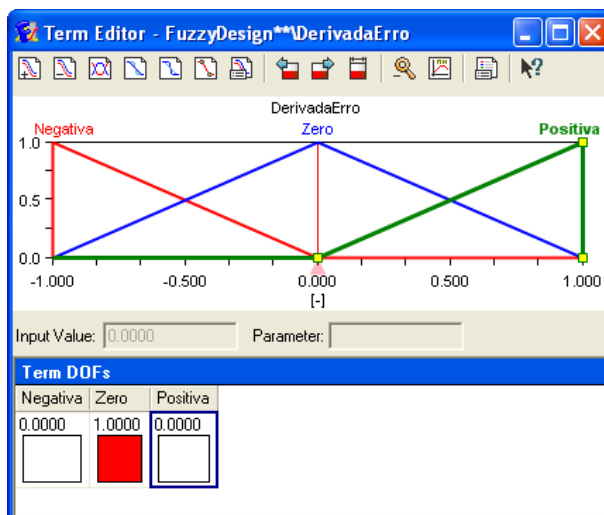


Figura 95 - Configuração da função de pertinência da variável linguística derivada do erro.

Procedeu-se a inserção do bloco de variável linguística que representará a saída do controlador. Este bloco denominado *Output Linguistic Variable* é responsável pela *defuzificação* da saída, ou seja, transforma a saída ou ações de controle do sistema (normalizadas no universo *Fuzzy*) para ações de controle não-*Fuzzy* (compatíveis com o universo real do sistema). Ao clicar no ícone correspondente a este bloco a janela de parametrização foi aberta. Nomeou-se o bloco de “Saída”. Na aba *Range* foi inserido os valores de limite inferior e superior da variável iguais a -1 e 1, respectivamente. Após o término da configuração, o bloco foi automaticamente posicionado na tela conforme ilustrado na Figura 96.

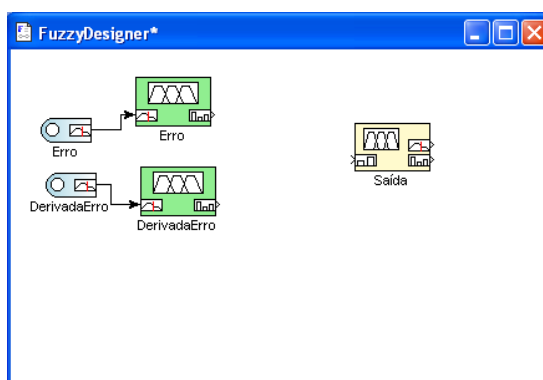


Figura 96 - Inserção do bloco *Output Linguistic Variable*.

Foi necessário configurar as funções de pertinência para o bloco *Output Linguistic Variable* definidas na seção 4.5. A função foi configurada da mesma forma dos blocos de entrada, clicando-se nos pontos indicados nas linhas e arrastando-os até ficar no formato desejado. A Figura 97 apresenta a tela com as funções de pertinência já configuradas.

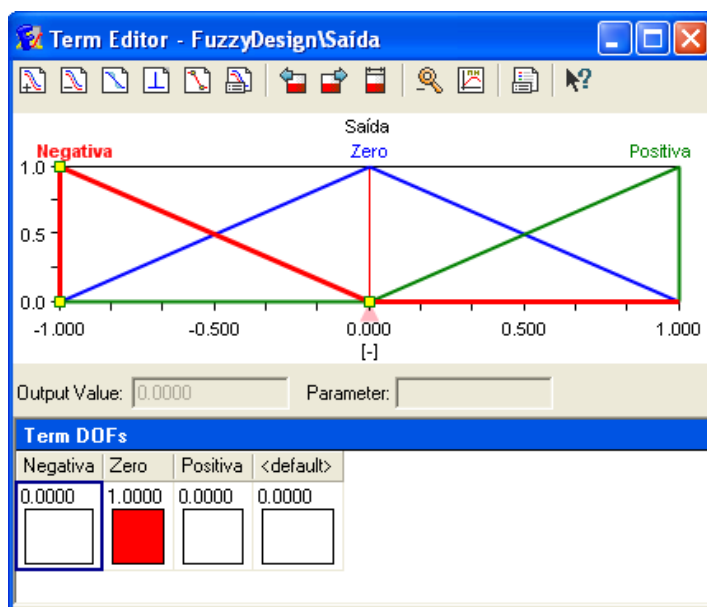


Figura 97 - Configuração da função de pertinência da variável linguística saída.

Inseriu-se o bloco *Output Port*. Este bloco é a saída do sistema, representando de maneira “figurativa” o borne do controlador responsável por enviar sinais ao atuador do sistema, como uma válvula, por exemplo. Ao clicar sobre o ícone para adicioná-lo irá aparecer a janela de parametrização onde foi necessário inserir o nome da saída ao qual foi nomeada de SAIDA. Após o término da configuração, o bloco *Output Port* foi automaticamente posicionado na tela conforme ilustrado na Figura 98.

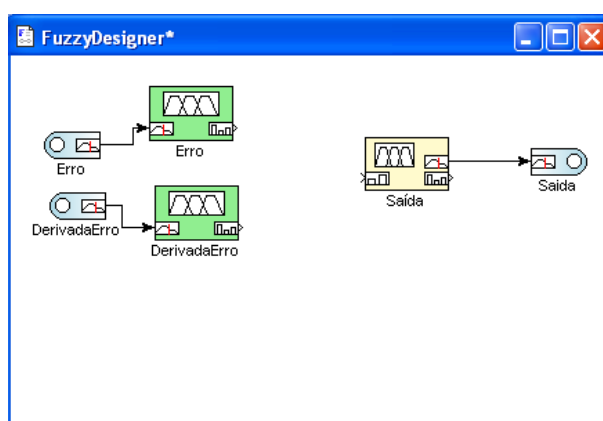


Figura 98 - Inserção do bloco Output Port.

Finalizado a inserção dos blocos de entrada e saída, procedeu-se a inserção do bloco *Rule Block*. É neste bloco onde são inseridas as regras do sistema *Fuzzy* definida pelo usuário

de acordo com suas necessidades, manipulando-se a saída de acordo com diferentes possibilidades de sinais de entrada. A Figura 99 ilustra o diagrama completo.

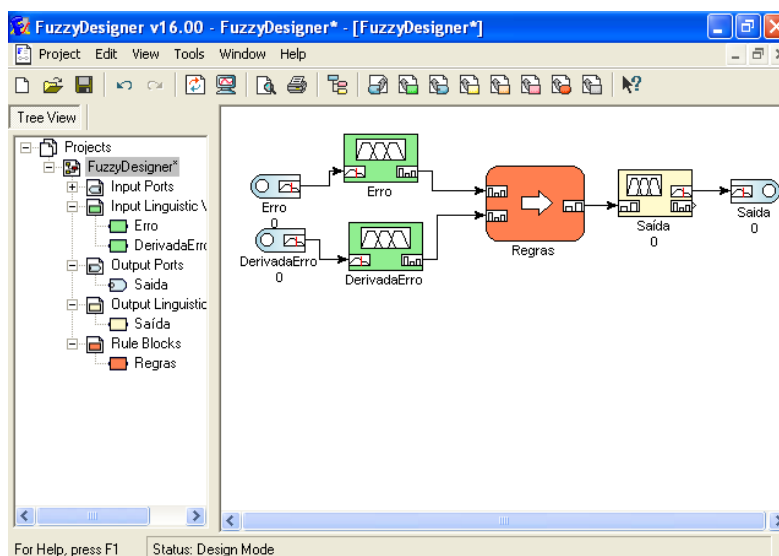


Figura 99 - Inserção do bloco Rule Block.

Após se inserir e interligar todos os blocos é necessário inserir as regras do sistema *Fuzzy* que são responsáveis por determinar as saídas do sistema de acordo com as diferentes entradas analisadas por ele. As nove regras *Fuzzy* que foram definidas na seção 4.4 foram inseridas conforme ilustrado na Figura 100.

Index	Active	Rule DDF	IF		THEN		RW
			Erro	DerivadaErro	Saída		
1	<input checked="" type="checkbox"/>	0.0000	Negativo	Negativa	Negativa	1	
2	<input checked="" type="checkbox"/>	0.0000	Negativo	Zero	Negativa	1	
3	<input checked="" type="checkbox"/>	0.0000	Negativo	Positiva	Zero	1	
4	<input checked="" type="checkbox"/>	0.0000	Zero	Negativa	Negativa	1	
5	<input checked="" type="checkbox"/>	1.0000	Zero	Zero	Zero	1	
6	<input checked="" type="checkbox"/>	0.0000	Zero	Positiva	Positiva	1	
7	<input checked="" type="checkbox"/>	0.0000	Positivo	Negativa	Zero	1	
8	<input checked="" type="checkbox"/>	0.0000	Positivo	Zero	Positiva	1	
9	<input checked="" type="checkbox"/>	0.0000	Positivo	Positiva	Positiva	1	
*							

Figura 100 - Inserindo a base de regras no Rule Block.

O FuzzyDesigner[®] permite a simulação das entradas e monitoramento das saídas geradas pelo sistema *Fuzzy*. Este recurso serve como uma base para o ajuste *off-line* dos parâmetros de projeto do sistema. A simulação é feita clicando-se em *Tools* → *Simulation*. Irá ser exibida a tela ilustrada na Figura 101.

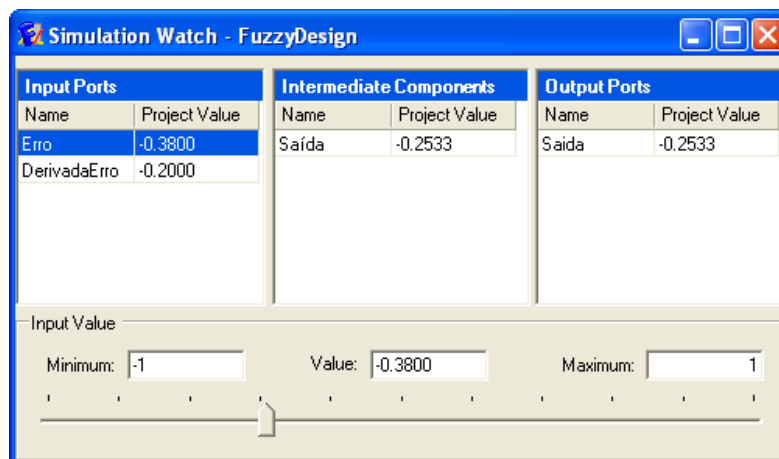


Figura 101 - Simulação do sistema *Fuzzy* no FuzzyDesigner[®].

Na tela de simulação os valores de todas as variáveis dos componentes utilizados são exibidos em três colunas. A primeira coluna mostra os valores referentes às portas de entrada. Já a segunda coluna mostra os componentes intermediários. A última coluna mostra os valores referentes à porta de saída. O parâmetro *Value* no canto inferior da tela permite definir o valor da entrada selecionada. Quando o valor de entrada é alterado, o projeto do sistema difuso é avaliado estaticamente, isto é, a operação de inferência é executada e o valor da saída é atualizado. Isso permite ter acesso a uma simulação em tempo real sobre o que está ocorrendo no sistema da mesma forma que foi realizado na *toolbox Fuzzy* do *software Matlab*[®] na seção 4.5.

Para utilizar no RSLogix5000[®] o controlador *Fuzzy* criado no FuzzyDesigner[®] é necessário exportar todas as configurações em forma de um *Add-On Instruction*. Com isso, pode-se importar o sistema *fuzzy* no RSLogix5000[®], implementar o controlador na rotina *ladder* e executá-lo em controladores da família *Logix5000* da Rockwell Automation. Este procedimento será realizado na próxima seção.

. Realizou-se a geração do *Add-On Instruction* clicando-se em *Tools* → *Add-On Instruction* → *Instruction Generator*. Porém, as seguintes condições devem ser atendidas para gerar uma AOI:

- Todos os blocos do sistema *Fuzzy* devem estar conectados;

- O sistema *Fuzzy* deve incluir pelo menos uma porta de entrada e uma porta de saída
- As regras devem ter condições não vazias.

Como todas as regras foram atendidas, uma tela foi aberta devendo-se informar o nome da *Add-On Instruction* conforme indicado na Figura 102.

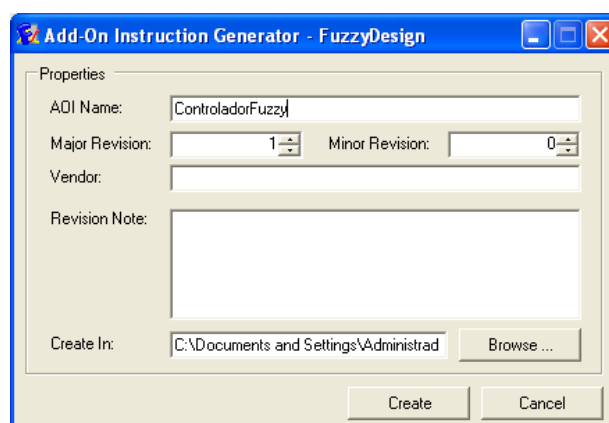


Figura 102 - Configurando o *Add-On Instruction* a ser criado.

Para importar o controlador criado no RSLogix5000® abriu-se novamente a rotina da seção 3.6 e clicou-se com o botão direito na pasta *Add-On Instructions* na barra de arquivos do projeto escolhendo a opção *Import Add-On Instruction* conforme ilustrado na Figura 103.

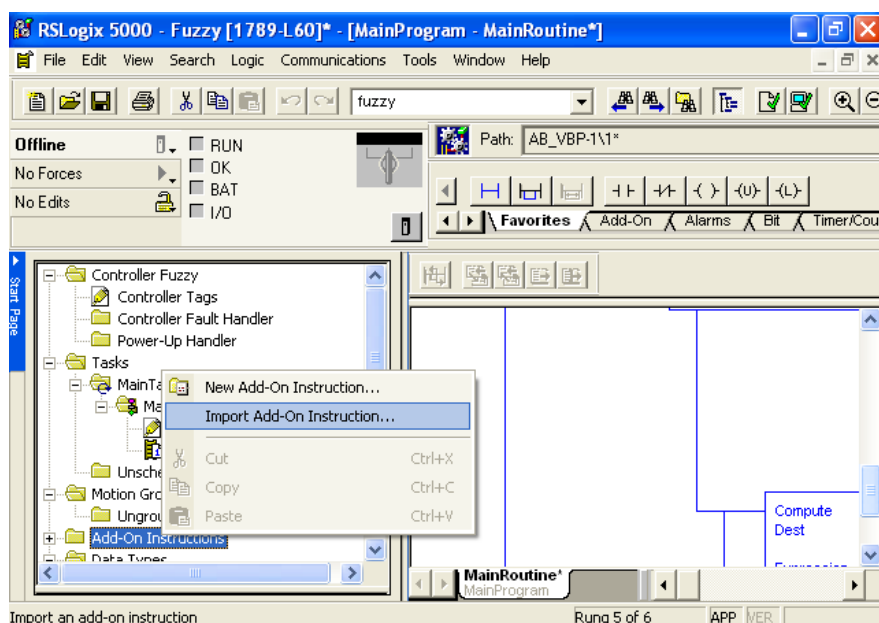


Figura 103 - Procedimento utilizado para importar o controlador Fuzzy.

Abrirá uma tela conforme ilustrado na Figura 104 onde foi necessário selecionar o arquivo de importação XML de extensão “L5X” o qual contém todas as configurações do sistema *fuzzy* criado no FuzzyDesigner®.

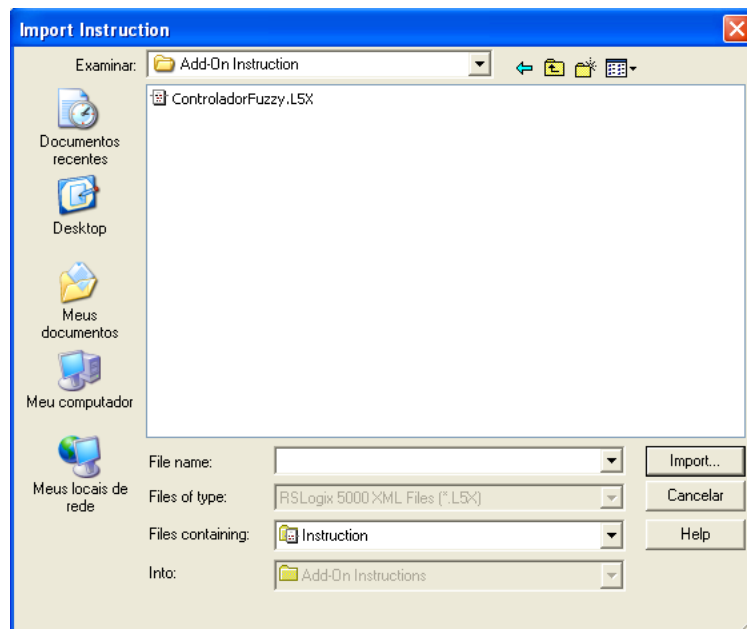


Figura 104 - Selecionando o arquivo XML de importação.

Em seguida, o controlador selecionado é devidamente importado para a biblioteca de instruções do tipo *Add-On Instruction*, e deve então ser inserido manualmente na rotina de acordo com a necessidade do usuário.