

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA

Desenvolvimentos de Malhas de Controle por Técnicas de Escalonamento de Ganhos e Lógica *Fuzzy*

Carlos Roberto de Araújo

**Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica
como parte dos requisitos necessários para a obtenção de título de Mestre em
Engenharia Elétrica.**

SETEMBRO DE 2009

Itajubá – MG

A meus pais e à minha namorada, dedico este trabalho com muito respeito e admiração.

AGRADECIMENTOS

Agradeço aos meus pais, Nelza e Roberto, por tudo o que fizeram e continuam fazendo por mim.

À minha namorada, Flávia, pelo incentivo e compreensão nos momentos de dificuldade.

À Universidade Federal de Itajubá pela estrutura disponibilizada para a realização deste trabalho.

Ao Professor-Orientador Carlos Alberto Murari Pinheiro, pelo competente e paciente acompanhamento.

Ao Professor da Universidade Federal de Ouro Preto, Agnaldo José de Rocha Reis, pela ajuda na obtenção de material teórico para o trabalho.

A todos os meus amigos.

E, finalmente, a Deus, que sempre nos conforta e direciona para um porto seguro.

RESUMO

Neste trabalho, apresenta-se o desenvolvimento de malhas de controle por meio de técnicas de escalonamento de ganhos e lógica *fuzzy*. Esta também é conhecida como lógica difusa ou nebulosa. As estratégias de controle foram aplicadas em uma planta de nível não-linear cujo comportamento se modifica de forma bastante significativa conforme a mudança da referência do sistema. Foram realizadas três abordagens distintas. Os resultados obtidos foram confrontados entre si, incluindo comparações com resultados de um controlador Proporcional + Integral (PI) com ganhos fixos.

Técnicas de controle linear apresentam limitações em processos não-lineares porque os controladores possuem ganhos fixos que não fornecem bons resultados em uma ampla faixa de operação do sistema. Dessa forma, a motivação para este trabalho é o desenvolvimento de estratégias capazes de aumentar a eficiência do sistema de controle através de escalonamento de ganhos e lógica *fuzzy*.

As estratégias de controle desenvolvidas utilizaram técnicas de escalonamento de ganhos através de informações da referência de entrada da malha de controle do processo e de informações da saída da planta. A última estratégia de controle utilizou lógica *fuzzy*.

Como exemplo de aplicação foi utilizado um sistema de controle de nível. Foram realizadas simulações computacionais com o emprego de um modelo matemático do tipo *NARMAX* (não linear auto-regressivo de média móvel com entradas exógenas) do processo citado. Também foi utilizada uma planta em escala reduzida objetivando a realização de testes reais, complementando os resultados obtidos por simulações numéricas.

Os resultados dos ensaios demonstraram que as abordagens utilizadas foram capazes de melhorar o desempenho do controlador PI de ganhos fixos. E as simulações reproduziram bem os ensaios realizados, demonstrando a boa qualidade do modelo não linear obtido.

Palavras-chave:

Lógica *Fuzzy*, Escalonamento de ganhos, processos não-lineares, sistema de controle de nível.

ABSTRACT

In this work is presented the development of control loops using techniques of gain scheduling and fuzzy logic. This is also known as diffuse logic. The control strategies were applied to a nonlinear plant-level whose behavior is changed quite significantly as the change of reference system. Were made three distinct approaches. The results were compared with each other, including comparisons with results of a Proportional + Integral controller (PI) with fixed gains.

Linear control techniques have limitations in nonlinear processes because the controllers have fixed gains that do not provide good results in a wide range of system operation. Thus, the motivation for this work is the development of strategies to increase the efficiency of the control system by scheduling gains and fuzzy logic.

The developed strategies of control had used techniques of scheduling gains through information of the input's reference of the control loop of the process and information of the plant's output. The last control strategy used fuzzy logic.

As an application example it was used a level control system. Computer simulations were performed with the use of a mathematical model of the type NARMAX (nonlinear autoregressive moving average with exogenous inputs) of cited process. It was also used a plant in reduced scale objectifying accomplishment of real tests, complementing the results obtained by numerical simulations.

The results of the tests showed that the approaches used were able to improve the performance of the PI controller of fixed gains. And the simulations reproduced well the tests, demonstrating the good quality of the nonlinear model obtained.

Keywords:

Fuzzy Logic, Gain scheduling, nonlinear processes, level control system.

SUMÁRIO

1. INTRODUÇÃO	1
2. REVISÃO BIBLIOGRÁFICA	3
3. CONCEITOS BÁSICOS	5
3.1 Modelagem Matemática	5
3.2 Sistemas <i>Fuzzy</i>	7
4. MÉTODOS E DESENVOLVIMENTOS	16
4.1 Identificação do Sistema – Modelo Linear	17
4.2 Modelo Não-Linear	18
5. ESTRATÉGIAS DE CONTROLE	21
5.1 Controlador PI com Ganhos Fixos	21
5.2 Controlador PI com Ganhos Escalonados pela Referência	22
5.3 Controlador PI com Ganhos Escalonados via Dados de Saída	23
5.4 Controlador <i>Fuzzy</i>	25
6. RESULTADOS	32
6.1 Controlador PI com Ganhos Fixos	32
6.2 Controlador PI com Ganhos Escalonados pela Referência	32
6.3 Controlador PI com Ganhos Escalonados via Dados de Saída	33
6.4 Controlador <i>Fuzzy</i>	34
7. CONCLUSÕES E TRABALHOS FUTUROS	37
REFERÊNCIAS BIBLIOGRÁFICAS	38
ANEXO 1 – PROGRAMA DO MODELO <i>NARMAX</i>	41
ANEXO 2 – PROGRAMA DE OBTENÇÃO DO MODELO <i>FUZZY</i>	43
ANEXO 3 – PROGRAMA DE SIMULAÇÃO DO CONTROLE <i>FUZZY</i>	46
ANEXO 4 – SUB-ROTINA DO <i>LABVIEW</i> DO CONTROLADOR <i>FUZZY</i>	48
ANEXO 5 – SUB-ROTINA PARA ESCALONAMENTO DE GANHOS I	51
ANEXO 6 – SUB-ROTINA PARA ESCALONAMENTO DE GANHOS II	52

LISTA DE FIGURAS

Figura 1 – Sistema de Controle de Nível.....	1
Figura 2 – Tanque com área de seção transversal variável.....	6
Figura 3 – Exemplo de variável linguística.	9
Figura 4 – Conjuntos <i>fuzzy</i> para a variável temperatura.	10
Figura 5 – Exemplo de domínio associado a conjuntos <i>fuzzy</i>	10
Figura 6 – Ilustração de um sistema de inferência <i>fuzzy</i>	14
Figura 7 – Bancada de ensaios.	16
Figura 8 – Resultados do modelo <i>NARMAX</i>	20
Figura 9 – AutoCorrelação do sinal medido (nível)	20
Figura 10 – Respostas do sistema para diferentes pontos de operação.....	22
Figura 11 – Gráfico com as informações dos ganhos variáveis.	23
Figura 12 – Controlador <i>fuzzy</i>	26
Figura 13 – Dados de ensaios práticos.	26
Figura 14 – Funções de pertinência para a entrada relativa ao erro.	27
Figura 15 – Funções de pertinência para a entrada relativa à integral do erro.	27
Figura 16 – Superfície de controle.	28
Figura 17 – Dados de medidas reais e do modelo <i>fuzzy</i> da variável de comando....	28
Figura 18 – Dados de medidas reais e do modelo <i>fuzzy</i> da variável erro.	29
Figura 19 – Dados de medidas reais e do modelo <i>fuzzy</i> da integração do erro.....	29
Figura 20 – Resultados de simulações com controlador PI convencional.....	32
Figura 21 – Resultados reais com PI de ganhos escalonados via referência.	33
Figura 22 – Simulações com escalonamento via referência de entrada.	33
Figura 23 – Resultados reais com escalonamento de ganhos via dados de saída. ...	34
Figura 24 – Resultados de simulações com escalonamento de ganhos via saída....	34
Figura 25 – Resultados de ensaios reais com controlador <i>fuzzy</i>	35
Figura 26 – Resultados de simulações com controlador <i>fuzzy</i>	35
Figura 27 – Interface gráfica do sistema de controle.....	50
Figura 28 – Realização do controlador <i>fuzzy</i>	50

LISTA DE TABELAS

Tabela 1 – Dados de ensaios para identificação ($SP = 4$ para $SP = 6$).....	17
Tabela 2 – Ganhos para diversos pontos de operação.....	22
Tabela 3 – Comparação entre controladores (Referência = 8 cm).....	36

LISTA DE ABREVIATURAS

<i>Anfis</i>	<i>Adaptive Neuro-Fuzzy Inference System</i>
<i>I-PD-Fuzzy</i>	<i>Integral-Proportional Derivative-Fuzzy</i>
K_i	Ganho Integral
K_p	Ganho Proporcional
M_p	Máximo Pico
<i>NARMAX</i>	<i>Non-Linear Auto-Regressive Moving Average with Exogenous Input</i>
<i>PI</i>	Proporcional + Integral
<i>RBF</i>	<i>Radial Basis Function</i>
<i>SP</i>	<i>Set-Point</i>
T_a	Tempo de Acomodação
ω	Frequência natural de oscilação
ξ	Fator de Amortecimento

1. INTRODUÇÃO

Este trabalho consiste em implementações e comparações de estratégias de controle para um processo de nível típico em aplicações industriais. O sistema escolhido servirá de modelo para comparações de algumas estratégias de controle que serão implementadas experimentalmente nesta dissertação. A Figura 1 ilustra um processo de nível. A capacitância (C) representa a capacidade de armazenamento do tanque, e geralmente está relacionada com a seção transversal do reservatório. A resistência (R) representa o esforço do deslocamento do fluido através da tubulação e da válvula de saída (LUNA, et al., 2002), e geralmente depende da seção transversal da tubulação de saída e da viscosidade do fluido.

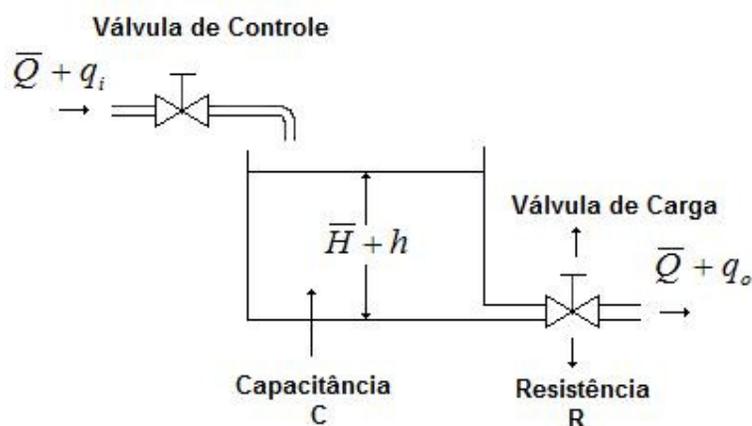


Figura 1 – Processo de nível.

Este tipo de sistema é bastante relevante em aplicações industriais. Ele está presente em diversas áreas de aplicações, como mineração, siderurgia, saneamento básico, etc. Em alguns casos, um sistema de controle de nível eficiente pode estar relacionado à diminuição nos custos de produção e, em outros, a questões de segurança. Independente dos aspectos de interesse, é fato que as indústrias que empregam esse tipo de processo necessitam de procedimentos adequados, principalmente devido à alta competitividade dos mercados atuais.

Sistemas de nível possuem características não-lineares. Assim, a atuação de malhas de controle que utilizam técnicas lineares, para diversos pontos de operação destes processos, pode não apresentar bons resultados. Neste contexto, são empregadas estratégias de controle diferenciadas que possam obter melhores

resultados. Uma abordagem usual é a utilização de informações das variáveis principais ou auxiliares do processo. Por exemplo, a informação da saída do sistema, objetivando a atualização dos ganhos dos controladores utilizados (ASTRÖM e WITTENMARK, 1989). Esta abordagem pode ser classificada como uma técnica de controle com características adaptativas, e usualmente recebe a denominação de técnica de escalonamento de ganhos.

Outras abordagens têm empregado técnicas de inteligência artificial, como lógica *fuzzy* e redes neurais artificiais. Técnicas de controle *fuzzy* têm se mostrado muito oportunas. Encontram-se na literatura alguns trabalhos nesse sentido, como em sistemas de tanques acoplados (LUNA, et al., 2002, MELO e BERNARDES, 2006), em que o objetivo é o estudo e a comparação de técnicas de controle para sistemas não-lineares com múltiplas entradas e saídas.

Este trabalho está dividido nas seções citadas a seguir. O Capítulo 2 traz uma resenha bibliográfica de temas pesquisados e relacionados aos assuntos abordados nesta dissertação. O Capítulo 3 mostra a conceituação básica dos principais tópicos abordados. O Capítulo 4 aborda a metodologia utilizada na pesquisa. No Capítulo 5 são apresentadas as realizações das estratégias de controle propostas para este trabalho. O Capítulo 6 contém os resultados obtidos. E, finalmente, no Capítulo 7 encontram-se as conclusões e propostas para trabalhos futuros.

2. REVISÃO BIBLIOGRÁFICA

Sistemas de controle de nível são processos não-lineares fundamentais em muitas plantas industriais (BHAMBHANI e CHEN, 2008). A lógica *fuzzy* tem sido utilizada em sistemas não-lineares e, principalmente, em sistemas cujo modelo matemático está sujeito a incertezas (*apud* SANDRI e CORREA, 1999). Os principais problemas para se projetar controladores em processos de nível são as não-linearidades, variações de parâmetros, atrasos de transporte e perturbações diversas (KOUHI et al., 2008). Alguns trabalhos abordaram questões de processos com tanques acoplados (LUNA FILHO, GOSMANN e BAUCHSPIESS, 2002, MELO e BERNARDES, 2006), em que o objetivo é o estudo e a comparação de técnicas de controle em um sistema não-linear com múltiplas entradas e saídas. Controladores PID-*Fuzzy* baseados em modelos de previsão foram propostos em KAYACAN e KAYNAK (2006), sendo que o objetivo era obter um controlador que possuísse a capacidade de prever o valor futuro da saída do sistema e realizar correções, antecipando variações intensas na regulação do processo. CHATRATTANAWUTH et al. (2006) controlaram um sistema de nível por meio de um controlador I-PD-*Fuzzy*. Uma comparação entre estruturas de controladores PI em controle de nível pode ser encontrada em BHAMBHANI e CHEN (2008). Leis de controle difusas que usam fórmulas matemáticas ao invés de conhecimento heurístico sobre o processo também são utilizadas (HECKENTHALER e ENGELL, 1994). DUSSUD et al. (1998), desenvolveram um controlador nebuloso para o controle de nível de metal fundido em processos de vazamento contínuo, uma vez que um controlador PID clássico não é capaz de regular de modo adequado variações intensas de nível (o que pode levar a derramamento de material).

Na teoria de controle clássica, os controladores apresentam características operacionais fixas. Isso acarreta algumas limitações, uma vez que sistemas reais necessitam de certa capacidade de adaptação em face às variações dos processos envolvidos (COELHO e ALMEIDA, 2003). Nesse sentido, os sistemas de controle *fuzzy* podem ser um meio eficiente para se superar as limitações dos controladores clássicos, porque o algoritmo de controle dos mesmos é formado por um conjunto de regras lógicas capazes de descrever em uma rotina a experiência humana, intuição e heurística para controlar um processo (*apud* SANDRI e CORREA, 1999).

Processos reais apresentam características não-lineares, o que leva as malha de controle clássicas a apresentarem desempenhos diferentes para diversos pontos de operação em muitos sistemas práticos. Dessa forma, malhas de controle *fuzzy* podem constituir uma solução interessante, porque apresentam potencial para tratar processos com grau de incerteza nos seus valores, e capacidade de agregação de informações linguísticas por meio de um conjunto de regras (PINHEIRO, 2000). Isso permite que os sistemas de controle resultantes possam apresentar melhores resultados.

Outra estratégia para se controlar sistemas não-lineares consiste no uso de medidas de variáveis auxiliares. Dentre elas pode-se usar a informação da saída do sistema para a atualização de ganhos de controladores convencionais (ASTRÖM e WITTENMARK, 1989).

3. CONCEITOS BÁSICOS

3.1 Modelagem Matemática

A utilização de modelos matemáticos de sistemas práticos é parte integrante de várias áreas das atividades humanas. A área do conhecimento que estuda maneiras de construir e implementar modelos matemáticos de sistemas reais chama-se modelagem matemática (AGUIRRE, 2004). Há várias formas de se classificar técnicas de modelagem. Frequentemente, as técnicas são classificadas em três grupos: modelagem caixa branca, modelagem caixa preta e modelagem caixa cinza.

Na modelagem caixa branca é necessário um conhecimento sobre as leis físicas que governam o sistema. Por essa razão, esse tipo de modelagem é também conhecido como modelagem pela física ou natureza do processo. Infelizmente, devido ao conhecimento e ao tempo necessários para modelar um sistema prático, partindo-se das relações físicas envolvidas, muitas vezes, torna-se oneroso a utilização desse procedimento.

Identificação de sistemas é uma área de modelagem matemática que estuda técnicas alternativas à modelagem caixa branca. Ela utiliza pouco ou nenhum conhecimento prévio do sistema, e é conhecida como modelagem caixa preta ou empírica. Em muitas aplicações será preferível utilizar técnicas de identificação para a obtenção de modelos. Em sistemas reais existem componentes de difícil modelagem por meio de equações físicas. O objetivo de modelos obtidos por técnicas de identificação é descrever relações de causa e efeito entre variáveis de entrada e saída. Nesse caso, os tipos de modelos, as técnicas usadas e os requisitos necessários são bastante distintos dos relacionados com modelagens pela natureza do processo.

A identificação caixa cinza situa-se entre a identificação caixa branca e a caixa preta. São empregadas informações auxiliares, não presentes no conjunto de dados utilizados durante a identificação. O tipo de informação auxiliar e a forma como a mesma é usada varia bastante. Assim sendo, existem métodos mais “claros” e mais “escuras” de identificação caixa cinza, de acordo com a quantidade de informação auxiliar.

Modelagem matemática é um procedimento muito importante para a engenharia de sistemas de controle. Os modelos matemáticos podem assumir diferentes formas. Dependendo do sistema e das circunstâncias do problema, um tipo de modelo pode ser mais adequado do que outro. Uma vez obtido o modelo matemático, podem ser utilizadas várias ferramentas analíticas e de computação para efeito de análise e projeto de controladores (OGATA, 2003).

3.1.1 Modelagem Matemática de Sistemas de Nível

O modelo matemático de sistemas de nível utilizado neste trabalho é definido da seguinte maneira: Considere um reservatório cuja área de seção transversal (A) pode se modificar com a altura (h) do nível do mesmo, como ilustrado na Figura 2 (ASTRÖM e WITTENMARK, 1989).

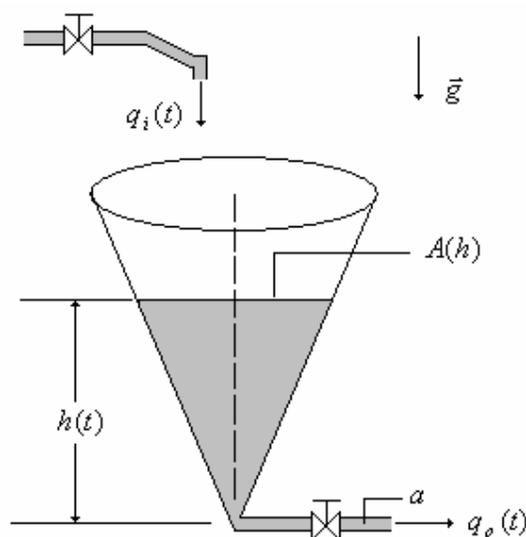


Figura 2 – Tanque com área de seção transversal variável.

A variação do nível no tempo (t) é definida pela vazão de entrada (q_i), pela área da seção transversal (a) da tubulação de escoamento de saída do reservatório, e pela aceleração da gravidade (g). A equação (1) representa um modelo não-linear usual de sistemas de nível.

$$\frac{d}{dt}[A(h) \cdot h(t)] = q_i(t) - a\sqrt{2gh(t)} \quad (1)$$

Para que se possa extrair a função de transferência da equação 1, é preciso que a mesma seja linearizada, uma vez que a função de transferência apenas é aplicada para sistemas lineares. Assim, considerando pequenas variações para a vazão de entrada e para o nível do tanque, e realizando algumas simplificações, pode-se linearizar o modelo em questão em torno de um determinado ponto de operação (q_{in}^0, h^0) , obtendo-se a função de transferência (2).

$$G(s) = \frac{\beta}{s + \alpha} \quad (2)$$

Sendo que:

$$\beta = \frac{1}{A(h^0)}; \quad \alpha = \frac{q_{in}^0}{2A(h^0)h^0} = \frac{a\sqrt{2gh^0}}{2A(h^0)h^0}. \quad (3)$$

3.2 Sistemas *Fuzzy*

Seres humanos são capazes de lidar com processos complexos baseados em informações vagas, em que as mesmas podem ser expressas em termos linguísticos. A teoria dos conjuntos *fuzzy* foi desenvolvida a partir de 1965 por Lotfi Zadeh, para tratar do aspecto vago da informação (SANDRI e CORREA, 1999). Esta teoria pode ser utilizada para traduzir em termos matemáticos informações vagas ou imprecisas (TANSCHKEIT, 2003). Sistemas *fuzzy* possuem algumas vantagens interessantes em relação a sistemas de controle clássico: são fáceis de entender; robustos a ruídos ou incertezas dos dados; conseguem modelar características não-lineares de complexidade arbitrária; podem ser combinados com outras técnicas; podem ser construídos por meio da experiência de especialistas; entre outras (DRIANKOV et al., 1996).

As primeiras aplicações bem sucedidas da lógica *fuzzy* situaram-se na área de sistemas de controle, e há uma utilização crescente de sistemas *fuzzy* em outros campos como, por exemplo, tomada de decisões, previsão de séries temporais,

mineração de dados, planejamento e otimização de sistemas, etc (TANSCHHEIT, 2003).

Na teoria clássica dos conjuntos, o conceito de pertinência de um elemento relacionado a um determinado conjunto é bem definido. Dado um conjunto A e um conjunto universo X , os elementos do universo somente pertencem ou não àquele conjunto. Isto pode ser expresso pela função característica definida por (4).

$$f_A(x) = \begin{cases} 1 & \Leftrightarrow x \in A \\ 0 & \Leftrightarrow x \notin A \end{cases} \quad (4)$$

Na teoria dos conjuntos *fuzzy* há generalização da função característica de modo que ela possa assumir um número infinito de valores no intervalo $[0,1]$. Um conjunto *fuzzy* A em um universo X é definido por uma função de pertinência $\mu_A(x) : X \rightarrow [0,1]$, e é representado pelo conjunto de pares ordenados (5).

$$A = \left\{ \frac{\mu_A(x)}{x} \right\}, \forall x \in X \quad (5)$$

Um determinado elemento pode pertencer a mais de um conjunto *fuzzy*, tendo diferentes graus de pertinência em cada um deles. O conjunto suporte de um conjunto *fuzzy* A é o conjunto de elementos no universo X para os quais $\mu_A(x) > 0$. Um conjunto *fuzzy* cujo suporte é um único ponto x' com $\mu_A(x') = 1$ é chamado de conjunto unitário *fuzzy* ou *singleton*. Assim, um conjunto *fuzzy* também pode ser visto como o mapeamento do conjunto suporte no intervalo $[0,1]$, o que implica em expressar o conjunto *fuzzy* por sua função de pertinência.

Conjuntos *fuzzy* apresentam universos contínuos ou discretos. Se o universo X for discreto e finito, o conjunto *fuzzy* A é normalmente representado por meio de um vetor (6), que contém os graus de pertinência em relação ao conjunto A dos elementos correspondentes de X .

$$\sum_{i=1}^n \frac{\mu_A(x_i)}{x_i} \quad (6)$$

Se o universo X for contínuo, emprega-se a notação (7), em que o símbolo de integral deve ser interpretado da mesma forma que o da soma no caso de um universo discreto.

$$\int_X \frac{\mu_A(x)}{x} \quad (7)$$

Conjuntos *fuzzy* podem ter denominações linguísticas. Uma variável linguística possui como valores nomes de conjuntos *fuzzy*. Por exemplo, o nível de um determinado processo pode ser uma variável linguística com as denominações: baixo; médio; alto; etc. Estes valores são representados por intermédio de conjuntos *fuzzy*, que, por sua vez, utilizam funções de pertinência, conforme ilustrado na Figura 3.

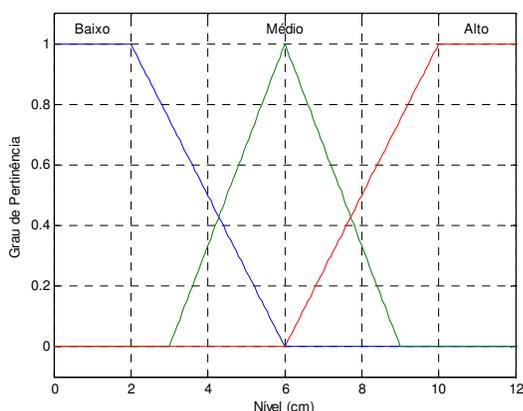


Figura 3 – Exemplo de variável linguística.

A vantagem das variáveis linguísticas é fornecer uma maneira sistemática para uma caracterização aproximada de fenômenos complexos ou vagos. Em essência, o uso de descrição linguística empregada por seres humanos, permite o tratamento de sistemas que são muito complexos para serem analisados em termos matemáticos convencionais (TANSCHKEIT, 2003).

Existem vários tipos de funções de pertinência utilizados na prática. A escolha do tipo depende do contexto e do conceito que se pretende representar (TANSCHKEIT, 2003). Suponha que se deseja construir um sistema de controle para ar condicionado. Para a descrição da variável temperatura, podem ser definidos os

conjuntos difusos quente, frio, confortável, fresco e morno. Estes conjuntos *fuzzy* estão mostrados na Figura 4, e a variável linguística em questão depende do domínio físico do problema (DRIANKOV et al., 1996).

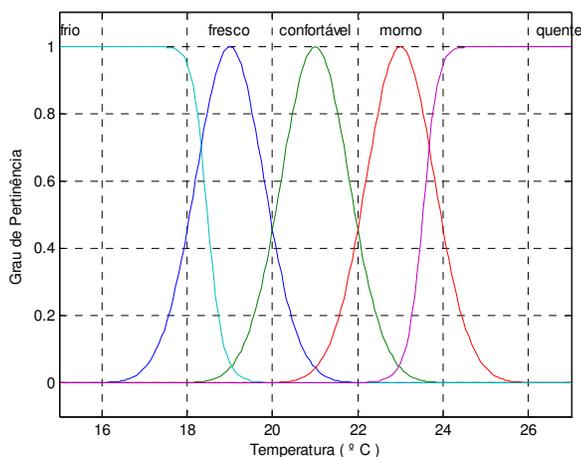


Figura 4 – Conjuntos *fuzzy* para a variável temperatura.

Assim, a princípio, as variáveis linguísticas são dependentes do domínio físico em que se está trabalhando. Mas, geralmente, em sistemas de controle *fuzzy* é conveniente o uso de variáveis linguísticas que não dependem do domínio físico particular. Para isso, pode-se, por exemplo, construir um domínio que tenha sete conjuntos *fuzzy* como Positivo Grande (PG), Positivo Médio (PM), Positivo Pequeno (PP), Zero (ZO), Negativo Pequeno (NP), Negativo Médio (NM) e Negativo Grande (NG). A Figura 5 ilustra os conjuntos *fuzzy* associados ao exemplo.

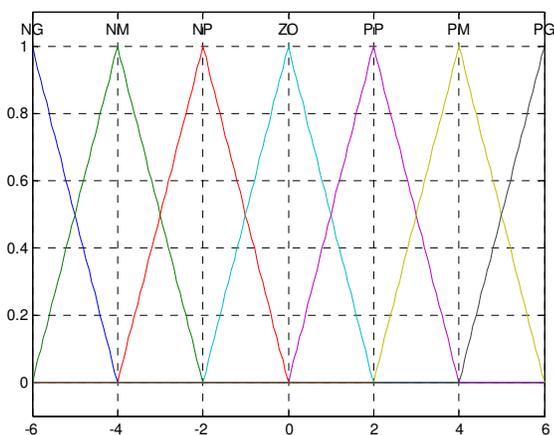


Figura 5 – Exemplo de domínio associado a conjuntos *fuzzy*.

Muitas vezes, as funções de pertinência podem ser construídas por meio da experiência do projetista, mas o mais frequente é o uso de funções padrão, como as funções triangular, trapezoidal ou gaussiana (TANSCHKEIT, 2003).

Funções de pertinência no domínio contínuo podem ser definidas por meio de funções analíticas. Por exemplo, a função (8) pode ser usada para definir as funções de pertinência associadas aos conjuntos baixo, médio e alto da Figura 3:

$$\mu_A(x) = (1 + (a(x - c))^b)^{-1} \quad (8)$$

A forma da função anterior pode ser modificada por meio da manipulação dos parâmetros a , b e c (TANSCHKEIT, 2003), resultando em (9), (10) e (11).

$$\mu_{baixo}(x) = (1 + 9x^2)^{-1} \quad (9)$$

$$\mu_{médio}(x) = (1 + 9(x - 0.5)^2)^{-1} \quad (10)$$

$$\mu_{alto}(x) = (1 + 9(x - 2)^2)^{-1} \quad (11)$$

As funções de pertinência de domínio discreto consistem de conjuntos de valores correspondendo a elementos discretos do universo. Por exemplo, se $X = \{0, 1, 2, 3, 4, 5, 6\}$, uma representação possível seria (12), (13) e (14).

$$\mu_{pequeno}(x) = \{0,3; 0,7; 1; 0,7; 0,3; 0; 0\} \quad (12)$$

$$\mu_{médio}(x) = \{0; 0; 0,3; 0,7; 1; 0,7; 0,3\} \quad (13)$$

$$\mu_{grande}(x) = \{0; 0; 0; 0; 0,3; 0,7; 1\} \quad (14)$$

Um conjunto *fuzzy* A em um universo X é vazio se e somente se sua função de pertinência é igual a zero sobre todo X :

$$A = \emptyset \Leftrightarrow \mu_A(x) = 0, \quad \forall x \in X. \quad (15)$$

O complemento A' de um conjunto *fuzzy* A é normalmente expresso por:

$$\mu_{A'}(x) = 1 - \mu_A(x), \quad \forall x \in X. \quad (16)$$

Dois conjuntos *fuzzy* A e B em X são iguais se suas funções de pertinência forem iguais sobre todo X :

$$A = B \Leftrightarrow \mu_A(x) = \mu_B(x), \quad \forall x \in X. \quad (17)$$

Um conjunto *fuzzy* A é um subconjunto de B se sua função de pertinência for menor ou igual à de B sobre todo X :

$$A \subset B \Leftrightarrow \mu_A(x) \leq \mu_B(x), \quad \forall x \in X. \quad (18)$$

Utilizando os operadores mínimo (min ou \wedge) e máximo (max ou \vee), assim como conceitos de soma algébrica, podem ser definidas as operações (19) e (20).

$$\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) = \mu_A(x) \cdot \mu_B(x) \quad \forall x \in X \quad (19)$$

$$\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x) \quad \forall x \in X \quad (20)$$

Uma norma-t é uma operação binária $\otimes: [0,1] \times [0,1] \rightarrow [0,1]$ tal que, $\forall x, y, z, w \in [0,1]$, as seguintes propriedades são satisfeitas:

- Comutatividade: $x \otimes y = y \otimes x$
- Associatividade: $(x \otimes y) \otimes z = x \otimes (y \otimes z)$
- Monotonicidade: se $x \leq y$, $w \leq z$, então $x \otimes w \leq y \otimes z$
- Condições de contorno: $x \otimes 0 = 0$ e $x \otimes 1 = x$

Uma co-norma-t, ou norma-s, é uma operação binária $\oplus: [0,1] \times [0,1] \rightarrow [0,1]$, que satisfaz às seguintes propriedades:

- Comutatividade: $x \oplus y = y \oplus x$
- Associatividade: $(x \oplus y) \oplus z = x \oplus (y \oplus z)$
- Monotonicidade: se $x \leq y$, $w \leq z$, então $x \oplus w \leq y \oplus z$
- Condições de contorno: $x \oplus 0 = x$ e $x \oplus 1 = 1$

Existem várias normas-t e normas-s. As mais utilizadas em trabalhos de engenharia são os operadores mínimo e produto algébrico para interseção, e o operador máximo para união.

As representações de sistemas *fuzzy* mais conhecidas são do tipo Mamdani e Takagi-Sugeno. Em geral um procedimento de inferência *fuzzy* é composto de três etapas: *fuzzyficação* das variáveis de entrada; composições das regras; agregação dos resultados por técnicas de *defuzzificação*. Estes passos serão mais detalhados a seguir.

A etapa de *fuzzyficação* consiste em determinar o grau de pertinência em relação aos conjuntos *fuzzy* utilizados, dado o conjunto de valores das entradas, por meio de funções de pertinência ou tabelas.

O processo de composição é realizado para cada regra e os métodos mais utilizados nos respectivos antecedentes empregam os operadores mínimo ou produto. Também é admitida operação de união nos antecedentes (operador máximo).

A agregação é o processo segundo o qual os conjuntos *fuzzy* que representam as saídas das regras são combinados em um único conjunto *fuzzy* ou valor ponderado. O processo de agregação é comutativo, assim, a ordem na qual as regras são executadas não é importante. Os métodos mais utilizados nesse processo empregam o operador máximo ou soma.

O valor resultante da etapa de agregação é convertido em um dado correspondente à grandeza física do modelo *fuzzy* em questão. O método mais usual de *defuzzificação* é o do centróide ou centro de área para os controladores do tipo Mamdani.

Os passos descritos anteriormente podem ser ilustrados na Figura 6, que representa valores de um sistema *fuzzy* com duas variáveis de entrada e uma de saída. Este sistema *fuzzy* ilustra o problema “Jantar para Dois” e é composto por três regras linguísticas descritas a seguir:

- Se o serviço é péssimo ou a comida é rançosa, então a gorjeta é baixa;
- Se o serviço é bom, então a gorjeta é média;
- Se o serviço é excelente ou a comida é deliciosa, então a gorjeta é alta.

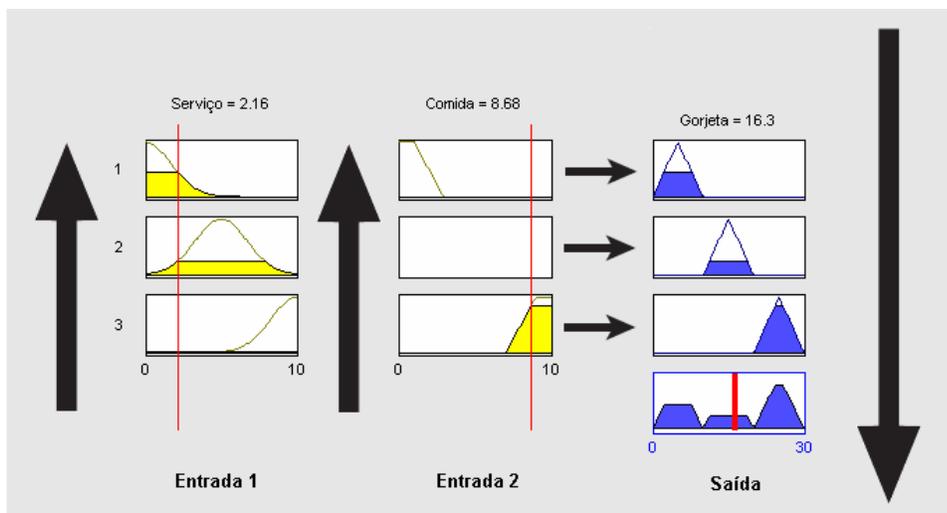


Figura 6 – Ilustração de um sistema de inferência *fuzzy*.

O modelo *fuzzy* da Figura 6 é do tipo Mamdani. Ele utiliza o operador máximo nos antecedentes das regras. Um conjunto *fuzzy* global que representa a composição das saídas das regras é gerado e uma ação de controle é retirada do mesmo.

Para um modelo *fuzzy* do tipo Mamdani a representação de uma regra genérica é dada por (21), onde X_1^i, \dots, X_k^i e Y^i são conjuntos nebulosos, x_1, \dots, x_k são as variáveis de entrada e y a informação da variável de saída (SANDRI e CORREA, 1999).

$$r_i : \text{If } x_1 = X_1^i \text{ And } x_2 = X_2^i \text{ And } \dots \text{ And } x_k = X_k^i \text{ Then } y_i = Y^i \quad (21)$$

Para um modelo *fuzzy* do tipo Takagi-Sugeno a representação de uma regra genérica é dada por (22), onde X_1^i, \dots, X_k^i são conjuntos nebulosos, x_1, \dots, x_k são as variáveis de entrada e y a informação da variável de saída, cujo valor é definido pela função $f_i(x_1, x_2, \dots, x_k)$, que geralmente é do tipo polinomial (23) (PINHEIRO, 2000).

$$r_i : \text{If } x_1 = X_1^i \text{ And } x_2 = X_2^i \text{ And } \dots \text{ And } x_k = X_k^i \text{ Then } y_i = f_i(x_1, x_2, \dots, x_k) \quad (22)$$

$$f_i(x_1, x_2, \dots, x_k) = c_{0i} + c_{1i}x_1 + \dots + c_{ki}x_k \quad (23)$$

Existem vários métodos para obtenção das funções de pertinência associadas aos conjuntos *fuzzy*, dos coeficientes das funções polinomiais, e das próprias regras *fuzzy*. Alguns procedimentos utilizam informações de especialistas ou dos operadores de um determinado processo. Outros utilizam formulações numéricas como métodos de agrupamento, otimização, mínimos quadrados, etc. Também existem procedimentos que utilizam técnicas de inteligência artificial como algoritmos evolutivos e redes neurais artificiais (SANDRI e CORREA, 1999).

4. MÉTODOS E DESENVOLVIMENTOS

Considera-se neste trabalho um processo de controle de nível como plataforma para os desenvolvimentos realizados. O processo utilizado é composto de um sistema de nível em escala reduzida, que contém um módulo eletrônico com a instrumentação do processo, um computador e um sistema de aquisição de dados. O módulo condiciona o sinal do transdutor capacitivo localizado no tanque superior, assim como o sinal de controle que aciona a servo-bomba elétrica. O sistema de aquisição de dados é um modelo NI USB-6008 do fabricante *National Instruments*, que possui 8 entradas e 2 saídas analógicas, 12 entradas/saídas digitais e um contador de 32 bits, além de uma interface USB. A foto da Figura 7 ilustra a bancada de ensaios resultante.

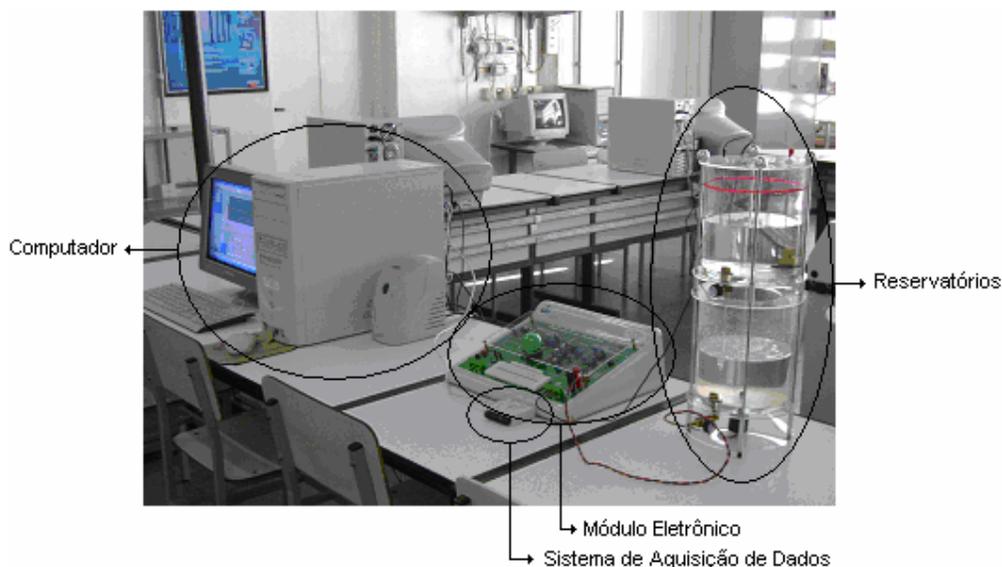


Figura 7 – Bancada de ensaios.

O processo possui dois reservatórios sobrepostos. Sem a ação do sistema de controle, o líquido do processo (água) se concentra no reservatório inferior. A partir do momento em que o sistema entra em funcionamento e um determinado valor de nível é ajustado na referência da malha de controle do processo, uma ação de comando é exercida sobre a servo-bomba elétrica do sistema, transferindo o líquido do tanque inferior para o superior. Existe uma interligação entre os dois tanques, de modo que o objetivo é controlar o nível de líquido no reservatório superior, onde um

transdutor do tipo capacitivo realiza a medição da grandeza de interesse. Uma entrada analógica do sistema de aquisição de dados coleta os dados do transdutor e uma saída analógica aciona o *driver* da servo-bomba elétrica do processo.

Foram desenvolvidas algumas estratégias de controle que serão mostradas nos próximos itens, assim como modelos matemáticos identificados referentes ao processo em questão.

4.1 Identificação do Sistema – Modelo Linear

Para a identificação do processo utilizado neste trabalho foram realizados experimentos e os resultados obtidos foram gravados com as respectivas informações da tensão de comando da bomba elétrica e do nível de líquido (no tanque superior). Foi utilizada uma malha de controle com um controlador PI de ganhos fixos. Dessa forma, a identificação foi feita em malha fechada (LYUNG, 2000). Os ensaios foram realizados para diferentes pontos de operação do sistema, ou seja, para diferentes valores de referência de entrada ou *set-point* (*SP*). O motivo de se utilizar diferentes pontos de operação é que, conforme já mostrado, o modelo real de um sistema de nível apresenta característica não-linear. Com os dados coletados foram obtidas funções de transferência (correspondentes a modelos lineares) para os pontos de operação do sistema. Para a obtenção dessas funções de transferência foi utilizado o *toolbox* de identificação de sistemas do *software Matlab* e dados como os da Tabela 1.

Tabela 1 – Dados de ensaios para identificação (*SP* = 4 para *SP* = 6).

N.º do par de Entrada/Saída	Tensão elétrica (V) - Entrada	Nível no tanque (cm) - Saída	<i>Set-Point</i>
1	5	3,95	6,00
2	5	3,95	6,00
3	5	3,98	6,00
4	5	4,01	6,00
5	5	4,01	6,00
6	5	3,98	6,00
7	5	3,95	6,00
...	6,00
999	0,05	6,10	6,00

Foram obtidos modelos lineares do tipo *ARX* (*auto-regressive with exogenous inputs*). A função do *Matlab* utilizada para a obtenção desses modelos foi a *arx*, cuja sintaxe é a seguinte: *arx(Data, [na nb nk])*. O parâmetro *Data* representa os dados de entrada e saída. Os parâmetros *na* e *nb* são ordens de polinômios (quando se expressam funções de transferência na forma polinomial) e o parâmetro *nk* é o atraso entre a entrada e a saída. As funções obtidas foram:

- *SP de 4 para 6 cm:*

$$\frac{Y(s)}{U(s)} = \frac{4,1901}{82,6446s + 1} \quad (24)$$

- *SP de 6 para 8 cm:*

$$\frac{Y(s)}{U(s)} = \frac{5,0722}{103,0928s + 1} \quad (25)$$

- *SP de 8 para 10 cm:*

$$\frac{Y(s)}{U(s)} = \frac{5,6778}{111,1111s + 1} \quad (26)$$

- *SP de 10 para 12 cm:*

$$\frac{Y(s)}{U(s)} = \frac{6,2195}{121,9512s + 1} \quad (27)$$

Verifica-se que as funções de transferência se modificam conforme a faixa de operação do sistema, pois o processo real é não-linear.

4.2 Modelo Não-Linear

Na modelagem, uma importante questão é a escolha da estrutura que deverá representar o comportamento de um sistema dinâmico. Algumas representações utilizadas na modelagem de sistemas não-lineares são: redes neurais; funções de base radial, *RBF*; séries de Volterra; *Wavelets*; funções polinomiais e racionais; equações diferenciais polinomiais (AGUIRRE et al., 1998).

Na tentativa de se obter um modelo mais completo do processo de nível, empregou-se uma representação não-linear tipo *NARMAX* (sigla em inglês para modelo não-linear auto-regressivo de média móvel com entradas exógenas) polinomial. Esta estrutura se ajusta aos dados de ensaio com boa exatidão, desde que os mesmos não apresentem variações abruptas. Foi elaborado um programa (ANEXO 1) em *Matlab* para estimar os parâmetros (θ) do modelo em questão (28), em que o vetor Y contém medidas da saída do sistema e F é a matriz de medidas dos regressores. Os parâmetros podem ser estimados pela expressão (32) obtida pelo método dos mínimos quadrados.

$$y(k) = \theta_1 y(k-1) + \theta_2 y(k-2) + \theta_3 u(k-3) + \theta_4 u(k-1) + \theta_5 y^2(k-1)u(k-1) + \theta_6 u^2(k-1)u(k-3) + \theta_7 y^3(k-3) + \theta_8 y(k-2)u(k-2) + \theta_9 y^2(k-3)u(k-3) \quad (28)$$

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \dots \\ \theta_9 \end{bmatrix} \quad (29)$$

$$Y = \begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \dots \\ \dots \end{bmatrix} \quad (30)$$

$$F = \begin{bmatrix} y(2) & y(1) & u(1) & u(2) & y^2(2)u(2) & u^2(2)u(0) & y^3(0) & y(1)u(1) & y^3(0)u(0) \\ y(3) & y(2) & u(2) & u(3) & y^2(3)u(3) & u^2(3)u(1) & y^3(1) & y(2)u(2) & y^3(1)u(1) \\ y(4) & y(3) & u(3) & u(4) & y^2(4)u(4) & u^2(4)u(2) & y^3(2) & y(3)u(3) & y^3(2)u(2) \\ \dots & & & & & \dots & & & \dots \\ \dots & & & & & \dots & & & \dots \end{bmatrix} \quad (31)$$

$$\theta = \frac{F^T Y}{F^T F} \quad (32)$$

Em (33) têm-se os valores dos coeficientes resultantes do modelo *NARMAX* adotado. O modelo obtido apresentou uma correlação cruzada igual a 99,46%. A Figura 8 ilustra os dados de saída do processo real (provenientes das medições) e

os dados resultantes do modelo (provenientes de simulação computacional). A Figura 9 apresenta a autocorrelação do sinal medido.

$$\theta = \begin{bmatrix} 0.50097 \\ 0.49709 \\ 0.075072 \\ -0.046523 \\ 0.0012312 \\ -0.01726 \\ 8.0467 \times 10^{-6} \\ 0.0050303 \\ -0.0015785 \end{bmatrix} \quad (33)$$

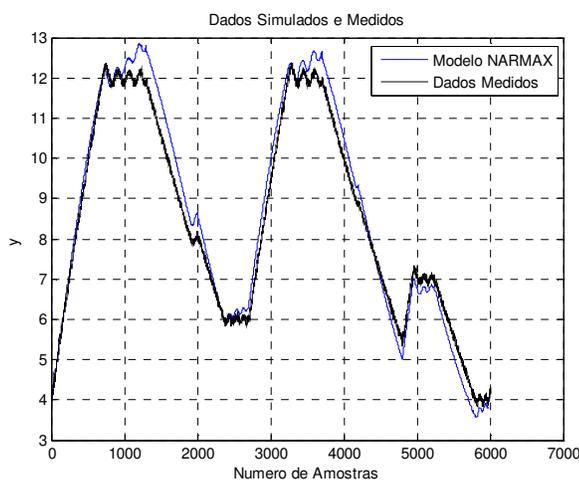


Figura 8 – Resultados do modelo *NARMAX*.

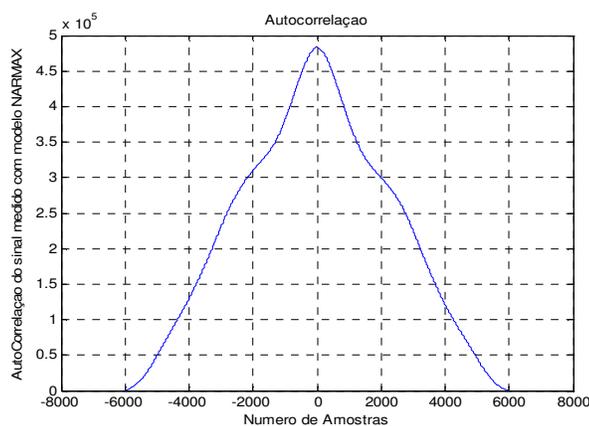


Figura 9 – AutoCorrelação do sinal medido (nível)

5. ESTRATÉGIAS DE CONTROLE

5.1 Controlador PI com Ganhos Fixos

O modelo básico de um controlador PI contínuo é expresso por (34), onde K_p é o ganho proporcional, e K_i é o ganho integral. A variável E representa o erro da malha de controle, a diferença entre a referência de entrada (*set-point*) do sistema e o valor da grandeza controlada do processo. A variável U representa a informação de comando da planta. Os valores adotados para os ganhos foram $K_i = 5$ e $K_p = 4$. Estes valores foram ajustados experimentalmente objetivando estabelecer um máximo pico (M_p) menor ou igual a 5% e um tempo de acomodação (T_a) em torno de 60 segundos.

$$\frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} \quad (34)$$

A representação discreta do controlador PI, utilizando-se a aproximação direta de Euler (35) e um tempo de amostragem T de 0,1 segundo, é dada por (36). A equação a diferença resultante é expressa por (37).

$$\frac{1}{s} \rightarrow \frac{Tz^{-1}}{1-z^{-1}} \quad (35)$$

$$\frac{U(z)}{E(z)} = \frac{4-3,5z^{-1}}{1-z^{-1}} \quad (36)$$

$$u(n) = u(n-1) - 3,5e(n-1) + 4e(n) \quad (37)$$

A Figura 10 contém gráficos de respostas normalizadas (h/SP) de experimentos realizados na bancada de ensaios utilizando-se um controlador PI com os ganhos citados. Nota-se que as respostas dinâmicas se modificam conforme o ponto de operação do sistema, ou seja, para diferentes valores de *set-point*. Isto já era esperado, uma vez que o processo é não-linear e o controlador possui ganhos fixos.

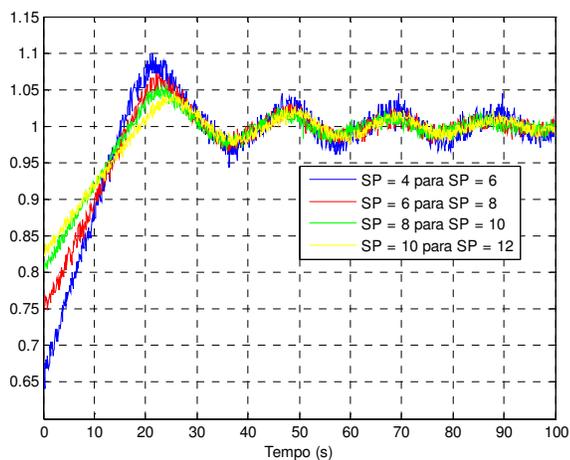


Figura 10 – Respostas do sistema para diferentes pontos de operação.

5.2 Controlador PI com Ganhos Escalonados pela Referência

A primeira abordagem de escalonamento de ganhos utilizada neste trabalho emprega ganhos variáveis escalonados pela informação da referência de entrada (ASTRÖM e WITTENMARK, 1989). A idéia consiste em realizar ensaios em uma malha de controle convencional, utilizando-se ganhos distintos para diferentes faixas de operação do sistema de modo a manter as especificações desejadas (máximo pico menor ou igual a 5% e tempo de acomodação em torno de 60 segundos). A Tabela 2 mostra os valores obtidos após alguns ensaios realizados. A Figura 11 ilustra as variações dos ganhos resultantes (objetivando-se manter as características de resposta desejadas em diferentes pontos de operação do sistema).

Tabela 2 – Ganhos para diversos pontos de operação.

SP	K_p	K_i
4	10	25
6	10	25
8	8	7
10	6	8
12	4	5
14	4	5

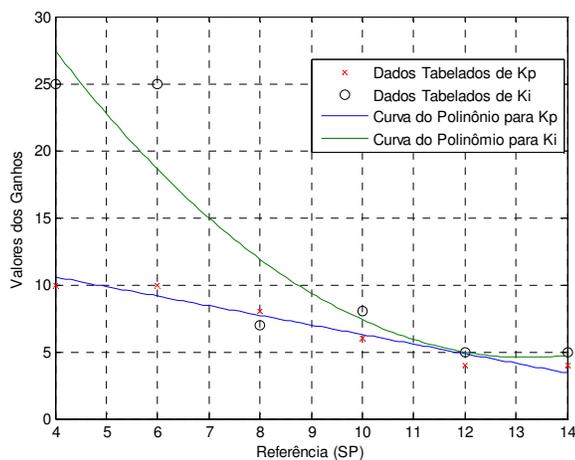


Figura 11 – Gráfico com as informações dos ganhos variáveis.

Com os dados da Tabela 1 foram obtidas as funções de interpolação (38) e (39) para os ganhos do controlador. A equação de diferenças correspondente é dada por (40). Os resultados de ensaios com a estrutura de ganhos escalonados pela referência de entrada serão mostrados no próximo capítulo.

$$K_p = -0,7143sp + 13,4286 \quad (38)$$

$$K_i = 0,2976sp^2 - 7,0929sp + 51,5143 \quad (39)$$

$$u(n) = u(n-1) + (K_i \cdot T - K_p) e(n-1) + K_p \cdot e(n) \quad (40)$$

5.3 Controlador PI com Ganhos Escalonados via Dados de Saída

Técnicas de escalonamento de ganhos são classificadas como um tipo de sistema de controle adaptativo. A questão principal desta modalidade de sistema de controle é a determinação das variáveis que possam ser utilizadas neste procedimento. As variáveis devem refletir as condições de operação da planta, e a seleção das mesmas nem sempre é trivial. O ideal é obter expressões simples que relacionam os ganhos do controlador com as variáveis de escalonamento.

A segunda abordagem realizada neste trabalho utiliza a informação de saída do sistema, e consiste de um método apresentado em ASTRÖM e WITTENMARK (1989). O modelo adotado para o processo de nível é representado pelas expressões (1), (2) e (3). O modelo contínuo do controlador PI é dado por (41) ou (42), em que o parâmetro K representa o ganho proporcional e T_i define a constante de integração do controlador.

$$u(t) = K[e(t) + \frac{1}{T_i} \int e(t) dt] \quad (41)$$

$$C(s) = \frac{U(s)}{E(s)} = K(1 + \frac{1}{sT_i}) \quad (42)$$

Será admitido que a função de transferência em malha fechada do sistema de controle é aproximada por (43), em que a variável R representa a referência (*set-point*) da malha de controle, Y simboliza a variável de saída (nível), sendo o erro da malha definido como $E = R - Y$. Assim, os parâmetros do controlador ficam definidos por (44) e (45) (ASTRÖM e WITTENMARK, 1989).

$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1 + C(s)G(s)} = \frac{\omega^2}{s^2 + 2\xi\omega s + \omega^2} \quad (43)$$

$$K = K_p = \frac{2\xi\omega - \alpha}{\beta} \quad (44)$$

$$T_i = \frac{2\xi\omega - \alpha}{\omega^2} \quad (45)$$

Substituindo-se as expressões para α e β mostradas em (3) em (44) e a expressão de α em (45), e considerando $\alpha \ll 2\xi\omega$ a ponto de poder ser desprezado, obtém-se as seguintes expressões:

$$K = K_p = 2\xi\omega A(h) \quad (46)$$

$$T_i = \frac{2\xi}{\omega} \quad (47)$$

Com o objetivo de se cumprir as metas desejadas (máximo pico menor ou igual a 5% e tempo de acomodação em torno de 60 segundos), sejam as especificações de desempenho: $\omega = 0,2238$ rad/s e $\xi = 0,7448$. Substituindo estes valores nas expressões dos ganhos definidos anteriormente, e considerando o diâmetro do tanque igual a 17 cm, obtém-se (48) e (49), em que $h = Y$ consiste na informação da saída do sistema controlado. O termo dh no desenvolvimento da expressão do ganho K_p é a área $A(h)$ obtida ao se projetar o volume de líquido sobre uma superfície plana.

$$K_p = 2\xi\omega A(h) = 2\xi\omega dh = 2(0,7448)(0,2238)(17)(h) = 5,6678h \quad (48)$$

$$K_i = \frac{K_p}{T_i} = \frac{\omega}{2\xi} K_p = \frac{0,2238}{(2)(0,7448)} (5,6678)h = 0,8527h \quad (49)$$

Os resultados de ensaios da malha de controle com um controlador PI de ganhos escalonados pela informação de saída do sistema serão mostrados no próximo capítulo.

5.4 Controlador *Fuzzy*

A terceira estratégia empregada neste trabalho utiliza um controlador *fuzzy* cuja estrutura está ilustrada na Figura 12. As informações de entrada do sistema nebuloso são o valor do erro da malha de controle e o da sua respectiva integração numérica. A informação de saída do controlador é o valor inferido das regras *fuzzy*. As mesmas são obtidas a partir de dados de ensaio da malha de controle com modificações de ganhos de um controlador convencional (para diversos valores de referência de entrada), ou então com dados já estabelecidos de um controlador com escalonamento de ganhos. Dados provenientes de ensaios reais (mostrados na Figura 13) foram utilizados para gerar as funções de pertinência, os coeficientes e as regras de um modelo Takagi-Sugeno para realizar o controlador *fuzzy*.

Foi utilizado o *toolbox* “*Anfis*” do *Matlab* para esta finalidade (*Fuzzy Logic Toolbox*, 2004). O mesmo utiliza um algoritmo de rede neural para obtenção dos parâmetros das funções de pertinência, dos coeficientes das funções polinomiais e das regras *fuzzy* do modelo Takagi-Sugeno. O ANEXO 2 ilustra os comandos utilizados para a obtenção do modelo.

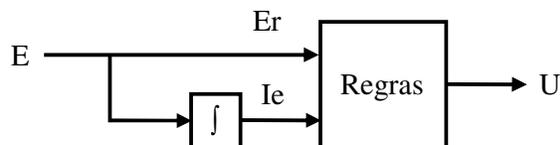


Figura 12 – Controlador *fuzzy*.

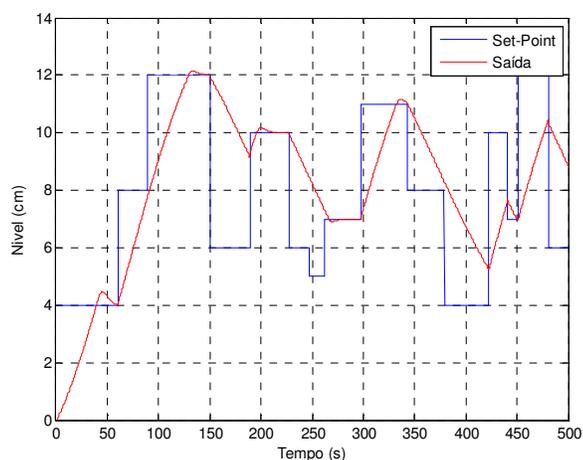


Figura 13 – Dados de ensaios práticos.

As Figuras 14 e 15 ilustram, respectivamente, as funções de pertinência obtidas com o *toolbox* de lógica *fuzzy* do *Matlab* para as informações do erro da malha de controle e da sua integração. As mesmas foram obtidas utilizando-se a função *plotmf* do *Matlab*. Essa função apresenta todas as funções de pertinência de uma dada variável e possui a seguinte sintaxe: *plotmf(sist_fuzzy, tipo da variável, índice da variável)*. No lugar do parâmetro *sist_fuzzy* foi utilizado o sistema *fuzzy* obtido com o *Anfis*. Em *tipo da variável* foi colocado ‘*input*’, uma vez que se trata de uma variável de entrada. Já em *índice da variável* foi usado o número 1 para o Erro e o 2 para a Integral do Erro, uma vez que, no sistema em

questão, a Entrada 1 é o Erro e a Entrada 2 é a sua integração. As funções de pertinência das variáveis de entrada do sistema *fuzzy* foram definidas como gaussianas.

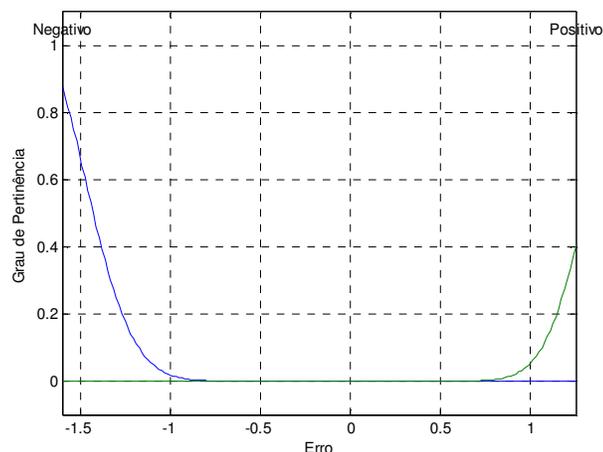


Figura 14 – Funções de pertinência para a entrada relativa ao erro.

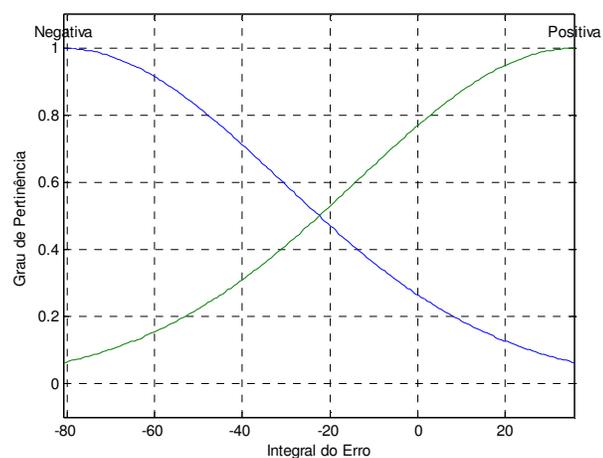


Figura 15 – Funções de pertinência para a entrada relativa à integral do erro.

A Figura 16 ilustra a superfície relacionada com a ação de controle (U) de acordo com valores de informações do erro e da sua integração. Essa superfície é gerada utilizando-se a função *gensurf(sist_fuzzy)* do *Matlab*. Essa função gera uma superfície para um dado sistema *fuzzy* utilizando as duas primeiras entradas e a primeira saída. A Figura 17 compara a ação de controle simulada do controlador *fuzzy* com a real de um controlador PI com escalonamento de ganhos utilizado no ensaio, que gerou os dados utilizados no *Anfis* posteriormente. As Figuras 18 e 19

mostram resultados de medidas reais do erro e da sua integração, respectivamente, e os valores estimados em relação às informações simuladas de entrada e saída do controlador *fuzzy* resultante. O deslocamento (*offset*) que aparece na resposta do modelo *fuzzy* na Figura 19 é devido à integração numérica da simulação, e no sistema real este efeito é corrigido pela realimentação da malha de controle resultante. O modelo difuso mapeou bem as relações entre suas variáveis de entrada e saída, e deste modo espera-se que na realização real da malha de controle o controlador nebuloso apresente um bom desempenho.

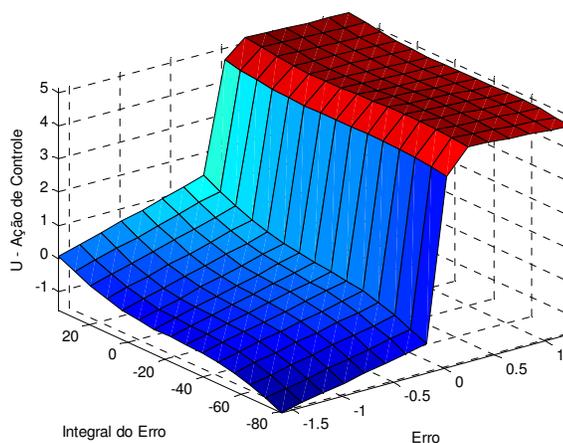


Figura 16 – Superfície de controle.

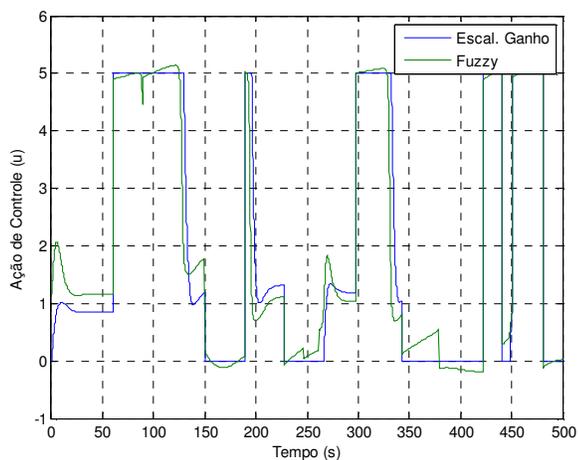


Figura 17 – Dados de medidas reais e do modelo *fuzzy* da variável de comando.

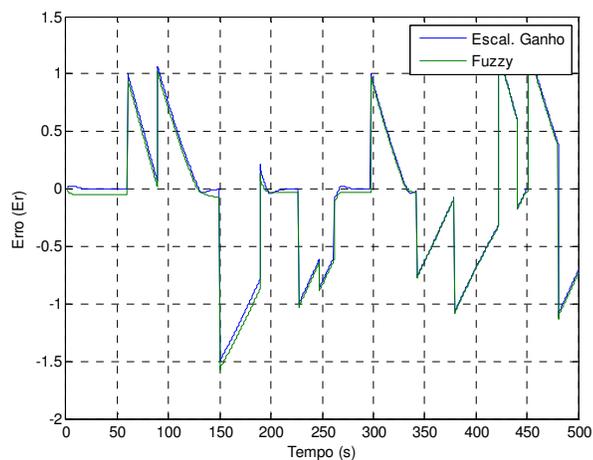


Figura 18 – Dados de medidas reais e do modelo *fuzzy* da variável erro.

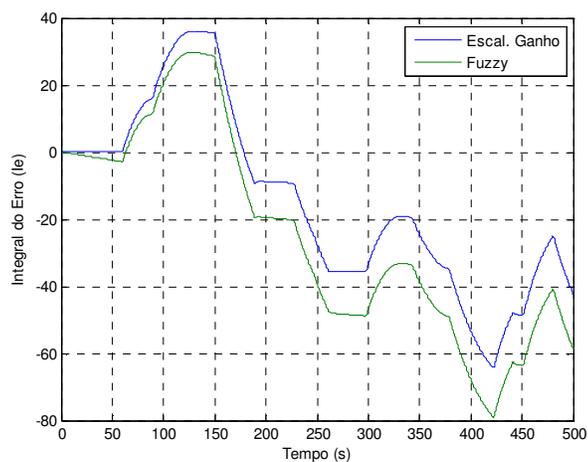


Figura 19 – Dados de medidas reais e do modelo *fuzzy* da integração do erro.

As regras resultantes e os coeficientes das funções polinomiais do modelo *fuzzy* estão relacionados a seguir:

Regras (Er = Erro e Ie = Integração do Erro):

- 1) Se $Er = \text{Negativo}$ e $Ie = \text{Negativa}$ então $U = \text{MuitoBaixo}$;
- 2) Se $Er = \text{Negativo}$ e $Ie = \text{Positiva}$ então $U = \text{Baixo}$;
- 3) Se $Er = \text{Positivo}$ e $Ie = \text{Negativa}$ então $U = \text{Alto}$;
- 4) Se $Er = \text{Positivo}$ e $Ie = \text{Positiva}$ então $U = \text{MuitoAlto}$.

Funções Polinomiais:

- 1) $f_1(Er, Ie) = 2,422 + 0,6018 \cdot Er + 0,03503 \cdot Ie$
- 2) $f_2(Er, Ie) = -0,03486 + 0,8918 \cdot Er + 0,03790 \cdot Ie$
- 3) $f_3(Er, Ie) = 5,792 - 0,2232 \cdot Er + 0,008929 \cdot Ie$
- 4) $f_4(Er, Ie) = 4,68 - 0,08374 \cdot Er + 0,01179 \cdot Ie$

As funções polinomiais mostradas anteriormente representam a resposta fornecida por cada regra. Estas respostas serão usadas no cálculo da resposta global gerada pelo controlador *fuzzy*.

Como exemplo dos processamentos realizados pelo controlador, seja um valor de entrada de Erro $Er = 1$ e da sua Integração $Ie = -40$. Substituindo-se esses valores nas expressões anteriores, obtém-se:

$$f_1 = 1,6226; \quad f_2 = -0,6591;$$

$$f_3 = 5,2116; \quad f_4 = 4,1247.$$

Os valores *fuzzyficados* correspondentes são:

$$Er: \quad \text{Negativo} = e^{-\left(\frac{Er+1,733}{0,2599}\right)^2} = 9,48x10^{-49}; \quad \text{Positivo} = e^{-\left(\frac{Er-1,572}{0,2367}\right)^2} = 2,91x10^{-3};$$

$$Ie: \quad \text{Negativa} = e^{-\left(\frac{Ie+81}{49,73}\right)^2} = 0,51; \quad \text{Positiva} = e^{-\left(\frac{Ie-36,1}{43,73}\right)^2} = 4,84x10^{-2}.$$

Utilizando-se o operador produto nos antecedentes das regras, têm-se:

$$w_1 = 9,48x10^{-49} \cdot 0,51 = 4,83x10^{-49};$$

$$w_2 = 9,48x10^{-49} \cdot 4,84x10^{-2} = 4,59x10^{-50};$$

$$w_3 = 2,91x10^{-3} \cdot 0,51 = 1,48x10^{-3};$$

$$w_4 = 2,91x10^{-3} \cdot 4,84x10^{-2} = 1,41x10^{-4}.$$

O valor *defuzzyficado* correspondente à variável de comando está exemplificado abaixo:

$$Saída = \frac{(1,6226) \cdot (4,83 \times 10^{-49}) + (-0,6591) \cdot (4,59 \times 10^{-50}) + (5,2116) \cdot (1,48 \times 10^{-3}) + (4,1247) \cdot (1,41 \times 10^{-4})}{4,83 \times 10^{-49} + 4,59 \times 10^{-50} + 1,48 \times 10^{-3} + 1,41 \times 10^{-4}} = 5,11 [V]$$

Dessa forma, quando $Er = 1$ e $Ie = -40$, uma tensão elétrica de 5,11 V será aplicada na servo-bomba elétrica. Na verdade, a tensão aplicada será igual a 5 V, uma vez que o intervalo de tensão é de 0 V a 5 V. Os resultados de ensaios reais com o controlador *fuzzy* obtido serão mostrados no próximo capítulo.

6. RESULTADOS

Nesta seção serão apresentados resultados reais relacionados com ensaios práticos no processo em escala reduzida e também de simulações numéricas.

6.1 Controlador PI com Ganhos Fixos

A Figura 20 apresenta resultados de simulações numéricas que utilizam a função do controlador PI de ganhos fixos e o modelo *NARMAX* do processo empregado. Os gráficos das respostas estão normalizados (h/SP). Os resultados obtidos são similares aos relacionados aos ensaios reais mostrados na Figura 10, indicando que a modelagem do sistema está adequada.

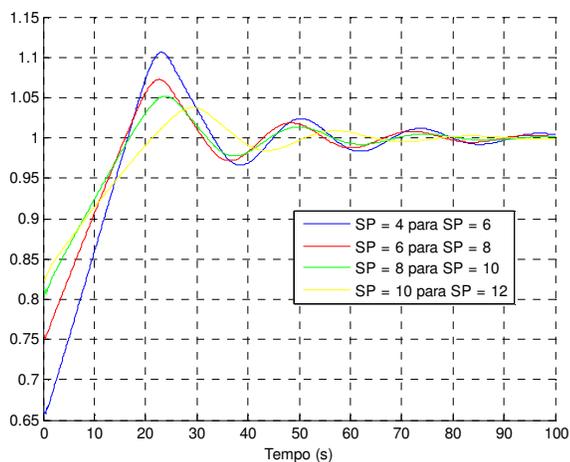


Figura 20 – Resultados de simulações com controlador PI convencional.

6.2 Controlador PI com Ganhos Escalonados pela Referência

A Figura 21 apresenta respostas do sistema real que utiliza agora um controlador PI com ganhos escalonados por intermédio das informações da referência de entrada (SP) da malha de controle. As respostas dinâmicas para diferentes pontos de operação do sistema apresentam características semelhantes, diferentemente das respostas (Figura 10 e Figura 20) com o controlador convencional de ganhos fixos.

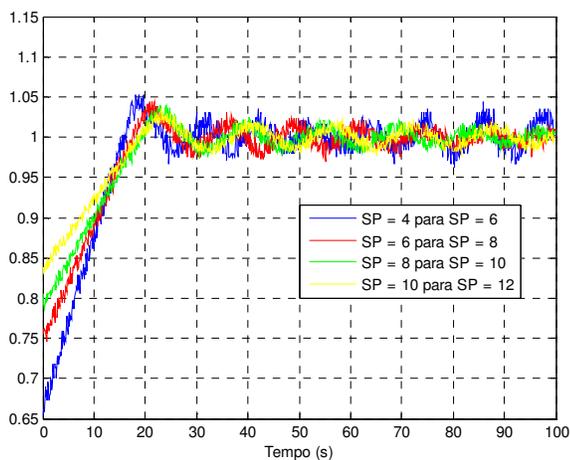


Figura 21 – Resultados reais com PI de ganhos escalonados via referência.

A Figura 22 apresenta resultados de simulações no contexto citado. Novamente, os resultados são coerentes com os resultados práticos.

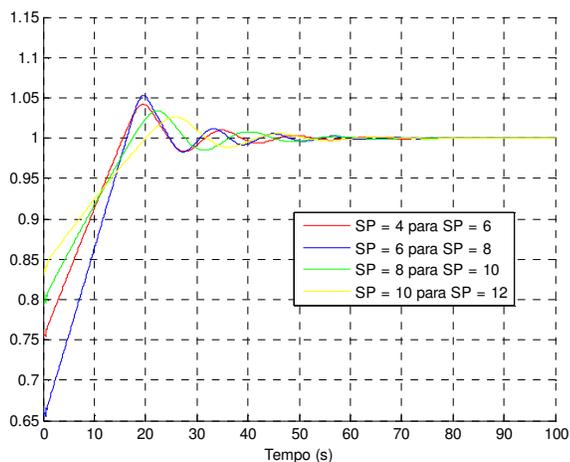


Figura 22 – Simulações com escalonamento via referência de entrada.

6.3 Controlador PI com Ganhos Escalonados via Dados de Saída

A Figura 23 mostra as respostas reais de um controlador PI que faz uso da informação de saída do sistema para atualizar os valores de seus ganhos. Nota-se uma melhoria nas respostas dinâmicas obtidas, que apresentaram menores tempos de acomodação em relação às estratégias anteriores.

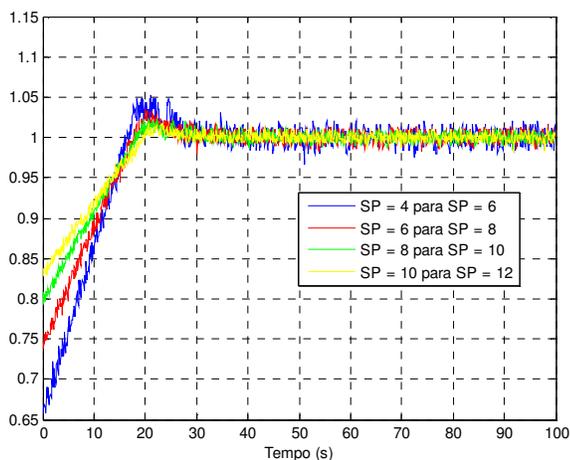


Figura 23 – Resultados reais com escalonamento de ganhos via dados de saída.

A Figura 24 apresenta simulações com o controlador PI que usa informação da saída do sistema. Os resultados estão coerentes com os dados experimentais.

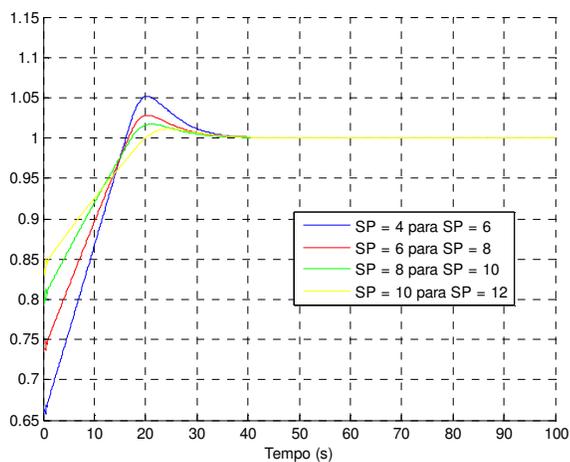


Figura 24 – Resultados de simulações com escalonamento de ganhos via saída.

6.4 Controlador *Fuzzy*

A Figura 25 apresenta os resultados de ensaios reais com o controlador *fuzzy*. Observa-se que a malha de controle resultante apresentou menores valores de máximo pico e tempo de acomodação. A explicação pode ser atribuída ao fato de

que sistemas *fuzzy* são aproximadores universais de funções. O modelo difuso resultante mapeou bem os efeitos das alterações dos ganhos resultantes da etapa de treinamento (que utilizou dados reais de medições experimentais realizadas na bancada de ensaio).

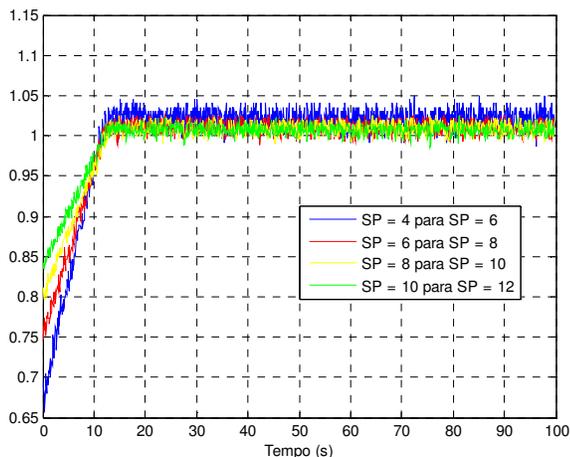


Figura 25 – Resultados de ensaios reais com controlador *fuzzy*.

A Figura 26 apresenta simulações com o controlador *fuzzy*. Elas reproduzem com razoável precisão os ensaios da Figura 22, embora apresentem certo erro em regime permanente que pode ser atribuído a problemas numéricos da implementação computacional da simulação.

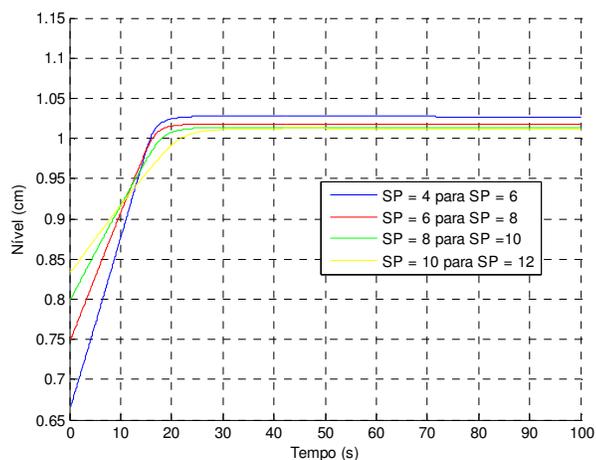


Figura 26 – Resultados de simulações com controlador *fuzzy*.

No ANEXO 3 desta dissertação encontra-se o programa de simulação da malha de controle com o controlador *fuzzy*. No ANEXO 4 encontra-se o programa que implementa a malha de controle real do mesmo. O programa em questão foi realizado no ambiente *LabView*. O aplicativo grava os dados dos ensaios em um arquivo de planilha (*Excel*) que, posteriormente, pode ser lido no ambiente *Matlab* para gerar os gráficos dos resultados dos experimentos realizados.

O ANEXO 5 contém o programa aplicativo referente à abordagem de escalonamento de ganhos via informações da referência de entrada da malha de controle. E no ANEXO 6 tem-se o programa relativo à técnica de escalonamento de ganhos via informações da saída do processo controlado.

Pode-se afirmar que o algoritmo do controlador *fuzzy* exige maiores recursos computacionais comparado às metodologias de escalonamento de ganhos. Entretanto, com os recursos de *hardware* e *software* disponíveis atualmente, não existem grandes dificuldades em implementações práticas de algoritmos desta natureza.

A Tabela 3 compara o desempenho dos controladores com escalonamento de ganhos (referência e saída do sistema) e do controlador *fuzzy*.

Tabela 3 – Comparação entre controladores (Referência = 8 cm).

Controlador	Máximo Pico (M_p) - (%)	Tempo de Acomodação (T_a) - (s)
Proporcional + Integral (PI) com escalonamento pela referência do sistema	4,45	25,2
Proporcional + Integral (PI) com escalonamento pela saída do sistema	3,42	20,5
Proporcional + Integral (PI) <i>fuzzy</i>	2,52	10,7

7. CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foram avaliadas malhas de controle que utilizaram técnicas de escalonamento de ganhos e lógica *fuzzy*. Como exemplo de aplicação foi empregado um sistema de controle de nível. Os resultados obtidos mediante simulações computacionais e desenvolvimentos experimentais se mostraram bem promissores no contexto empregado, o que sugere aplicações igualmente adequadas a outros tipos de sistemas reais. É bem provável que, em plantas industriais de maiores complexidades, as dificuldades a serem transpostas serão superiores. Mesmo assim, as técnicas apresentadas neste trabalho contribuem no sentido de indicar algumas opções de estratégias de controle, que podem ser alternativas interessantes face à utilização de malhas de controle convencionais.

Com relação à lógica *fuzzy*, é possível comprovar que se trata de uma ferramenta bastante poderosa para a engenharia de controle, embora possa demandar mais recursos computacionais, e até mesmo conhecimentos complementares dos projetistas e operadores de sistemas industriais.

As técnicas de escalonamento de ganhos, por sua vez, podem constituir alternativas bastante viáveis de serem implementadas. Elas não demandam muitos recursos computacionais e/ou conhecimentos amplos sobre o tema.

Como trabalhos futuros são pretendidas implementações das abordagens estudadas em sistemas embarcados, ao invés de se utilizar um computador e um sistema de aquisição de dados como plataforma de desenvolvimentos e testes. Também se planeja o projeto de sistemas híbridos, utilizando-se uma combinação de lógica *fuzzy* com outras técnicas de inteligência artificial, como algoritmos genéticos, redes neurais artificiais, etc.

REFERÊNCIAS BIBLIOGRÁFICAS

AGUIRRE, Luiz Antonio. **Introdução à Identificação de Sistemas**, 2ª Edição. Belo Horizonte: Editora UFMG, 2004.

AGUIRRE, Luiz Antonio; RODRIGUES, Giovani G.; JÁCOME, R. F. Identificação de Sistemas Não-Lineares Utilizando Modelos Narmax Polinomiais – Uma Revisão e Novos Resultados. **Revista Controle e Automação**, v. 9, n. 2, maio, jun., jul. e agosto de 1998. Disponível em <http://www.fee.unicamp.br/revista_sba/vol9/V9p90.pdf>. Acesso em: 26 jan. 2009.

ASTRÖM, Karl Johan; WITTENMARK, Björn. Adaptive Control. In: **Design of Gain Scheduling Regulators**. Nova Iorque: Addison-Wesley, 1989. cap. 9, p.346-349.

BHAMBHANI, Varsha; CHEN, YangQuan. Experimental Study of Fractional Order Proportional Integral (FOPI) Controller for Water Level Control. In: **47th IEEE Conference on Decision and Control**, Cancun, Mexico, 9-11 dez. 2008. Disponível em: <<http://ieeexplore.ieee.org/articleSale/Sarticle.jsp?arnumber=4739341>>. Acesso em: 29 jun. 2009.

CHATRATTANAWUTH, Wicharn; SUKSARIWATTANAGUL, Napatpong; BENJANARASUTH, Taworn; NGAMWIWIT, Jongkol. Fuzzy I-PD Controller for Level Control. In: **SICE-ICASE International Joint Conference 2006**, Busan, 18-21 out. 2006. Disponível em: <<http://ieeexplore.ieee.org/articleSale/Sarticle.jsp?arnumber=4108583>>. Acesso em: 29 jun. 2009.

COELHO, Leandro dos Santos; ALMEIDA, Otacílio da M. Projeto e Estudo de Caso da Implementação de um Sistema de Controle Nebuloso. **Revista Controle & Automação**, Campinas, v. 14, n. 1, mar. 2003. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-17592003000100003>. Acesso em: 28 jun. 2009.

DRIANKOV, Dimiter; HELLENDORRN, Hans; REINFRANK, Michael. **An Introduction to Fuzzy Control**: 2nd Edition. London: Springer, 1996.

DUSSUD, Michel; GALICHET, Sylvie; FOULLOY, Laurent P. Application of Fuzzy Logic Control for Continuous Casting Mold Level Control. In: **IEEE Transactions on Control Systems Technology**, vol. 6, n. 2, mar. 1998. Disponível em: <<http://www.ieee.org/portal/site>>. Acesso em: 29 jun. 2009.

Fuzzy Logic Toolbox for Use With MATLAB. User's Guide Version 2, Natick, jun. 2004.

HECKENTHALER, Thomas; ENGELL, Sebastian. Approximately Time-Optimal Fuzzy Control of a Two-Tank System. **IEEE Control Systems Magazine**, Ann Arbor, Vol. 29, Issue 5, 2009. Disponível em: <<http://ieeexplore.ieee.org/articleSale/Sarticle.jsp?arnumber=291460>>. Acesso em: 29 jun. 2009.

KAYACAN, Erdal; KAYNAK, Okyay. Grey Prediction Based Control of a Non-Linear Liquid Level System Using PID Type Fuzzy Controller. In: **ICM 2006 - IEEE 3rd International Conference on Mechatronics**, Budapeste, 3-5 jul. 2006. Disponível em: <<http://ieeexplore.ieee.org/articleSale/Sarticle.jsp?arnumber=4018376>>. Acesso em: 29 jun. 2009.

KOUHI, Y.; LABIBI, B.; FATEHI, A.; ADLGOSTAR, R. H. and QFT Robust Control Designs for Level Control Plant. **India Conference - INDICON 2008**, Kanpur, vol. 2, p. 388-393, 11-13 dez. 2008. Disponível em: <<http://ieeexplore.ieee.org/articleSale/Sarticle.jsp?arnumber=4768755>>. Acesso em: 29 jun. 2009.

LUNA FILHO, Fernando de Melo; GOSMANN, Hugo Leonardo; BAUCHSPIESS, Adolfo. **Controle Fuzzy para Sistemas de Nível de Líquidos**. In: XIV CONGRESSO BRASILEIRO DE AUTOMÁTICA, set. 2002, Natal. Disponível em: <<http://www.ene.unb.br/adolfo/papers/CBA2002.pdf>>. Acesso em: 28 jun. 2009.

LYUNG, Lennart. **System Identification Toolbox for Use With MATLAB**. User's Guide Version 5, Natick, nov. 2000.

MELO, Gustavo Amaral Ferreira de; BERNARDES, Mariana Costa. **Instrumentação e Controle de uma Maquete de Nível de Líquido com Quatro Tanques Interligados**. Brasília, mar. de 2006. Disponível em: <<http://www.ene.unb.br/~gaborges/orientacoes/tg/pf.mariana.bernardes.gustavo.melo.2005.2.pdf>>. Acesso em: 28 jun. 2009.

OGATA, Katsuhiko. Engenharia de Controle Moderno, 4ª Edição. In: **Modelagem Matemática de Sistemas Dinâmicos**. São Paulo: Prentice Hall, 2003. capítulo 3, p. 45.

PINHEIRO, Carlos A. Murari. **Análise e Projeto de Sistemas de Controle Fuzzy: Uma Abordagem no Domínio da Frequência**. Campinas, fev. 2000. Disponível em: <<http://libdigi.unicamp.br/document/?code=vtls000197797>>. Acesso em: 23/03/2009.

TANSCHUIT, Ricardo. Sistemas Fuzzy. In: **VI Simpósio Brasileiro de Automação Inteligente**. Mini-curso, Bauru, set. 2003. Disponível em: <<http://www.ica.ele.puc-rio.br/Downloads/41/LN-Sistemas%20Fuzzy.pdf>>. Acesso em: 28 jun. 2009.

ANEXO 1 – PROGRAMA DO MODELO *NARMAX*

```

%Estimação de modelo NARMAX

n=6001; %Numero de amostras
m=n+3; %Indexador

%Dados para Identificação
load E:\MATLAB6p5\work\le_varios_sp4.txt;%carrega arquivo
in=5*e_varios_sp4(1:m,6); %Entrada
out=e_varios_sp4(1:m,2); %Saida

%Regressores
F(1:n,1)=out(3:n+2); %y(k-1)
F(1:n,2)=out(2:n+1); %y(k-2)
F(1:n,3)=in(2:n+1); %u(k-2)
F(1:n,4)=in(3:n+2); %u(k-1)
for i=1:n
    F(i,5)=out(i+2)^2 * in(i+2); %y(k-1)^2*u(k-1)
    F(i,6)=in(i+2)^2 * in(i); %u(k-1)^2*u(k-3)
end
F(1:n,7)=out(1:n).^3; %y(k-3)^3
for j=1:n
    F(j,8)=out(j+1)*in(j+1); %y(k-2)*u(k-2)
    F(j,9)=out(j)^2*in(j); %y(k-3)^2*u(k-3)
end
Y=out(4:n+3); %y(k)

%Estimação de Parametros por Minimos Quadrados
P = inv(F'*F);
Q = F'*Y;
T = P*Q; %Vetor de Parametros Estimados
Teta = T;

```

```

'Vetor de Parametros Estimados'
format short g
Teta

%Dados para Validaçao
load E:\MATLAB6p5\work\e_varios_sp4.txt;%carrega arquivo
iv=5*e_varios_sp4(1:m,6); %Entrada
ov=e_varios_sp4(1:m,2); %Saida

%Validaçao
S1(1)=4.115776; S1(2)=4.066851; S1(3)=3.944540;
for k=4:m
    S1(k)=T(1)*S1(k-1) + T(2)*S1(k-2) + T(3)*iv(k-2) + T(4)*iv(k-1) + T(5)*(S1(k-1)^2)*iv(k-1) + T(6)*(iv(k-1)^2)*iv(k-3) + T(7)*S1(k-3)^3 + T(8)*S1(k-2)*iv(k-2) + T(9)*(S1(k-3)^2)*iv(k-3);
end
S1=S1'; %Dados de saida simulados
plot(1:m,S1,'b',1:m,ov,'k'); grid; xlabel('Numero de Amostras'); ylabel('y');
legend('Modelo NARMAX','Dados Medidos');
title('Dados Simulados e Medidos');

%Autocorrelaçao do sinal simulado com as medidas de saida
rr=xcorr(S1,ov); %AutoCorrelaçao dos dados simulados e medidas de saida
Crr=corrcoef(S1,ov);
crr=Crr(1,2)*100; %Coeficiente de correlaçao entre os dados simulados e os
medidos
'Coeficiente de Correlaçao do Modelo NARMAX'
crr
figure;
plot(-6003:6003,rr); grid; xlabel('Numero de Amostras'); ylabel('AutoCorrelaçao do
sinal medido com modelo NARMAX');
title('Autocorrelaçao');

```

ANEXO 2 – PROGRAMA DE OBTENÇÃO DO MODELO *FUZZY*

```
clear all;
```

```
T = 0.1;
```

```
N = 5000;
```

```
load E:\MATLAB6p5\work\anfis_varios_sp.txt;%carrega arquivo
```

```
SP = anfis_varios_sp;%Vetor de Set-Points
```

```
li = 0;
```

```
Kp = 0; %ganho proporcional
```

```
Ki = 0; %ganho integral
```

```
%valores iniciais de tempo, saida, ação de controle, erro e integral do erro.
```

```
t(1)=0;
```

```
t(2)=0.1;
```

```
t(3)=0.2;
```

```
y(1)=4;
```

```
y(2)=4;
```

```
y(3)=4;
```

```
u(1)=0;
```

```
u(2)=0;
```

```
u(3)=0;
```

```
Er(1)=0;
```

```
Er(2)=0;
```

```
Er(3)=0;
```

```
le(1) = 0;
```

```
le(2) = 0;
```

```
le(3) = 0;
```

```

%Vetor teta com os parametros do modelo nao-linear
Teta = [0.50097 0.49709 0.015014 -0.0093045 0.00024623 -0.00013808 8.0467e-
006 0.0010061 -0.00031571];
%Dados para a obtenção do Controlador Fuzzy
for n=4:N
    t(n)=n*T;
    Er(n) = (SP(n) - y(n-1))/4; %divisao por 4 devido a caracteristica do transdutor
    le(n) = le(n-1) + Er(n)*T;
    Kp = 5.6673*y(n-1); % Escalonamento dos ganhos
    Ki = 0.8515*y(n-1);
    P = Kp*Er(n); % Parte Proporcional
    li = li + Ki*Er(n)*T; % Parte Integral
    if (li<=0)
        li=0;
    end
    if (li>=5)
        li=5;
    end
    u(n)=P + li;
% Limitação na faixa de 0 a 5 devido ao conversor D/A do sistema
    if u(n)<=0
        u(n)=0;
    end
    if u(n)>=5
        u(n)=5;
    end
    y(n) = Teta(1)*y(n-1) + Teta(2)*y(n-2) + Teta(3)*u(n-2) + Teta(4)*u(n-1) +
Teta(5)*y(n-1)*y(n-1)*u(n-1) + Teta(6)*u(n-1)*u(n-1)*u(n-3) + Teta(7)*y(n-3)*y(n-
3)*y(n-3) + Teta(8)*y(n-2)*u(n-2) + Teta(9)*y(n-3)*y(n-3)*u(n-3);
end

plot(t,SP,t,y,'r'); % Gráficos
grid;
axis([0 500 0 14]);

```

```
xlabel('Tempo (s)');
ylabel('Nivel (cm)');
legend('Set-Point','Saída')

%Obtenção do Controlador Fuzzy
Ent = [Er' le'];%Dados de entrada
in_fis = genfis1([Ent u'],2,'gaussmf');
out_fis = anfis([Ent u'],in_fis,1000); % Geração do modelo Fuzzy
su = size(u);
np = su(2);
for n=1:np
    Ent1 = [Er(n); le(n)];
    uf = evalfis(Ent1,out_fis);
    Vuf(n) = uf;
end
figure(2)
plot(t,u,t,Vuf);
legend('Resultados do Modelo Fuzzy');
grid;
```

ANEXO 3 – PROGRAMA DE SIMULAÇÃO DO CONTROLE *FUZZY*

```
clear all;
```

```
T = 0.1;
```

```
N = 5000;
```

```
load E:\MATLAB6p5\work\anfis_varios_sp.txt;
```

```
SP = anfis_varios_sp;
```

```
%valores iniciais de tempo, saída, ação de controle, erro e integral do erro.
```

```
le(1) = 0;
```

```
le(2) = 0;
```

```
le(3) = 0;
```

```
t(1)=0.1;
```

```
t(2)=0.2;
```

```
t(3)=0.3;
```

```
y(1)=4;
```

```
y(2)=4;
```

```
y(3)=4;
```

```
Er(1)=0;
```

```
Er(2)=0;
```

```
Er(3)=0;
```

```
u(1)=0;
```

```
u(2)=0;
```

```
u(3)=0;
```

```
fismat = readfis('cont_fuzzy.fis');
```

```

%Vetor teta com os parametros do modelo nao-linear
Teta = [0.50097 0.49709 0.015014 -0.0093045 0.00024623 -0.00013808 8.0467e-
006 0.0010061 -0.00031571];

for n=4:N
    t(n)=n*T;
    Er(n) = (SP(n) - y(n-1))/4; %divisao por 4 devido a caracteristica do conversor
    le(n) = le(n-1) + Er(n)*T;
    Ent = [Er(n); le(n)];
    Vu = evalfis(Ent,fismat);
% limitacao na faixa de 0 a 5
    if Vu<=0
        Vu=0;
    end
    if Vu>=5
        Vu=5;
    end
    u(n)=Vu;
    y(n) = Teta(1)*y(n-1) + Teta(2)*y(n-2) + Teta(3)*u(n-2) + Teta(4)*u(n-1) +
Teta(5)*y(n-1)*y(n-1)*u(n-1) + Teta(6)*u(n-1)*u(n-1)*u(n-3) + Teta(7)*y(n-3)*y(n-
3)*y(n-3) + Teta(8)*y(n-2)*u(n-2) + Teta(9)*y(n-3)*y(n-3)*u(n-3);
end
plot(t,y,'g');
grid;
axis([0 N/10 0 14]);
xlabel('Tempo (s)');
ylabel('Nivel (cm)');

```

ANEXO 4 – SUB-ROTINA DO *LABVIEW* DO CONTROLADOR *FUZZY*

```

if(i<1) {li=0;}
li=li+Er*T;
x1=Er;
x2=li;
X1a = -1.6;
X1b =1.254;
X1c = -1.6;
X1d = 1.254;
X2a = -81;
X2b = 36.1;
X2c = -81;
X2d = 36.1;
cx1ab = -1.733;
cx1cd = 1.572;
desvx1ab = 0.2599;
desvx1cd = 0.2367;
cx2ab = -81;
cx2cd = 36.1;
desvx2ab = 49.73;
desvx2cd = 49.73;
% Fuzzyficacao.
% Funções de pertinencia gaussianas.
ux1ab = exp(-((x1 - cx1ab)*(x1 - cx1ab)/desvx1ab*desvx1ab));
ux1cd = exp(-((x1 - cx1cd)*(x1 - cx1cd)/desvx1cd*desvx1cd));
ux2ab = exp(-((x2 - cx2ab)*(x2 - cx2ab)/desvx2ab*desvx2ab));
ux2cd = exp(-((x2 - cx2cd)*(x2 - cx2cd)/desvx2cd*desvx2cd));

Y1=0;
Y2=0;
Y3=0;
Y4=0;

```

```
if ((x1 >= X1a) && (x1 <= X1b) && (x2 >= X2a) && (x2 >= X2b)){ Y1=0.6018*x1 +
0.03503*x2 + 2.422;}
```

```
if ((x1 >= X1a) && (x1 <= X1b) && (x2 >= X2c) && (x2 <= X2d)){
Y2=0.8918*x1 + 0.03793*x2 - 0.03486;}
```

```
if ((x1 >= X1c) && (x1 <= X1d) && (x2 >= X2a) && (x2 <= X2b)){
Y3=-0.2232*x1 + 0.008929*x2 + 5.791;}
```

```
if ((x1 >= X1c) && (x1 <= X1d) && (x2 >= X2c) && (x2 <= X2d)){
Y4=-0.08374*x1 + 0.01179*x2 + 4.68;}
```

```
% Comoposicao
```

```
r1 = ux1ab * ux2ab;
```

```
r2 = ux1ab * ux2cd;
```

```
r3 = ux1cd * ux2ab;
```

```
r4 = ux1cd * ux2cd;
```

```
% Defuzzificacao
```

```
de = (Y1*r1+Y2*r2+Y3*r3+Y4*r4);
```

```
sur = (r1+r2+r3+r4);
```

```
de = de / sur;
```

```
Vu = de;
```

```
% limitacao na faixa de 0 a 5
```

```
if (Vu<=0){Vu=0;}
```

```
if (Vu>=5){Vu=5;}
```

```
u=Vu; % ação de controle
```

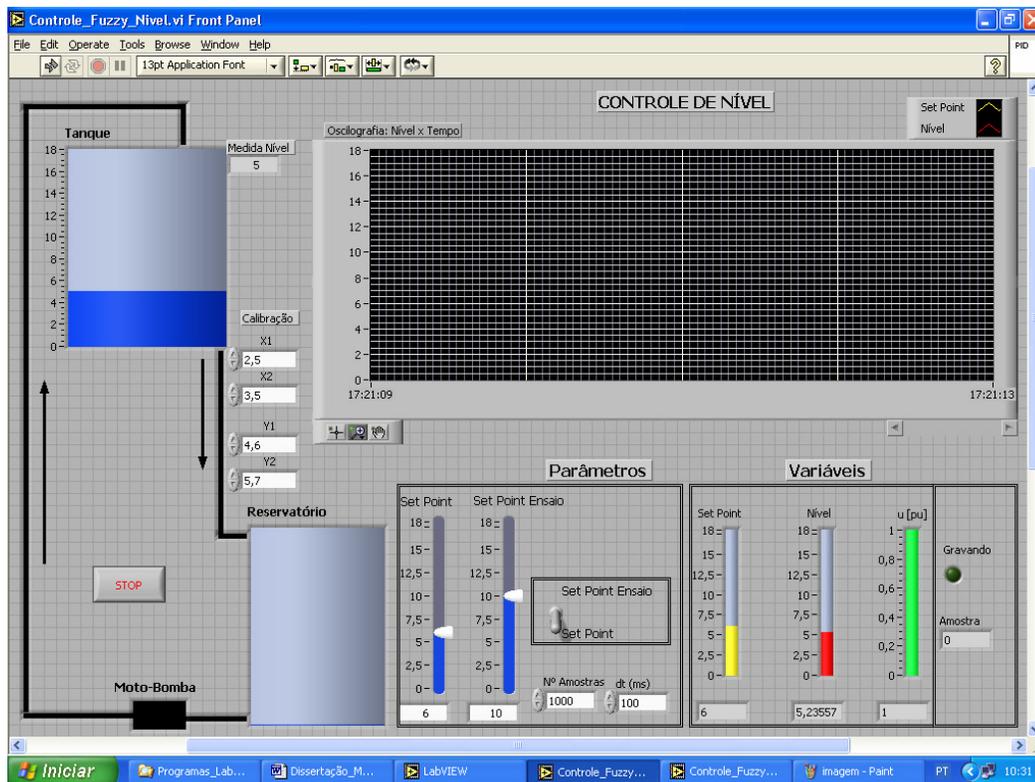


Figura 27 – Interface gráfica do sistema de controle.

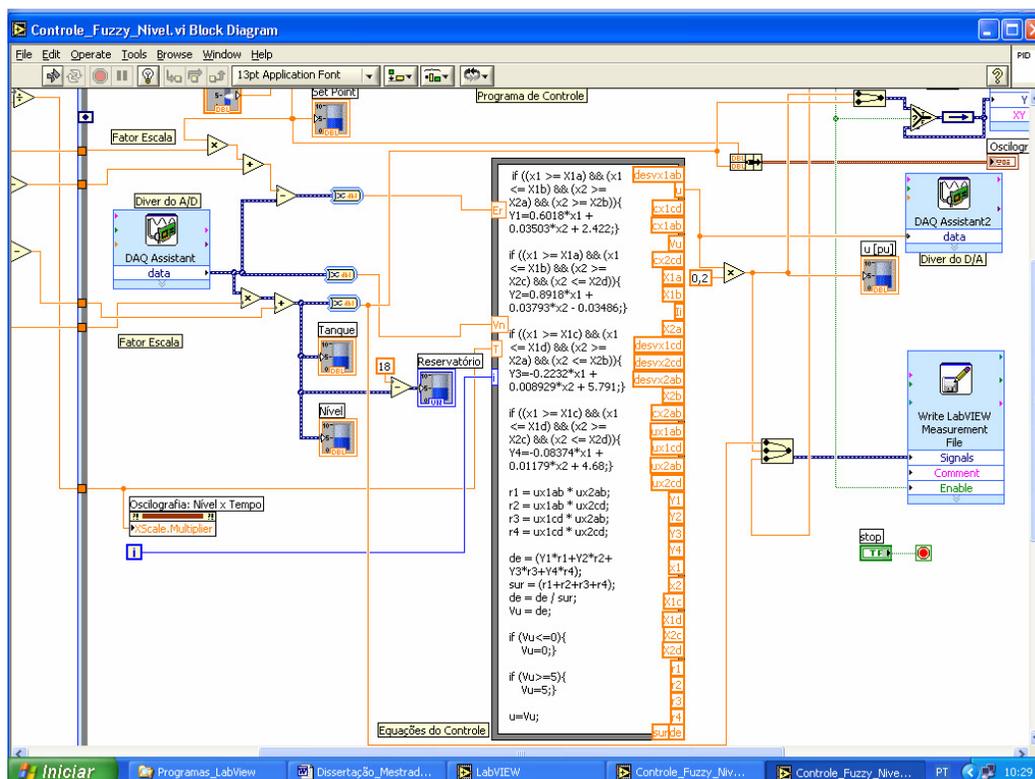


Figura 28 – Realização do controlador *fuzzy*.

ANEXO 5 – SUB-ROTINA PARA ESCALONAMENTO DE GANHOS I

```
If (i<1) { li=0; }
```

```
Kp = -0.7143*SP + 13.4286; %ganho proporcional escalonado pelo SP
```

```
Ki = 0.2976*SP*SP -7.0929*SP + 51.5143; %ganho integral
```

```
li = li+Ki*Er*T;
```

```
If (li<0) li=0;
```

```
If (li>5) li=5;
```

```
u = Kp*Er+li; %ação de controle
```

```
% limitação na faixa de 0 a 5
```

```
if (u<0) u=0;
```

```
if (u>5) u=5;
```

ANEXO 6 – SUB-ROTINA PARA ESCALONAMENTO DE GANHOS II

If ($i < 1$) { $l_i = 0$;}

$K_p = 5.6673 \cdot h$; % ganho proporcional escalonado pela informação do nível

$K_i = 0.8515 \cdot h$; % ganho integral escalonado pela informação do nível

$l_i = l_i + K_i \cdot E_r \cdot T$;

If ($l_i < 0$) $l_i = 0$;

If ($l_i > 5$) $l_i = 5$;

$u = K_p \cdot E_r + l_i$; % ação de controle

% limitação na faixa de 0 a 5

if ($u < 0$) $u = 0$;

if ($u > 5$) $u = 5$;