

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

**Desenvolvimento de um Controlador *Rough* com Ação Proporcional-Integral para um Processo com Característica Não-Linear e Comparações de Desempenho em Relação a um Controlador *Fuzzy* e um Compensador Linear Discreto**

**Juliana Ribas Monteiro**

**Itajubá, Novembro de 2015**

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

**Juliana Ribas Monteiro**

**Desenvolvimento de um Controlador *Rough* com Ação Proporcional-Integral para um Processo com Característica Não-Linear e Comparações de Desempenho em Relação a um Controlador *Fuzzy* e um Compensador Linear Discreto**

**Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica como parte dos requisitos para obtenção do Título de Mestre em Ciências em Engenharia Elétrica.**

**Área de Concentração: Automação e Sistemas Elétricos Industriais**

**Orientador: Prof. Dr. Carlos Alberto Murari Pinheiro  
Co-Orientador: Prof. Dr. Benedito Isaias de Lima Lopes**

**Novembro de 2015  
Itajubá - MG**

As informações são fornecidas pela Biblioteca Mauá - BIM

MONTEIRO, Juliana Ribas

Desenvolvimento de um controlador *rough* com ação proporcional-integral para um processo com característica não-linear e comparações de desempenho em relação a um controlador *fuzzy* e um compensador linear discreto / Juliana Ribas Monteiro.

Itajubá: UNIFEI, 2015.

121p.

Dissertação (mestrado) – Universidade Federal de Itajubá, 2015.

Orientador: Prof. Dr. Carlos Alberto Murari Pinheiro.

Co-Orientador: Prof. Dr. Benedito Isaias de Lima Lopes.

1. Malhas de controle. 2. sistemas não-lineares. 3. controladores discretos, *fuzzy* e aproximados. 4. respostas dinâmicas. 5. tempo de processamento. I. Pinheiro, Carlos Alberto Murari. II. Lopes, Benedito Isaias de Lima. III. Universidade Federal de Itajubá. IV. Título.

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

**Juliana Ribas Monteiro**

**Desenvolvimento de um Controlador *Rough* com Ação Proporcional-Integral para um Processo com Característica Não-Linear e Comparações de Desempenho em Relação a um Controlador *Fuzzy* e um Compensador Linear Discreto**

Dissertação aprovada por banca examinadora em 12 de Novembro de 2015, conferindo ao autor o título de *Mestre em Ciências em Engenharia Elétrica*.

**Banca Examinadora:**

Prof. Dr. Carlos Alberto Murari Pinheiro (Orientador)

Prof. Dr. Benedito Isaias de Lima Lopes (Co-Orientador)

Prof. Dr. Ronaldo Rossi

Prof. Dr. Luís Henrique de Carvalho Ferreira

**Itajubá  
2015**

Aos meus pais.  
À minha querida avó.

## **Agradecimentos**

A Deus, pela força e proteção.

Aos meus pais Luiz Carlos e Maria Inês, pelo incentivo e apoio que sempre me deram.

Aos meus irmãos Maíra e Luiz Fernando, pelo companheirismo e amizade.

Ao meu namorado Eugênio, pelo carinho, paciência e motivação.

Aos meus familiares e amigos, pela companhia mesmo que distante.

Um agradecimento especial aos Professores Pinheiro e Isaias, pela atenção, paciência na solução das dúvidas, pelo tempo dedicado e amizade.

À CAPES, pela bolsa de mestrado.

“O rio atinge os seus objetivos porque aprendeu a contornar seus obstáculos”.

(Anônimo)

## Resumo

Este trabalho tem como objetivo realizar comparações de desempenho de malhas de controle associadas a um processo não-linear, onde serão realizados três tipos de controladores: linear discreto; *fuzzy*; aproximado (*rough*).

Como processo é utilizado uma planta de nível em escala reduzida, e através de ensaios práticos são obtidos dados reais do sistema, possibilitando a obtenção de modelos matemáticos associados: linear e não-linear.

São consideradas malhas de controle com a tarefa de compensar o processo em questão, e a partir de determinadas especificações são estabelecidas leis de controle com esta finalidade. As funções de controle estabelecidas estão relacionadas com os três tipos de controladores citados no primeiro parágrafo. Para efeito de comparação de desempenho das compensações consideradas, são realizadas simulações computacionais com os modelos obtidos e também ensaios práticos em laboratório.

Na realização dos experimentos práticos, utilizou-se uma bancada em laboratório contendo a planta de nível em escala reduzida e um sistema de aquisição de dados acoplado a um computador, realizando a coleta e processamento de dados dos ensaios em tempo real.

Nas respostas obtidas (simulações computacionais e ensaios de laboratório), considerou-se os valores de máxima sobrelevação (*overshoot*), tempo de acomodação (*settling time*) e o tempo de processamento dos algoritmos envolvidos como indicadores de desempenho.

Palavras chave: Malhas de controle, sistemas não-lineares, controladores discretos, *fuzzy* e aproximados, respostas dinâmicas, tempo de processamento.

## **Abstract**

This work aims to conduct control loop performance comparisons associated with a nonlinear process, which will be tested three types of controllers: discrete linear; fuzzy; and rough.

A level system in reduced-scale is used as process, and through practical tests mathematical models (linear and nonlinear) are obtained from the real data of the system.

Control loops are considered with the task of compensating the process in question, and from certain specifications control laws were established for this purpose. The designated control functions are related to the three types of controllers mentioned in the first paragraph. In order to make comparisons of performance in the compensations considered, computational simulations with the models obtained and practical tests in laboratory are performed. In carrying out practical experiments was used a laboratory bench containing the small-scale level process and a data acquisition system coupled to a computer, whose task was to collect the test data and to process real-time control algorithms associated. In the responses (computer simulations and practical tests) were considered the maximum values (overshoot), the settling time and the processing time of the algorithms involved as performance indicators.

Key-words: control loops, nonlinear systems, discrete, fuzzy and rough controllers, dynamic responses, processing time.

## Lista de Abreviaturas e Siglas

A/D - conversor analógico/digital

ARMAX - *autoregressive moving average model with exogenous inputs*

D/A - conversor digital/analógico

FE - Fator de Escala

FPGA - *Field Programmable Gate Array*

IA - Inteligência Artificial

ITAE - *integral of time multiplied by absolute error*

NARMAX - *nonlinear autoregressive moving average model with exogenous inputs*

NARX - *nonlinear autoregressive model with exogenous inputs*

NI - *National Instruments*

PI - Proporcional e Integral

pv - *process variable*

sp - *set point*

USB - *Universal Serial Bus*

## Lista de Símbolos

$A$  - matriz de estado  $n \times n$

$A(z)$  - polinômio do modelo genérico

$a_1$  - coeficiente do modelo ARX

$B$  - matriz de entrada  $n \times r$

$b$  - valor modal das funções de pertinência

$B(z)$  - polinômio do modelo genérico

$b_1$  - coeficiente do modelo ARX

$C_c$  - valor modal das funções de pertinência

$C_i$  - valor modal das funções de pertinência

$c_{in}$  - coeficientes numéricos do modelo baseado em regras

$C_m$  - valor modal das funções de pertinência

$C_n$  - valor modal das funções de pertinência

$C_o$  - valor modal das funções de pertinência

$C(z)$  - função do controlador

$d$  - perturbação externa

$D(z)$  - polinômio da função característica

$D_n^{(i)}$  - regiões de funções de pertinência ou *crisp*

$E$  - entrada

$e$  - erro

$F(z)$  - polinômio de uma função no domínio  $Z$

$F^\ell[.]$  - regressores de  $y(k)$ ,  $u(k)$  e  $e(k)$

$G$  - abreviação de Grande

$g_e$  - fator de escala da variável  $e$

$g_d$  - fator de escala da variável de saída

$g_v$  - fator de escala da variável  $v$

$G(z)$  - Modelo da planta ou processo controlado

$h$  - nível

$\hat{h}$  - nível linearizado em torno de um ponto de operação

$I$  - sistema de informação  
 $I_e$  - integral do erro  
 $J$  - função de custo  
 $K$  - matriz de ganhos  
 $k$  - amostra  
 $K_g$  - ganho do processo  
 $K_i$  - ganho integral  
 $K_p$  - ganho proporcional  
 $\ell$  - grau de não linearidade  
 $M$  - módulo  
 $m$  - conjunto  
 $min$  - mínimo  
 $M_p$  - máximo pico  
 $N$  - abreviação de Negativo  
 $n$  - amostra corrente  
 $NL$  - conjunto associado com a variável  $U$   
 $NM$  - conjunto associado com a variável  $U$   
 $N_1$  - conjunto nebuloso relativo à variável  $E$   
 $N_2$  - conjunto nebuloso relativo à variável  $V$   
 $o_i$  - instância  
 $P$  - abreviação de Positivo  
 $PM$  - conjunto associado com a variável  $U$   
 $PL$  - conjunto associado com a variável  $U$   
 $P_2$  - conjunto nebuloso relativo à variável  $V$   
 $P_1$  - conjunto nebuloso relativo à variável  $E$   
 $P_n^{(m)}$  - conjuntos poliedrais  
 $q_i$  - fluxo de entrada  
 $q_o$  - fluxo de saída  
 $R$  - valor real  
 $r$  - referência ou *set point*

$r_i$  - regras

$s$  - conjuntos difusos

$T$  - tempo de amostragem/varredura

$T^T$  - matriz transposta

$T_a$  - tempo de acomodação

$T_d$  - atraso de tempo

$U$  - ação de comando

$u$  - ação de comando

$u_{Aj}$  - grau de pertinência

$V$  - variável de entrada

$v$  - incertezas ou ruídos

$V_0$  - variável de saída

$W_n$  - frequência natural não amortecida

$x_n$  - variável de entrada

$\underline{x}^{(j)}$  - vetor composto pelos valores de  $x_i$  correspondentes aos limites inferiores

$\overline{x}^{(j)}$  - vetor composto pelos valores de  $x_i$  correspondentes aos limites superiores

$y$  - variável de saída

$y_i^{(k)}$  - vetor composto pelos valores de  $x_i$  correspondentes aos limites inferiores

$y_i^{(m)}$  - vetor composto pelos valores de  $x_i$  correspondentes aos limites superiores

$Z$  ou  $ZE$  - abreviação de Zero

$ZE$  - conjunto associado com a variável  $U$

$Z_1$  - conjunto nebuloso relativo à variável  $E$

$Z_2$  - conjunto nebuloso relativo à variável  $V$

$z_{1,2}$  - polos dominantes

$z^{-1}$  - notação do operador de atraso de transporte

$\alpha_i$  - conjunto *crisp*

$\alpha_j$  - conjunto difuso

$\beta_i$  - conjunto *crisp*

$\beta_j$  - conjunto difuso

$\xi(k)$  - erros ou incertezas de modelagem

$\mu$  - valor de pertinência

$\psi(k - 1)$  - vetor das variáveis regressoras

$\sigma_j$  - conjunto difuso

$\tau_g$  - constante de tempo

$\theta$  - coeficientes do modelo

$\zeta$  - coeficiente de amortecimento

\* - produto

# Sumário

1 Introdução .....	16
2 Revisão Bibliográfica.....	18
3 Revisão de Conceitos .....	22
3.1 Modelagem e Identificação de Sistemas .....	22
3.1.1 Modelo ARX.....	24
3.1.2 Modelo NARMAX .....	26
3.2 Controlador PI Linear Discreto .....	27
3.3 Compensação via Lugar das Raízes.....	28
3.4 Controlador <i>Fuzzy</i> com Ação PI .....	29
3.5 Controlador <i>Rough</i> com Ação PI.....	33
3.5.1 Conversão de Modelos <i>Fuzzy</i> em Modelos <i>Rough</i> .....	34
4 Metodologia e Desenvolvimento .....	39
4.1 Descrição do Processo de Nível.....	39
4.2 Modelos Matemáticos.....	41
4.2.1 Obtenção do Modelo ARX.....	42
4.2.2 Obtenção do Modelo NARMAX .....	43
4.3 Projeto de um Controlador PI linear Discreto .....	45
4.3.1 Simulações da Malha de Controle com o Modelo NARMAX e com o Controlador PI Linear Discreto para Diversos Valores de Referência .....	47
4.4 Projeto de um Controlador <i>Fuzzy</i> com Ação PI.....	50
4.4.1 Simulações da Malha de Controle com o Modelo NARMAX e com o Controlador <i>Fuzzy</i> para Diversos Valores de Referência .....	51
4.5 Projeto de um Controlador <i>Rough</i> com Ação PI.....	53
4.5.1 Simulações da Malha de Controle com o Modelo NARMAX e com o Controlador <i>Rough</i> para Diversos Valores de Referência.....	56
4.6 Análises de Estabilidade.....	58
5 Resultados Experimentais.....	63
5.1 Programas de Controle e Ajustes de Escalas .....	64

<b>5.2 Ensaio Prático com o Controlador PI Linear Discreto</b> .....	<b>65</b>
<b>5.3 Ensaio Prático com o Controlador <i>Fuzzy</i></b> .....	<b>68</b>
<b>5.4 Ensaio Prático com o Controlador <i>Rough</i></b> .....	<b>71</b>
<b>5.5 Considerações Finais</b> .....	<b>74</b>
<b>6 Conclusões</b> .....	<b>78</b>
<b>Apêndice A – Telas de Controle e Supervisão do Programa de Controle Linear com Ação PI</b> .....	<b>81</b>
<b>Apêndice B – Programa de Estimação do Modelo ARX</b> .....	<b>83</b>
<b>Apêndice C - Programa de Estimação do Modelo NARMAX</b> .....	<b>84</b>
<b>Apêndice D – Programa de Simulação da Malha de Controle com Ação Linear</b> .....	<b>86</b>
<b>Apêndice E – Programa do Experimento Prático da Malha de Controle via Controlador PI Linear Discreto</b> .....	<b>88</b>
<b>Apêndice F – Programa de Simulação da Malha de Controle com o PI Linear Discreto</b>	<b>91</b>
<b>Apêndice G – Programa de Simulação da Malha de Controle com o PI <i>Fuzzy</i></b> .....	<b>93</b>
<b>Apêndice H – Programa de Avaliação do Tempo de Processamento das Regras <i>Fuzzy versus Rough</i></b> .....	<b>97</b>
<b>Apêndice I – Programa de Simulação da Malha de Controle com o PI <i>Rough</i></b> .....	<b>102</b>
<b>Apêndice J – Programa do Experimento Prático da Malha de Controle com o PI <i>Fuzzy</i></b> ....	<b>105</b>
<b>Apêndice L – Programa do Experimento Prático da Malha de Controle com o PI <i>Rough</i></b> .....	<b>109</b>
<b>Referências</b> .....	<b>113</b>
<b>Anexo A – Datasheet NI USB-6008</b> .....	<b>116</b>

# 1 Introdução

Esta dissertação tem como objetivo realizar comparações de desempenho de malhas de controle associadas a um processo não-linear, onde serão utilizados três tipos de controladores (um com característica de atuação linear e dois com atuações não-lineares).

Entre as abordagens utilizadas para se controlar sistemas que apresentam comportamentos não-lineares, pode-se citar as técnicas de linearização por realimentação, sistemas de controle de modos deslizantes, e controladores adaptativos (SLOTINE E LI, 1991), entre outras.

Quando não se dispõem de modelos dos processos a serem controlados com exatidões adequadas, aplicações de abordagens que empregam técnicas de inteligência artificial constituem alternativas que apresentam bons resultados em aplicações práticas (GUPTA E SINHA, 1996), com metodologias baseadas em lógica difusa (*fuzzy*) ou redes neurais artificiais, por exemplo.

Neste trabalho serão utilizados conceitos relacionados com conjuntos difusos ou *fuzzy sets* (ZADEH, 1965) e conjuntos aproximados ou *rough sets* (PAWLAK, 1982) para a realização de controladores específicos que são abordados no desenvolvimento desta dissertação. A finalidade consiste na obtenção de controladores com características adequadas para a compensação de processos com comportamentos não-lineares como alternativa aos procedimentos convencionais utilizados em sistemas de controle típicos.

Como objeto de estudo é considerado um processo não-linear usual em algumas aplicações industriais, especificamente um sistema de nível. E na para realização dos experimentos práticos é utilizada uma bancada de laboratório contendo um processo de nível em escala reduzida. Um sistema de aquisição de dados acoplado a um computador constitui o *hardware* utilizado com a tarefa de realizar coletas de dados dos ensaios experimentais, bem como processar (em tempo real) os algoritmos de controle associados. Modelos matemáticos correspondentes também são assumidos em uma etapa anterior a parte experimental, cuja finalidade é de se projetar, analisar e simular as malhas de controle consideradas com os respectivos controladores adotados.

A estrutura do texto está organizada conforme descrito a seguir. O Capítulo 2 faz uma revisão bibliográfica relacionada com os assuntos pesquisados para a elaboração deste documento. O Capítulo 3 revisa alguns conceitos utilizados durante o desenvolvimento da pesquisa. No Capítulo 4 é abordada a proposta do trabalho e a metodologia utilizada. O Capítulo 5 apresenta os resultados obtidos por meio de simulações computacionais e experimentos práticos. Finalmente, espera-se que os resultados obtidos indiquem uma melhoria no desempenho das estruturas dos controladores considerados em relação às estratégias de controle convencionais. Como indicadores de desempenho são utilizados algumas métricas que estão descritas no texto.

No Capítulo 6 encontram-se as considerações finais e sugestões para trabalhos futuros.

## 2 Revisão Bibliográfica

Malhas de controle são usualmente empregadas em processos industriais diversos, sistemas embarcados em veículos, aeronaves e em alguns equipamentos comerciais.

Como alternativa aos procedimentos convencionais utilizados em sistemas de controle típicos, este trabalho tem como objetivo realizar comparações de desempenho de malhas de controle associadas a um processo não-linear, onde são testados três tipos de controladores: um PI (proporcional - integral) linear discreto; um compensador PI *fuzzy*; e um controlador PI *rough*. E espera-se que os resultados obtidos com as técnicas de IA (inteligência artificial) promovam uma melhoria no desempenho das estruturas dos controladores considerados em relação a estratégias de controle convencionais. Considera-se um processo não-linear usualmente aplicado na área industrial, especificamente um sistema de nível.

A representação de processos por meio de modelos matemáticos é fundamental em várias áreas das ciências e engenharias. Eles fornecem informações fundamentais para técnicas de projeto de controladores associados, otimização de controles de processos, detecção de falhas em sistemas de automação, etc. Como consequência, uma forte demanda de técnicas de modelagem e identificação de sistemas dinâmicos surgiu no meio acadêmico e industrial (NELLES, 2001). Um sistema não-linear é caracterizado por não se aplicar o princípio da superposição, ou seja, a resposta produzida pela aplicação simultânea de duas funções de excitação não é a soma das correspondentes respostas individuais (OGATA, 2010). Existem modelagens adequadas para representar sistemas com características não-lineares (AGUIRRE, 2004; NELLES, 2001) como os modelos NARX (*nonlinear autoregressive model with exogenous inputs* ou modelo não-linear autorregressivo com entradas exógenas), NARMAX (*nonlinear autoregressive moving average model with exogenous inputs* ou modelo não-linear autorregressivo de média móvel com entradas exógenas), modelagens baseadas em redes neurais artificiais, modelos *fuzzy*, entre outros.

Nas abordagens tradicionais de malhas de controle aplicadas em sistemas não-lineares, empregam-se procedimentos de linearização dos modelos dos processos a se controlar (OGATA, 2010), cuja finalidade consiste em simplificar os procedimentos de projeto e sintonia dos controladores associados. A aplicação de controladores com ações lineares em

processos com comportamentos não-lineares resulta em desempenhos adequados das malhas de controle correspondentes quando se considera um valor nominal de operação de um determinado processo. Para contextos operacionais distintos de um determinado valor nominal de projeto, as malhas com controladores de ações lineares podem não apresentar desempenhos satisfatórios quando aplicadas a processos com características não-lineares. Sistemas reais apresentam não linearidades como: atrito de Coulomb, saturações, zonas mortas, folgas, e histerese, entre outras. Para controlar processos que possuem comportamentos não-lineares, pode-se citar as técnicas de linearização por realimentação, sistemas de controle de modos deslizantes e controladores adaptativos, entre outras (SLOTINE E LI, 1991).

O projeto de controle de sistemas não-lineares via linearização por realimentação consiste em transformar algebricamente a dinâmica modelada de um determinado sistema não-linear em um (total ou parcialmente) modelo linear usando realimentação de estados, de modo que as técnicas de controle lineares possam ser aplicadas. Isso difere de procedimentos de linearização convencional em que a modelagem resultante é alcançada por transformações exatas de estado do sistema considerado em vez de uma linearização aproximada da dinâmica considerada. Esta abordagem apresenta bons resultados em determinados problemas práticos de sistema de controle como: em helicópteros, aviões de alto desempenho, robôs industriais e outros. O método possui limitações, pois não pode ser utilizado em modelos que não apresentem todos os estados disponíveis (estimados ou mensurados). A resolução analítica das equações diferenciais parciais que definem as transformações de linearização de estado não é trivial. Outro problema se deve ao fato de que um modelo com elevada exatidão nem sempre está disponível para realização da linearização por realimentação (SLOTINE E LI, 1991).

Os controladores por modos deslizantes fornecem uma maneira sistemática para abordar questões de estabilidade e de desempenho consistentes em sistemas que apresentam incertezas em modelagens, distúrbios externos, comportamentos não-lineares ou variações de parâmetros (SLOTINE E LI, 1991). Entre os diversos campos de aplicações práticas em que os controladores por modos deslizantes obtiveram grande aceitação estão relacionados com conversores estáticos de potência, acionamentos elétricos e robótica. O princípio de operação

de técnicas de controles por modos deslizantes consiste em utilizar chaveamentos de ganhos nas leis de controle com o objetivo de modificar a dinâmica dos sistemas controlados de modo que os estados dos mesmos sejam levados e mantidos em uma superfície do espaço de estados especificada pelo projetista. Uma questão desta abordagem diz respeito à atenuação ou eliminação do fenômeno conhecido como *chattering*, responsável por determinados efeitos que podem ocasionar problemas práticos como na excitação de dinâmicas não modeladas, a deterioração do desempenho do controlador, o aumento do desgaste em partes mecânicas móveis e perdas por dissipação de calor em circuitos elétricos de potência. Guaracy (2013) apresentou uma abordagem para auxiliar na escolha dos ganhos de controladores de modos deslizantes, especificando de maneira mais exata a faixa na qual os ganhos devem ser escolhidos para resultar em uma malha de controle com respostas adequadas para processos representados ou aproximados por modelos de primeira ordem. Este trabalho também utilizou conceitos relacionados com conjuntos aproximados no procedimento de ajuste de parâmetros da lei de controle associada. Um problema relacionado com a técnica de controles por modos deslizantes está associado à obtenção de informações associadas com as variáveis de estado do processo controlado, onde geralmente se necessita empregar estimadores ou observadores de estados para se avaliar as variáveis correspondentes.

Um sistema de controle adaptativo é aplicado em processos dinâmicos que apresentem variações de parâmetros ou características não-lineares. A ideia básica consiste em identificar em tempo real o modelo correspondente do processo controlado, e a partir da modelagem resultante modificar os ganhos do controlador associado com a finalidade de estabelecer uma determinada resposta dinâmica especificada. O problema principal com esta abordagem está relacionado com o custo computacional para a identificação em tempo real dos parâmetros do processo a ser controlado.

Abordagens que empregam técnicas de inteligência artificial constituem alternativas que apresentam bons resultados em aplicações práticas quando não se dispõem de modelos dos processos a serem controlados com exatidões adequadas, como as metodologias baseadas em lógica difusa (*fuzzy*) ou redes neurais artificiais, por exemplo, (GUPTA E SINHA, 1996). Outra abordagem está relacionada com os controladores denominados aproximados (*rough*) que também podem ser utilizados para compensar sistemas com dinâmicas não-lineares

(PINHEIRO, CAMATTA E REZEK, 2012). Machado e Pinheiro (2013) propuseram uma nova abordagem para projetar controladores baseados em regras usando conceitos de conjuntos aproximados e o método de realimentação de estado. O procedimento foi aplicado em um processo de nível com comportamento não-linear. No trabalho de Silva, Pinheiro e Mendes (2014) foi verificada a viabilidade da aplicação de um controlador *lead/lag rough* em estabilizadores de sistema de potência, onde um menor tempo de processamento foi obtido na computação das regras de controle em relação às regras correspondentes de um controlador *fuzzy*. Este trabalho utilizou um procedimento de conversão de regras *fuzzy* em aproximadas (*rough*) apresentado no artigo de Guaracy, Pinheiro e Ferreira. (2014).

### 3 Revisão de Conceitos

Este Capítulo apresenta os conceitos básicos e fundamentos necessários para o desenvolvimento desta dissertação, fornecendo os subsídios essenciais para a compreensão da metodologia a ser utilizada.

#### 3.1 Modelagem e Identificação de Sistemas

As técnicas usuais de sintonia de malhas de controle utilizam modelos dos processos a serem controlados nos procedimentos de projeto e nas simulações associadas (OGATA, 2010). A modelagem de sistemas dinâmicos pode ser classificada como fenomenológica ou empírica (SOUZA E PINHEIRO, 2008), onde na primeira categoria são utilizadas relações matemáticas associadas a fenômenos conhecidos que estão relacionados com os processos considerados. Nas abordagens empíricas são utilizados dados provenientes de medidas das informações das entradas e saídas dos processos considerados e, através de processamentos apropriados, são obtidos modelos matemáticos associados, sendo estas abordagens conhecidas como identificação de sistemas. Outra classificação que utiliza conjuntamente as duas abordagens citadas anteriormente é conhecida como modelagem caixa cinza.

Na modelagem caixa branca ou fenomenológica é necessário conhecer as leis envolvidas na descrição matemática do processo considerado. Um exemplo de modelagem fenomenológica é mostrado abaixo, onde se tem ilustrado um processo de nível (Figura 3.1).

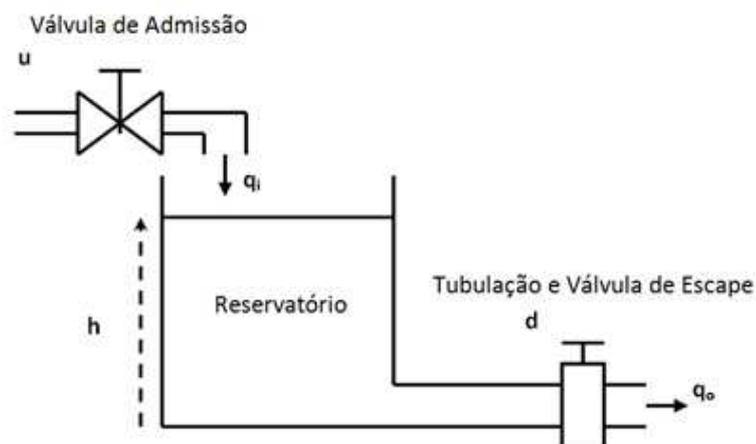


Figura 3.1 – Sistema de nível

O modelo matemático básico deste sistema é representado pela equação diferencial (3.1) (SOUZA E PINHEIRO, 2008), onde “A” denota a área da secção transversal do reservatório, “k” é uma constante de proporcionalidade e “c” é um parâmetro que depende da secção transversal da válvula de escape e da tubulação de saída do processo. A grandeza a ser controlada é o nível ( $h$ ) do sistema e a ação de controle é exercida via tensão ( $u$ ) de comando da válvula de admissão (ou da servo-bomba), que controla o fluxo de entrada ( $q_i$ ) de um determinado fluido no reservatório, onde o nível do processo fica estável em um valor nominal quando o fluxo de entrada se iguala ao de saída ( $q_o$ ). A perturbação externa “ $d$ ” representa outro processo que modifica o fluxo de saída do fluido do sistema, provocando uma variação correspondente no nível do reservatório.

$$A \frac{dh(t)}{dt} = q_i(t) - q_o(t) = k * u(t) - c\sqrt{h(t)} \quad (3.1)$$

Este modelo é não-linear e contínuo no tempo, sendo linearizado em torno de um ponto de operação  $h(t) = \hat{h}$ , onde o último termo da equação passa a ser modelado agora por uma função linear na forma  $c_1 h(t)$ , possibilitando a aplicação de técnicas clássicas de controle.

Entretanto, quando pouco ou nenhum conhecimento prévio de um determinado sistema é disponível, a modelagem empírica ou caixa preta é usualmente aplicada no lugar da abordagem fenomenológica.

Neste contexto os procedimentos básicos utilizados nas etapas de identificação de sistemas dinâmicos são:

**Coleta de dados.** São realizadas via sistemas de aquisição de dados, coletores de dados ou através da própria instrumentação do processo. Deve-se atentar aos tipos de sinais de excitação utilizados, número de amostras e o valor do tempo de amostragem empregado;

**Escolha da representação do modelo.** É realizada conforme as características do sistema a ser modelado (linear, não-linear, com ou sem variações de parâmetros).

**Escolha da estrutura do modelo.** Consiste na escolha do número de coeficientes da modelagem considerada. Quanto menor o número de coeficientes de um modelo, mais rápido será a sua computação.

**Estimação de parâmetros.** Constitui na aplicação de um processamento numérico nos dados mensurados de um sistema prático, visando à determinação dos coeficientes do modelo. O procedimento usual é baseado no método dos mínimos quadrados ou derivados.

**Teste de validação do modelo.** É necessário comparar os dados coletados com a simulação do modelo obtido para verificação da exatidão resultante. É comum usar outro conjunto de dados para validar o modelo, com informações diferentes das utilizadas na etapa de estimação de parâmetros.

### 3.1.1 Modelo ARX

Os modelos autorregressivos com entradas exógenas (ARX) são adequados para modelar sistemas com características lineares e parâmetros invariantes no tempo. A Equação (3.2) define a representação genérica deste e de outros modelos lineares discretos no tempo (AGUIRRE, 2004), sendo  $z^{-1}$  a notação do operador de atraso de transporte correspondente ao tempo de amostragem ( $T$ ) das informações associadas às variáveis do modelo (onde  $k = nT$  com “ $n$ ” denotando a amostra corrente). A variável  $v(k)$  simboliza incertezas ou ruídos associados com as medições das informações na entrada e saída de um sistema, e  $A(z)$ ,  $B(z)$ ,  $C(z)$ ,  $D(z)$  e  $F(z)$  são polinômios expressos por (3.3) que definem os coeficientes do modelo em questão.

$$A(z)y(k) = \frac{B(z)}{F(z)}u(k) + \frac{C(z)}{D(z)}v(k) \quad (3.2)$$

$$\begin{aligned} A(z) &= 1 - a_1z^{-1} - \dots - a_{ny}z^{-ny}; \\ B(z) &= b_1z - \dots - b_{nu}z^{-nu}; \\ C(z) &= 1 + c_1z^{-1} - \dots - c_{nv}z^{-nv}; \\ D(z) &= 1 + d_1z^{-1} - \dots - d_{nd}z^{-nd}; \\ F(z) &= 1 + f_1z^{-1} - \dots - f_{nf}z^{-nf} \end{aligned} \quad (3.3)$$

Considerando  $C(z) = D(z) = F(z) = 1$  em (3.2), tem-se (3.4) que define um modelo ARX genérico. E com  $D(z) = F(z) = 1$  tem-se uma modelagem ARMAX (autorregressiva

de média móvel com entradas exógenas) correspondente.

$$A(z)y(k) = B(z)u(k) + v(k) \quad (3.4)$$

Para a representação acima, assumindo  $A(\cdot) = a_1$  e  $B(\cdot) = b_1$ , a função (3.5) ilustra a modelagem correspondente e em (3.6) segue a equação de diferenças associada, onde estas representações modelam com um determinado erro o processo ilustrado na Figura 3.1, cujo modelo com maior exatidão é expresso por (3.1).

$$\frac{y(z)}{u(z)} = \frac{b_1}{z - a_1} \quad (3.5)$$

$$y(k) = a_1 y(k - 1) + b_1 u(k - 1) \quad (3.6)$$

Na estimação de parâmetros de modelos discretos, usualmente se emprega o método dos mínimos quadrados, onde um modelo típico de regressão é expresso por (3.7), onde  $\psi(k - 1)$  é o vetor das variáveis regressoras e  $\xi(k)$  representa erros ou incertezas de modelagem.

$$y(k) = \psi^T(k - 1)\theta + \xi(k) \quad (3.7)$$

O método dos mínimos quadrados tem como objetivo minimizar a função de custo (3.8) com a finalidade de se obter a menor somatória dos erros quadráticos da modelagem resultante.

$$J = \sum_{k=1}^N \xi(k)^2 = \xi^T \xi \quad (3.8)$$

Substituindo (3.7) em (3.8) e derivando a equação resultante, obtém-se (3.9) que possibilita numericamente o cálculo dos coeficientes do modelo considerado a partir dos dados de

medições do sistema.

$$\theta = [\psi^T \psi]^{-1} \psi^T y \quad (3.9)$$

### 3.1.2 Modelo NARMAX

Um modelo não-linear ARMAX denominado NARMAX (AGUIRRE, 2004) é adequado para representar sistemas com características não-lineares, principalmente quando modelos lineares ARX ou ARMAX não são suficientes para representar matematicamente vários pontos de operação de um processo prático que apresente não linearidades no seu comportamento real. A expressão (3.10) define um modelo polinomial com um atraso de tempo  $T_d$  e nenhum termo adicional em que algum parâmetro deva ser estimado com dependência de  $e(k)$ .

$$y(k) = F^\ell [y(k-1), \dots, y(k-n_y), u(k-T_d), \dots, u(k-T_d-n_u+1), e(k-1), \dots, e(k-n_e)] + e(k) \quad (3.10)$$

O termo  $e(k)$  simboliza todos os efeitos que não podem ser adequadamente representados por  $F^\ell[\cdot]$  uma função polinomial de  $y(k)$ ,  $u(k)$  e  $e(k)$  com grau de não linearidade  $\ell$ . A parte determinística de (3.10) pode ser expandida como o somatório de termos com graus de não linearidade em  $1 \leq m \leq \ell$ . Assim, cada termo de grau " $m$ " poderá conter um fator de grau " $p$ " em  $y(k-i)$  e um fator de grau  $(m-p)$  em  $u(k-i)$ . Os parâmetros representados por  $c_{p,m-p}(n_1, \dots, n_m)$  definem a correspondente equação (3.11).

$$y(k) = \sum_{m=0}^{\ell} \sum_{p=0}^m \sum_{n_1, n_m}^{n_y, n_u} c_{p,m-p}(n_1, \dots, n_m) \prod_{i=1}^p y(k-n_i) \prod_{i=p+1}^m u(k-n_i) \quad (3.11)$$

Neste trabalho é considerado um modelo NARMAX com grau de não linearidade e número de atrasos igual a três, cuja equação de diferenças correspondente é expressa por (3.12) com os regressores associados dados por (3.13), onde a expressão (3.9) é aplicável para

se calcular os valores dos coeficientes do modelo.

$$y(k) = \theta_1 y(k-1) + \theta_2 y(k-2) + \theta_3 u(k-2) + \theta_4 u(k-1) + \theta_5 y^2(k-1)u(k-1) + \theta_6 u^2(k-1)u(k-3) + \theta_7 y^3(k-3) + \theta_8 y(k-2)u(k-2) + \theta_9 y^2(k-3)u(k-3) \quad (3.12)$$

$$\Psi = \begin{bmatrix} y(2) & y(1) & u(1) & u(2) & y^2(2).u(2) & u^2(2).u(0) & y^3(0) & y(1).u(1) & y^3(0).u(0) \\ y(3) & y(2) & u(2) & u(3) & y^2(3).u(3) & u^2(3).u(1) & y^3(1) & y(2).u(2) & y^3(1).u(1) \\ y(4) & y(3) & u(3) & u(4) & y^2(4).u(4) & u^2(4).u(2) & y^3(2) & y(3).u(3) & y^3(2).u(2) \\ \dots & & & & & & & & \\ \dots & & & & & & & & \end{bmatrix};$$

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \dots \\ \theta_9 \end{bmatrix}; Y = \begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \dots \\ \dots \end{bmatrix}. \quad (3.13)$$

### 3.2 Controlador PI Linear Discreto

Controladores digitais são dispositivos utilizados frequentemente na realização de malhas de controle de processos industriais (DORF E BISHOP, 2001), e entre as leis de controle mais empregadas encontram-se os controladores com ações proporcionais e integrais ou PI (ASTROM E WITTENMARK, 1990). O diagrama de blocos na Figura 3.2 representa um controlador PI linear discreto, onde a variável “ $e$ ” representa a informação do erro da malha de controle associada, “ $u$ ” simboliza a ação de comando correspondente. Os ganhos associados com as ações (proporcional e integral) são expressos pelos parâmetros  $K_p$  e  $K_i$ , respectivamente.

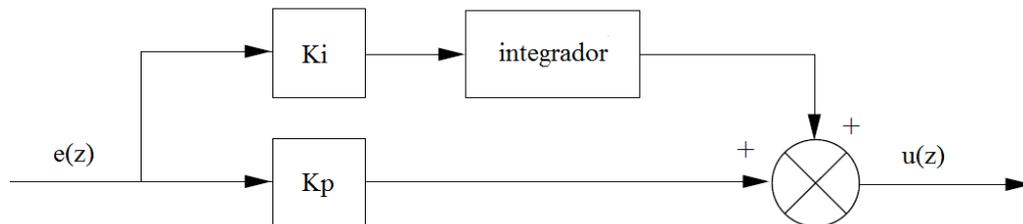


Figura 3.2 – Controle PI linear discreto

A função de modelagem básica de um controlador PI linear discreto é representada por (3.14), onde o integrador é definido pela aproximação de Euler (PHILLIPS E NAGLE, 1995), sendo “ $T$ ” o tempo de amostragem/varredura utilizado.

$$C(z) = \frac{u(z)}{e(z)} = K_p + K_i \frac{T}{z-1} = \frac{K_p z + K_i T - K_p}{z-1} \quad (3.14)$$

### 3.3 Compensação via Lugar das Raízes

A compensação de um sistema de controle via (conceituação) Lugar das Raízes tem como procedimento típico a adição de polos e zeros na função de compensação, cujo objetivo é posicionar as raízes da equação característica da função de malha fechada do sistema de controle em valores apropriados no plano  $Z$  (PHILLIPS E NAGLE, 1995). Uma das formas de compensação consiste em especificar os polos dominantes que atendam à dinâmica solicitada, igualando os valores associados com a equação característica da função de malha fechada, como exemplificado na equação (3.15) e pela Figura 3.3 que ilustra uma malha de controle básica, sendo  $R$  o valor real do polo especificado e  $I$  o valor imaginário associado.

$$(z - R - jI)(z - R + jI) = 1 + C(z)G(z) = 0 \quad (3.15)$$

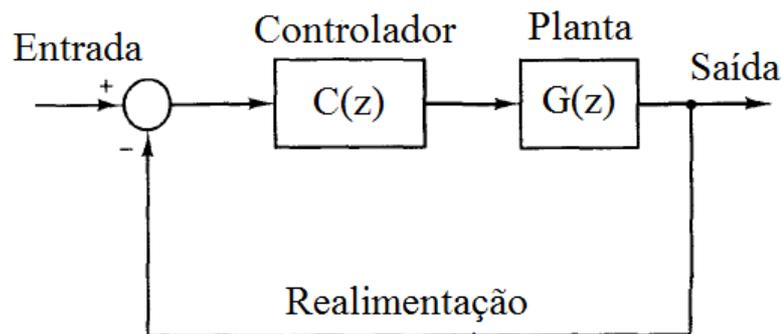


Figura 3.3 – Diagrama de blocos de uma malha de controle básica

Dada a especificação (3.16) de um determinado polo dominante de uma malha de controle, é possível via (3.17) e (3.18) estimar a resposta dinâmica (máximo pico e tempo de acomodação) expressa por (3.19) e (3.20), para uma entrada de referência em degrau aplicada

na malha de controle em questão (PHILLIPS E NAGLE, 1995).

$$z_{1,2} = R \pm jI = M \angle \pm \varphi \quad (3.16)$$

$$\zeta = \frac{-\ln(M)}{\sqrt{(\ln M)^2 + \varphi^2}} \quad (3.17)$$

$$W_n = \frac{1}{T} \sqrt{(\ln M)^2 + \varphi^2} \quad (3.18)$$

$$M_p = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} \quad (3.19)$$

$$T_a = \frac{4}{\zeta W_n} \quad (3.20)$$

### 3.4 Controlador *Fuzzy* com Ação PI

Os controladores *fuzzy* são modelos baseados em regras que utilizam os conjuntos difusos (*Fuzzy Sets*) propostos por Zadeh (1965) como alternativa para a representação de informações imprecisas ou vagas. A primeira aplicação dos conjuntos difusos em sistemas de controle foi proposta por Mamdani (1974) para um processo de aquecimento.

Os controladores *fuzzy* não são modelados facilmente através de equações diferenciais ou de diferenças, dificultando a aplicação da teoria clássica de controle na sintonia de malhas de compensação que utilizem estes controladores. Em Pinheiro (2000) foi proposta uma abordagem baseada em técnicas de resposta em frequência (balanço harmônico) para análises de estabilidade e sintonia de malhas de controle que utilizam controladores *fuzzy*, onde metodologias clássicas de projeto podem ser adaptadas e aplicadas.

Uma estrutura muito utilizada de controlador *fuzzy* com ação PI está ilustrada na Figura 3.4 (BRAAE E RUTHERFORD, 1979), onde é utilizado o valor do erro ( $e$ ) da malha de controle e a correspondente somatória (ou integral) do erro, que multiplicados pelos fatores de escala  $g_e$  e  $g_v$  resultam nas variáveis ( $E = g_e * e$ ;  $V = g_v * \Sigma e * T$ ) de entrada das regras *fuzzy* expressas por (3.21). A informação “ $U$ ” corresponde ao dado inferido das regras, sendo  $g_d$  um

fator de escala (geralmente de valor unitário) associado à variável de saída ( $u = g_d * U$ ) do controlador, ou seja, a ação de comando da malha de controle em questão. Os fatores de escala correspondem aos ganhos proporcional e integral de um controlador PI discreto, ou seja,  $g_e = K_p$  e  $g_v = K_i$ .

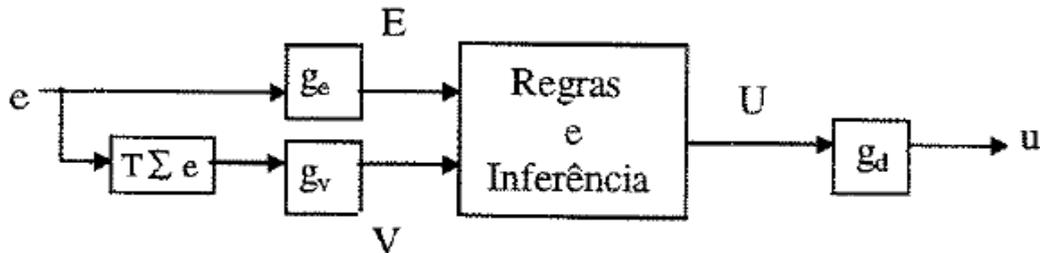


Figura 3.4 – Controlador *fuzzy* com ação PI

Na representação das regras ( $r_i$ ), os conjuntos difusos ( $\alpha_j$  e  $\beta_j$ ) associados estão relacionados com as informações das entradas ( $E$  e  $V$ ) das regras *fuzzy* do controlador. Os conjuntos representados por ( $\sigma_j$ ) também são difusos no caso de se usar regras denominadas linguísticas ou de Mamdani.

$$r_i: \text{IF } E = \alpha_j \text{ AND } V = \beta_j \text{ THEN } U = \sigma_j \quad (3.21)$$

A obtenção de regras *fuzzy*, dependendo da aplicação, é realizada de várias formas: via informações fornecidas por especialistas; por análise qualitativa de dados; com técnicas de regressão; técnicas de agrupamento, métodos de treinamento; e associações com outras técnicas de inteligência artificial (LIMA, PINHEIRO E SANTOS, 2014).

Em controladores *fuzzy* é usual empregar regras similares às representadas em (3.22), onde os conjuntos nebulosos relativos à variável “ $E$ ” são denominados como  $N_1$ ,  $Z_1$ ,  $P_1$ , e  $N_2$ ,  $Z_2$ ,  $P_2$  para a variável “ $V$ ”. As denominações para os conjuntos associados com a variável “ $U$ ” são  $NL$ ,  $NM$ ,  $ZE$ ,  $PM$  e  $PL$ . As etiquetas utilizadas são factíveis de interpretações linguísticas, por exemplo: “ $N$  = Negativo”; “ $P$  = Positivo”; “ $Z$  ou  $ZE$  = Zero”; “ $M$  = Médio”; “ $G$  = Grande”.

$$\begin{aligned}
r_1: & \text{IF } e = N_1 \text{ AND } v = N_2 \text{ THEN } V_0 = NL; \\
r_2: & \text{IF } e = Z_1 \text{ AND } v = N_2 \text{ THEN } V_0 = NM; \\
r_3: & \text{IF } e = P_1 \text{ AND } v = N_2 \text{ THEN } V_0 = ZE; \\
r_4: & \text{IF } e = N_1 \text{ AND } v = Z_2 \text{ THEN } V_0 = NM; \\
r_5: & \text{IF } e = Z_1 \text{ AND } v = Z_2 \text{ THEN } V_0 = ZE; \\
r_6: & \text{IF } e = P_1 \text{ AND } v = Z_2 \text{ THEN } V_0 = PM; \\
r_7: & \text{IF } e = N_1 \text{ AND } v = P_2 \text{ THEN } V_0 = ZE; \\
r_8: & \text{IF } e = Z_1 \text{ AND } v = P_2 \text{ THEN } V_0 = PM; \\
r_9: & \text{IF } e = P_1 \text{ AND } v = P_2 \text{ THEN } V_0 = PL
\end{aligned} \tag{3.22}$$

Em Pinheiro e Gomide (1999<sub>a</sub>) foi mostrado que para funções de pertinência não igualmente espaçadas (Figura 3.5) ou não uniformes que modelam os conjuntos difusos empregados, um controlador *fuzzy* associado apresenta um comportamento adaptativo que é adequado para compensar processos com características não-lineares ou com parâmetros variantes no tempo. As ações de comando resultantes de um controlador *fuzzy* com funções de pertinência com as formatações citadas equivalem às ações de um controlador com ganhos variáveis em relação à magnitude do valor do erro da malha de controle associada, resultando em compensações com atuações adaptativas.

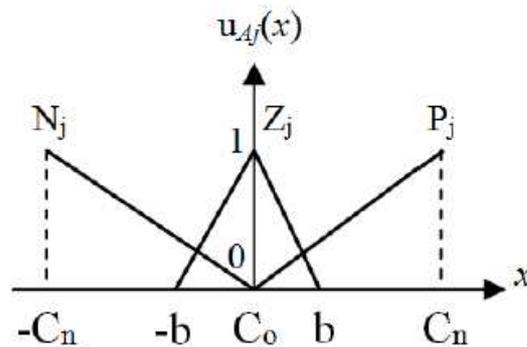


Figura 3.5 – Exemplo de funções de pertinência não igualmente espaçadas

A conversão de uma variável de entrada em um grau de pertinência correspondente ( $u_{Aj}(x) \in [0, 1]$ ) é conhecida como fuzificação de dados, e para funções de pertinência triangulares esta conversão é dada por (3.23).

$$\mu_{zj}(x) = 1 - \frac{|x - C_0|}{b} \quad (3.23)$$

Para cada regra  $r_i$  é realizada uma composição ( $\mu_{r_i}$ ) com os valores de pertinência  $\mu_{A_j}(E)$  e  $\mu_{B_j}(V)$  correspondentes às variáveis de entrada do controlador. Os operadores mais usuais de composição são o mínimo (*min*) ou o produto (\*), cujas expressões respectivas estão indicadas por (3.24) e (3.25).

$$\mu_{r_i} = \min[\mu_{A_j}(E), \mu_{B_j}(V)] \quad (3.24)$$

$$\mu_{r_i} = \mu_{A_j}(E) * \mu_{B_j}(V) \quad (3.25)$$

Um exemplo de funções de pertinência da variável de saída ( $U = V_0$ ) está ilustrado na Figura 3.6.

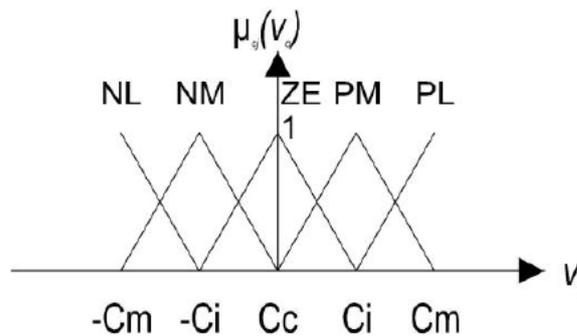


Figura 3.6 – Exemplo de funções de pertinência da variável de saída

A agregação dos valores das composições ( $\mu_{r_i}$ ) das regras resulta em um procedimento conhecido como defuzificação de dados, que agrega esses valores e fornece um dado correspondente à informação de saída do controlador *fuzzy* associado. Os métodos de centro de área ou da altura (DRIANKOV, HELLENDORRN E REINFRANK, 1996) são frequentemente utilizados, cujo procedimento é expresso por (3.26).

$$v_0 = \frac{\sum_{n=1}^N V_n \mu_{rj}(V_n)}{\sum_{n=1}^N \mu_{rj}(V_n)} \quad (3.26)$$

### 3.5 Controlador *Rough* com Ação PI

A conceituação formal de controladores aproximados (*rough*) foi apresentada em Pinheiro, Camatta e Rezek (2012), sendo previamente aplicada por Camatta (2009) na regulação de velocidade e corrente de um motor de corrente contínua, onde foram utilizados controladores *rough* com ações proporcionais e integrais. A denominação de controlador aproximado se deve à aplicação de conceitos de conjuntos aproximados (*rough sets*) com a representação resultante baseada em regras.

A Teoria dos Conjuntos Aproximados (*Rough Sets*) foi proposta por Pawlak (1982) como uma forma de tratar dados com ambiguidades, incertezas e imprecisões em aplicações de bancos de dados e sistemas especialistas. Esta teoria é aplicável em áreas diversas como: mineração de dados; em processos automáticos de decisão; sistemas de diagnóstico; em aprendizado de máquina; e outras.

Em Silva, Pinheiro e Mendes (2014) foi utilizado um procedimento para a conversão de um modelo de controlador *fuzzy* com ação de avanço/atraso de fase (PINHEIRO E GOMIDE, 1999<sub>b</sub>) em um controlador *rough* correspondente. Como aplicação foi considerada a compensação de um sistema elétrico de potência, onde se verificou que o tempo de processamento das regras aproximadas foi menor que o tempo de computação das regras *fuzzy*, pois não são necessárias as etapas de fuzificação e defuzificação de dados, um contexto interessante para aplicações em tempo real.

Este mesmo procedimento de conversão de regras é utilizado nesta dissertação com o objetivo de converter o modelo de um controlador *fuzzy* com ação PI em um controlador *rough* associado. Como aplicação é considerada uma malha de controle de um processo de nível com características não-lineares, onde se espera obter um melhor desempenho no tempo de processamento com o controlador aproximado em relação ao compensador *fuzzy* correspondente.

A próxima seção deste Capítulo aborda o procedimento proposto por Guaracy, Pinheiro, Ferreira (2014) para a conversão de modelos difusos em aproximados.

### 3.5.1 Conversão de Modelos *Fuzzy* em Modelos *Rough*

Seja um modelo *fuzzy* representado por “ $n$ ” variáveis de entrada. Para cada variável  $x_n (n = 1, \dots, N)$  com “ $s$ ” conjuntos difusos associados, identifica-se as regiões correspondentes aos intervalos em que a função de pertinência relacionada ao conjunto “ $m$ ” assume valores maiores do que zero. Essas regiões podem ser representadas por conjuntos poliedrais descritos por (3.27), em que  $H_n^{(m)} \in \mathbb{R}^{2 \times 1}$  e  $S_n^{(m)} \in \mathbb{R}^2$ . A Figura 3.7 ilustra três regiões para uma variável ( $x_1$ ) com três conjuntos *fuzzy* correspondentes.

$$P_n^{(m)} = \{x_n \in \mathbb{R} \mid H_n^{(m)} x_n \leq S_n^{(m)}\} \quad (3.27)$$

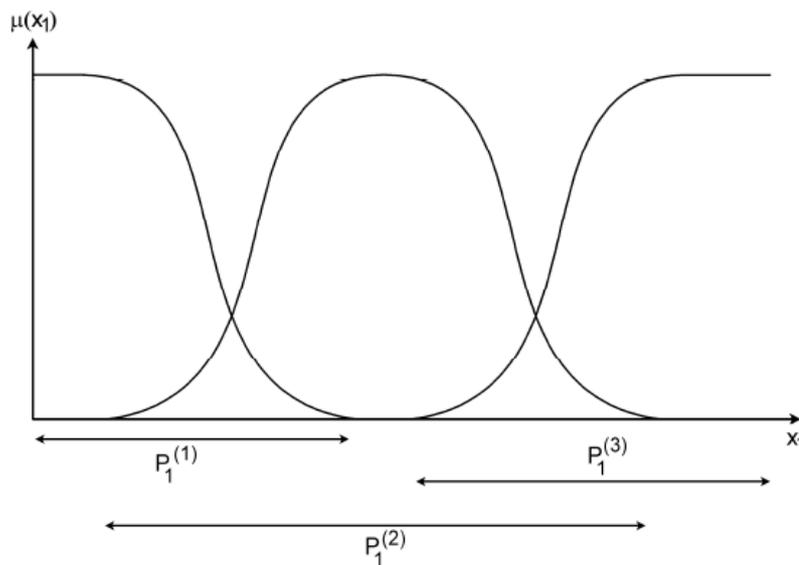


Figura 3.7 – Ilustração das regiões  $P_1^{(m)}$   
 Fonte: Guaracy, Pinheiro e Ferreira (2014)

Novas regiões  $D_n^{(i)}$  ( $i = 1, \dots, p$ ) são definidas nas interseções dos conjuntos  $P_n^{(m)}$ , conforme indicado por (3.28). São desconsideradas regiões redundantes, como as que ocorrem no caso em que mais de dois conjuntos *fuzzy* se sobrepõem no mesmo intervalo.

$$D_n^{(i)} = P_n^{(m)} \cap P_n^{(m+1)} \quad (3.28)$$

Os conjuntos  $D_n^{(i)}$  ( $i = p + 1, \dots, q$ ) restantes podem ser determinados pela equação (3.29).

$$\bigcup_{i=p+1}^q D_n^{(i)} = \left\{ x_n \in \mathbb{R} \mid x_n \in \bigcup_{m=1}^k P_n^{(m)}, x_n \notin \bigcup_{i=1}^p D_n^{(i)} \right\} \quad (3.29)$$

Os conjuntos *crisp* associados a uma determinada variável  $x_i$  são representados por  $D_n$  em (3.30).

$$D_n = \{D_n^{(1)}, D_n^{(2)}, \dots, D_n^{(q)}\} \quad (3.30)$$

Os conjuntos  $D_n^{(i)}$  são redefinidos de maneira que não haja sobreposição nos limites comuns entre duas regiões (Figura 3.8). Numa notação em forma de intervalos, têm-se as expressões (3.31) e (3.32).

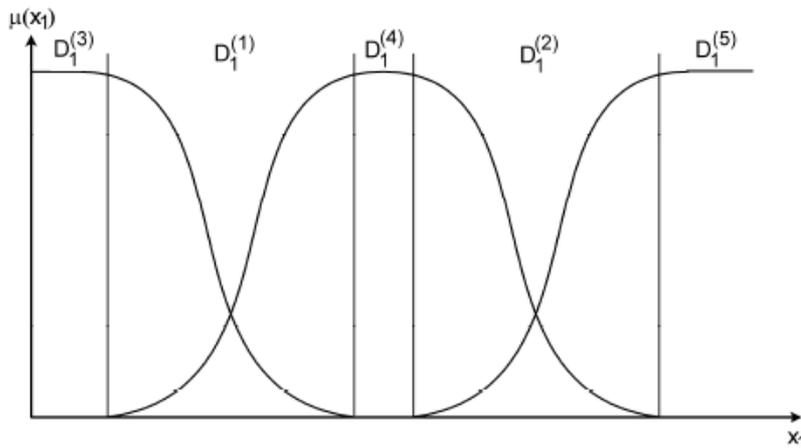


Figura 3.8 – Exemplo do mapeamento de conjuntos *fuzzy* em conjuntos *crisp*  
Fonte: Guaracy, Pinheiro e Ferreira (2014)

$$D_n^{(i)} = \left\{ x_n \in \mathbb{R} \mid \underline{x}_n^{(i)} \leq x_n < \bar{x}_n^{(i)} \right\} = \left[ \underline{x}_n^{(i)}, \bar{x}_n^{(i)} \right) \quad i = 1, \dots, q - 1 \quad (3.31)$$

$$D_n^{(q)} = \{x_n \in \mathbb{R} \mid \underline{x}_n^{(q)} \leq x_n \leq \bar{x}_n^{(q)}\} = [\underline{x}_n^{(q)}, \bar{x}_n^{(q)}] \quad (3.32)$$

Concluída a decomposição dos conjuntos *fuzzy* em conjuntos *crisp* associados, é definido um sistema de informação  $I = (U, A)$  em que o conjunto de atributos correspondente é dado por  $A = \{x_1, x_2, \dots, x_N\}$ , e os objetos do universo  $U$  são definidos a partir dos produtos cartesianos entre os conjuntos  $D_N$  conforme equação (3.33).

$$U = \left\{ \left( (D_1 \times D_2) \times D_3 \right) \times \dots \times D_N \right\} \quad (3.33)$$

A Tabela 3.1 ilustra um exemplo referente a um sistema de informação obtido a partir de um modelo *fuzzy* com duas variáveis de entrada ( $x_1, x_2$ ), cada uma delas dividida em dois conjuntos *crisp* correspondentes que definem quatro instâncias ( $o_i$ ) básicas.

Tabela 3.1 – Exemplo de sistema de informação obtido a partir de um modelo *fuzzy*

$o_i$	$x_1$	$x_2$
$o_1$	$[\underline{x}_1^{(1)}, \bar{x}_1^{(1)}]$	$[\underline{x}_2^{(1)}, \bar{x}_2^{(1)}]$
$o_2$	$[\underline{x}_1^{(1)}, \bar{x}_1^{(1)}]$	$[\underline{x}_2^{(1)}, \bar{x}_2^{(1)}]$
$o_3$	$[\underline{x}_1^{(2)}, \bar{x}_1^{(2)}]$	$[\underline{x}_2^{(1)}, \bar{x}_2^{(1)}]$
$o_4$	$[\underline{x}_1^{(2)}, \bar{x}_1^{(2)}]$	$[\underline{x}_2^{(2)}, \bar{x}_2^{(2)}]$

Fonte: Guaracy, Pinheiro e Ferreira (2014)

Na estimação do valor da saída  $y(x)$  do modelo em questão, considerando dados nos intervalos correspondentes a cada instância  $o_j$ , define-se como  $F(x)$  a saída do modelo *fuzzy* para  $x = (x_1, x_2, \dots, x_N)$ . Definindo  $\underline{x}^{(j)}$  como o vetor composto pelos valores de  $x_i$  correspondentes aos limites inferiores dos intervalos associados à  $o_j$ , ou seja,  $\underline{x}^{(j)} = (\underline{x}_1^{(j)}, \underline{x}_2^{(j)}, \dots, \underline{x}_N^{(j)})$  e  $\bar{x}^{(j)}$  como o vetor composto pelos valores de  $x_i$  correspondentes aos

limites superiores dos intervalos  $\bar{x}^{(j)} = (\bar{x}_1^{(j)}, \bar{x}_2^{(j)}, \dots, \bar{x}_N^{(j)})$ , tem-se a informação de saída obtida pela equação de interpolação (3.34) (PINHEIRO *et al*, 2010). A equação citada pode ser reescrita na notação (3.35), na qual o índice “j” indica o intervalo associado à variável  $x_n$ . As regras de modelagem correspondentes são obtidas agregando-se os atributos de cada instância da Tabela 3.1 com sua respectiva expressão de saída.

$$y = y_i^{(k)} + \frac{y_i^{(m)} - y_i^{(k)}}{N} \left( \sum_{n=1}^N \frac{x_n - x_n^{(k)}}{x_n^{(m)} - x_n^{(k)}} \right) \quad (3.34)$$

$$y^{(j)}(x) = F(\underline{x}^{(j)}) + \left( \frac{F(\bar{x}^{(j)}) - F(\underline{x}^{(j)})}{N} \right) \sum_{n=1}^N \frac{(x_n - \underline{x}^{(j)})}{(\bar{x}^{(j)} - \underline{x}^{(j)})} \quad (3.35)$$

Para um modelo *rough* (aproximado) a partir das regras (3.36), utilizam-se as equações (3.37) e (3.38) (MACHADO E PINHEIRO, 2013) obtidas de (3.34) e (3.35) respectivamente. Finalmente, obtêm-se os coeficientes numéricos do modelo baseado em regras.

$$r_i: IF x_1 = \alpha_i AND x_2 = \beta_i AND \dots AND x_N = \varphi_i THEN y_i = \sigma_j \quad (3.36)$$

$$c_{in} = \frac{1}{N} \left( \frac{y_i^{(m)} - y_i^{(k)}}{x_n^{(m)} - x_n^{(k)}} \right) \quad (3.37)$$

$$c_{i0} = y_i^{(k)} + \frac{y_i^{(m)} - y_i^{(k)}}{N} \left( \sum_{n=1}^N \frac{-x_n^{(k)}}{x_n^{(m)} - x_n^{(k)}} \right) \quad (3.38)$$

Assim para o exemplo considerado com duas variáveis de entrada  $x_1$  e  $x_2$ , as regras aproximadas resultantes assumem o formato indicado por (3.39), onde os conjuntos  $\alpha_i$  e  $\beta_i$  são agora *crisp* não necessitando de uma etapa de fuzificação de dados. Como a informação

de saída de cada regra é dada por uma função polinomial associada aos valores das variáveis de entrada atribuídas a conjuntos *crisp*, também não se necessita de uma etapa de defuzificação de dados. A abordagem de conversão de modelos *fuzzy* em modelos aproximados também se aplica em modelos difusos que utilizam regras funcionais (Takagi-Sugeno), além de regras linguísticas (Mamdani).

$$r_i: IF x_1 = \alpha_i AND x_2 = \beta_i AND \dots AND x_N = \varphi_i \\ THEN y_i = c_{i,0} + c_{i,1}x_1 + c_{i,2}x_2 + \dots + c_{i,N}x_N \quad (3.39)$$

## 4 Metodologia e Desenvolvimento

Neste Capítulo é aplicada a metodologia utilizada nesta dissertação, cuja fundamentação teórica foi apresentada no Capítulo anterior. Uma descrição detalhada da pesquisa desenvolvida neste trabalho é apresentada nas próximas seções deste Capítulo, onde um dos objetivos consiste na conversão do modelo baseado em regras de um controlador *fuzzy* com ação proporcional e integral em uma modelagem correspondente baseada em regras aproximadas (*rough*). Como exemplo de aplicação é considerado um processo de nível em escala reduzida, cujos dados provenientes de ensaios experimentais são utilizados para a obtenção de modelos associados ao processo. É esperado que os resultados obtidos com a proposta deste trabalho promovam uma melhoria no desempenho da malha de controle resultante.

### 4.1 Descrição do Processo de Nível

Em referência ao processo de nível básico ilustrado na Figura 3.1, a Figura 4.1 ilustra o processo utilizado neste trabalho. O sistema é composto por dois reservatórios sobrepostos e interligados, sendo que uma bomba motorizada transfere um determinado líquido (água) do tanque inferior para o superior, onde o nível do líquido do reservatório superior deve ser controlado. Faz parte do sistema um computador (PC), no qual os algoritmos de controle são executados, uma interface com conversores analógico-digital (A/D) e digital-analógico (D/A), um transdutor de nível e um módulo eletrônico para processamento dos sinais do transdutor e do *driver* da bomba motorizada. O nível do processo é medido por duas hastes metálicas em contato com o líquido, correspondendo a um transdutor capacitivo. A alteração do nível do líquido equivale a uma variação do dielétrico do capacitor associado às hastes, que é convertido em uma informação de tensão correspondente por meio de circuitos do módulo eletrônico, sendo esta informação convertida pelo conversor A/D e lida pelo PC. O dado de comando fornecido pelo algoritmo de controle processado pelo computador aciona o *driver* da bomba motorizada via interface D/A do sistema. A foto da Figura 4.2 mostra detalhes da bancada de testes adotada neste trabalho, cujas partes constituintes estão descritas sobre a foto da figura em questão.

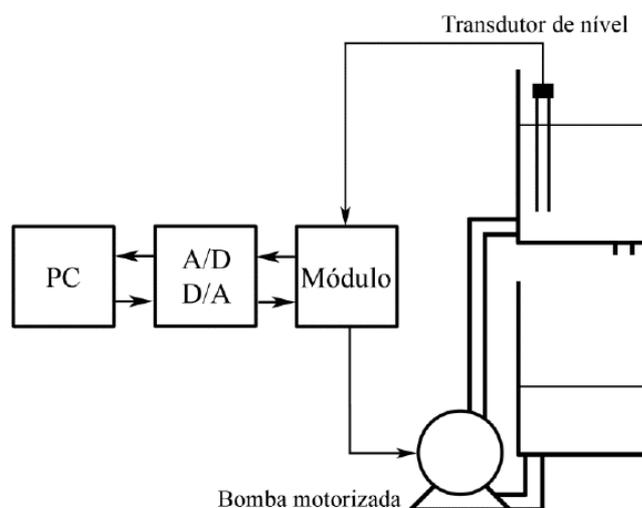


Figura 4.1 – Estrutura do processo utilizado  
Fonte: Guaracy (2013)



Figura 4.2 – Bancada utilizada nos experimentos práticos

As interfaces A/D e D/A utilizadas fazem parte de um sistema de aquisição de dados, o modelo NI USB-6008 da National Instruments, que emprega uma interface USB para comunicar dados com o computador da bancada. No Anexo A encontra-se as informações técnicas básicas do NI USB-6008, e no Apêndice A tem-se ilustrado a tela de supervisão do

programa elaborado no *software* LabVIEW para monitorar alguns ensaios realizados no processo de nível, juntamente com o diagrama de blocos funcional do programa de controle associado.

## 4.2 Modelos Matemáticos

Os modelos matemáticos considerados para o processo de nível descrito foram obtidos através de processamentos adequados nos dados coletados do sistema. As informações mostradas na Figura 4.3 são utilizadas para a obtenção de um modelo ARX, e as informações da Figura 4.4 para um modelo NARMAX. A finalidade destes modelos está associada às etapas de projeto, sintonia e simulação das malhas de controle e respectivos controladores adotados.

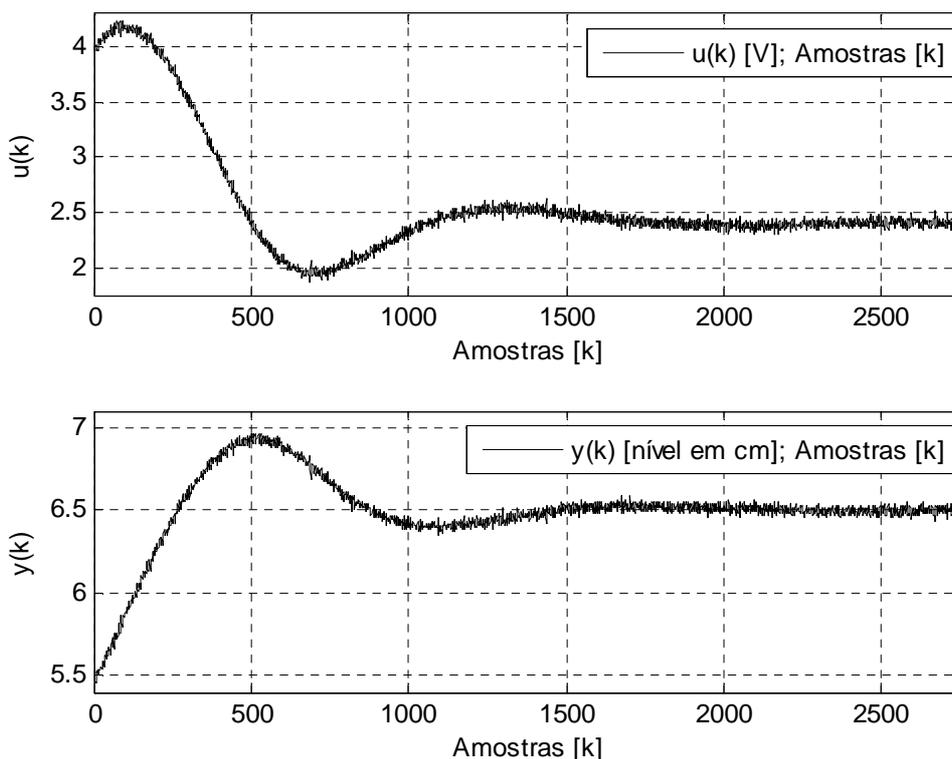


Figura 4.3 – Dados de ensaio do processo de nível em torno de um ponto de operação

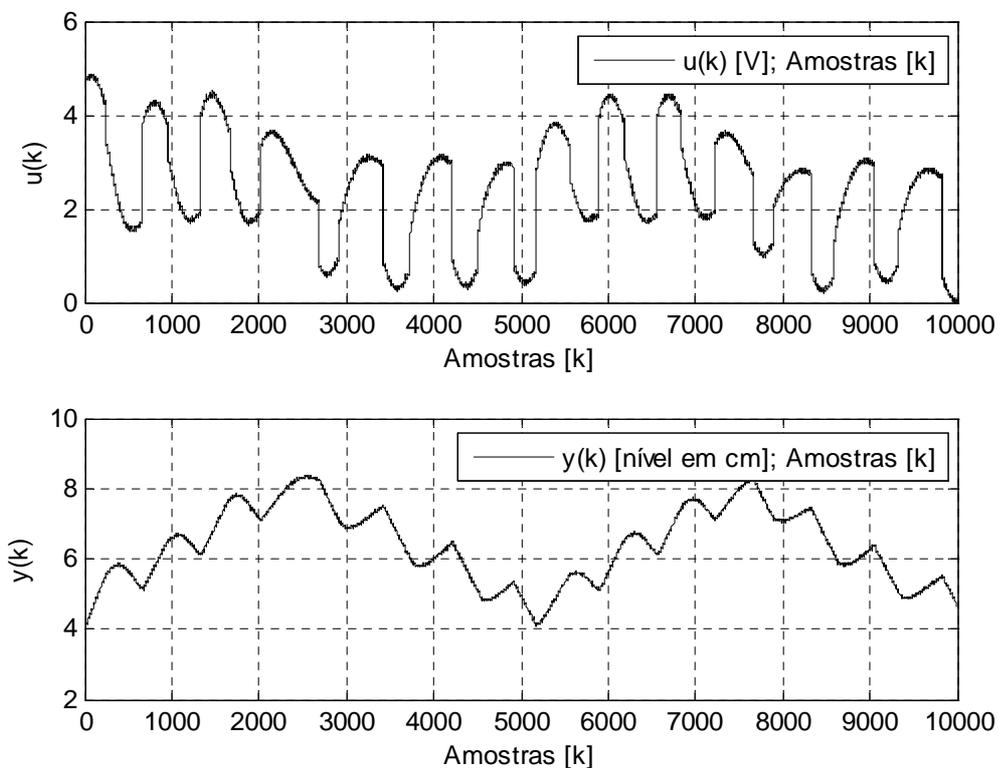


Figura 4.4 – Dados de ensaio do processo de nível para uma faixa de valores de operação

### 4.2.1 Obtenção do Modelo ARX

As técnicas usuais de compensação de malhas de controle utilizam modelos dos processos a serem controlados nos procedimentos de projeto como, por exemplo, a metodologia baseada na conceituação do Lugar das Raízes.

Para identificação de um modelo linear do processo de nível com objetivo de projetar um controlador linear discreto com ação PI, foram utilizados os dados do ensaio associado à Figura 4.3 com valores de operação entre 5,5 a 6,5 unidades de nível, cujo valor médio constitui o dado central da escala de operação do processo em escala reduzida utilizado (onde a unidade considerada está em centímetros). Foram coletadas 750 amostras de entrada ( $u$ ) e de saída ( $y$ ) do processo, onde o intervalo de tempo entre cada amostra foi de 0,065[s]. Os dados coletados foram processados (via programa disponível no Apêndice B), resultando no modelo discreto na forma de (3.5) com os parâmetros numéricos indicados por (4.1). O modelo ARX obtido apresentou um valor de correlação em torno de 87% em relação aos dados do ensaio e

os simulados pelo modelo resultante (Figura 4.5), utilizando-se as mesmas informações de entrada e condições iniciais de saída do processo considerado.

$$\frac{y(z)}{u(z)} = \frac{0,0027}{z - 0,999} \quad (4.1)$$

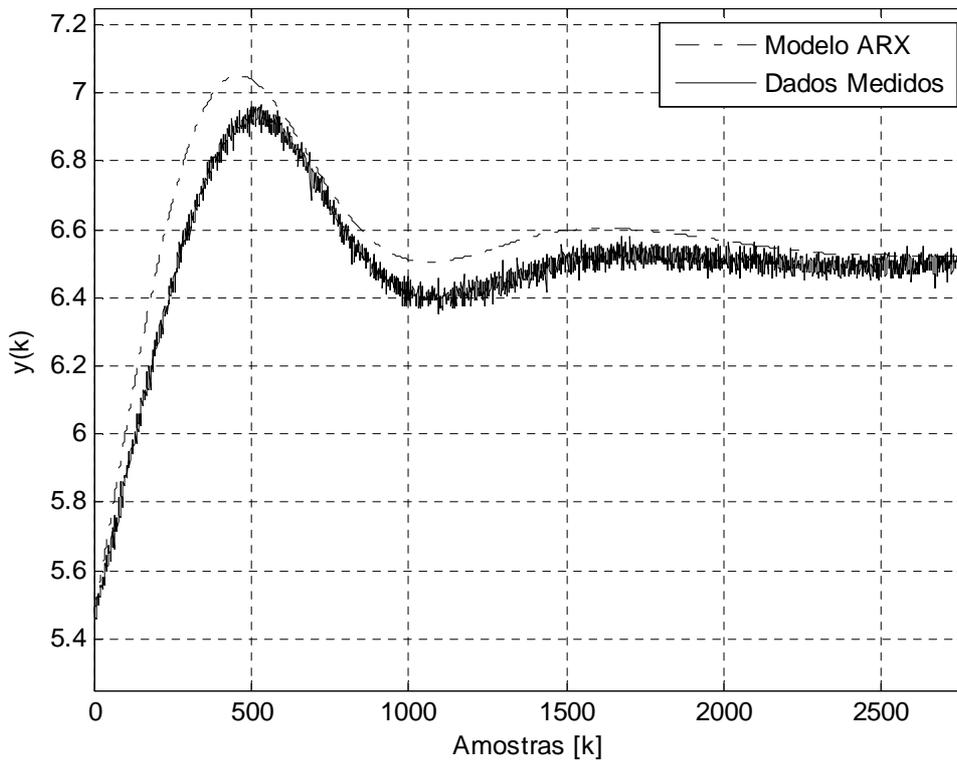


Figura 4.5 – Simulação do modelo ARX e os dados do ensaio realizado

### 4.2.2 Obtenção do Modelo NARMAX

Nesta seção é considerado um modelo com maior exatidão em relação ao modelo ARX obtido no item anterior. Para esta finalidade é considerado um modelo NARMAX, uma vez que não se dispõe de uma modelagem fenomenológica na forma de (3.1). Esta modelagem mais exata é importante para se testar os desempenhos dos controladores que serão adotados nos próximos itens deste Capítulo, utilizando um modelo que incorpore características não-lineares do processo real utilizado. Para identificação de um modelo não-linear do processo de nível com objetivo de projetar um controlador *fuzzy* com ação PI,

utilizou-se os dados do ensaio associado com a Figura 4.4 com valores de operação entre 4 a 8 unidades de nível. Foram coletadas 4999 amostras de entrada ( $u$ ) e de saída ( $y$ ) do processo, onde o intervalo de tempo entre cada amostra foi de 0,065[s]. Os dados coletados foram processados (via programa disponível no Apêndice C), resultando no modelo não-linear na forma de (3.12) com os parâmetros numéricos indicados na equação (4.2). O modelo NARMAX obtido apresentou um valor de correlação em torno de 97% em relação aos dados de ensaio e simulados (Figura 4.6), utilizando-se as mesmas informações de entrada e condições iniciais de saída do processo considerado.

$$\begin{aligned}
 y(k) = & 0,51469y(k-1) + 0,48182y(k-2) + 0,00042547u(k-2) \\
 & -0,0030307u(k-1) + 2,8506 \times 10^{-5}y^2(k-1)u(k-1) \\
 & -2,3374 \times 10^{-5}u^2(k-1)u(k-3) + 4,0015 \times 10^{-5}y^3(k-3) \\
 & +0,0030301y(k-2)u(k-2) - 0,00031422y^2(k-3)u(k-3) \quad (4.2)
 \end{aligned}$$

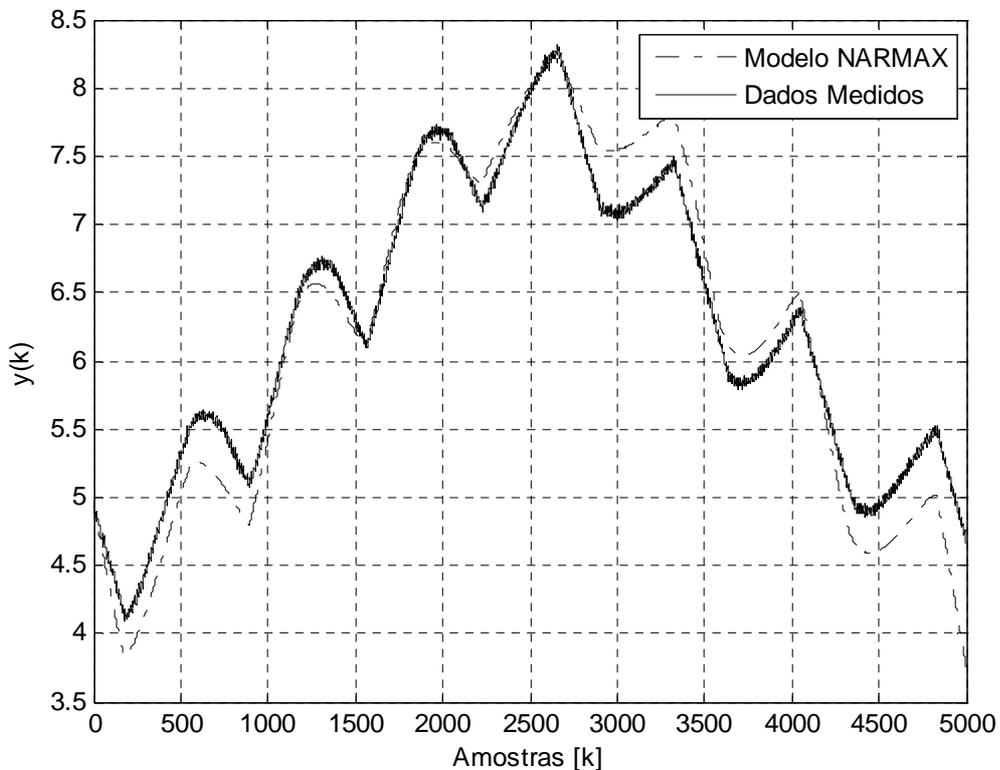


Figura 4.6 – Simulação do modelo NARMAX e os dados do ensaio realizado

### 4.3 Projeto de um Controlador PI linear Discreto

Nesta seção é considerado o cálculo dos ganhos de um controlador PI discreto típico, cujos valores servirão de base para a construção dos controladores *fuzzy* e aproximado. A metodologia de cálculo é baseada na conceituação do Lugar das Raízes. Os dados especificados para os polos dominantes da malha de controle resultante (considerando o modelo ARX (4.1) obtido) foram  $0,9981 \pm j 0,0038$ , cujos valores correspondem a uma dinâmica com um tempo de acomodação ( $T_a$ ) igual ou menor que 137 *segundos* e um máximo pico ( $M_p$ ) igual ou menor que 20% (informações estimadas por meio das equações (3.17)-(3.20), considerando um tempo de amostragem/varredura  $T = 0,065[s]$ ). O procedimento de compensação foi apresentado no Capítulo 3 via equação (3.15), ou por (4.3) de onde se obtêm os valores  $K_p \approx 1$ ,  $K_i \approx 0,1$  e a função de compensação dada pela equação (4.4) com ação proporcional e integral ( método de Euler).

$$(z - 0,9981 - j0,0038)(z - 0,9981 + j0,0038) = 1 + \left( \frac{K_p z + K_i T - K_p}{z - 1} \cdot \frac{0,0027}{z - 0,999} \right) \quad (4.3)$$

$$C(z) = 1 + 0,1 \frac{T}{z - 1} \quad (4.4)$$

Simulando a malha de controle linear resultante (similar a da Figura 3.3) via programa disponível no Apêndice D (incorporada a relação de realimentação do transdutor do processo), tem-se na Figura 4.7 a resposta ao degrau do sistema em questão, cuja dinâmica resultante apresentou os seguintes valores:  $M_p \approx 7\%$ ;  $T_a \approx 116[s]$ .

Com o objetivo de verificar se o modelo resultante é condizente com o processo experimental considerado, realizou-se um ensaio prático (resultado na Figura 4.8) na malha de controle associada (o programa de controle correspondente está disponível no Apêndice E). Como resultado, obteve-se a resposta dinâmica indicada na Figura 4.8 com os seguintes valores para a informação de nível do sistema:  $M_p \approx 10\%$ ;  $T_a \approx 97[s]$ . As respostas (simulada e

experimental) apresentaram características bem similares, onde as diferenças são atribuídas a discrepâncias entre a modelagem utilizada e o comportamento real do processo físico.

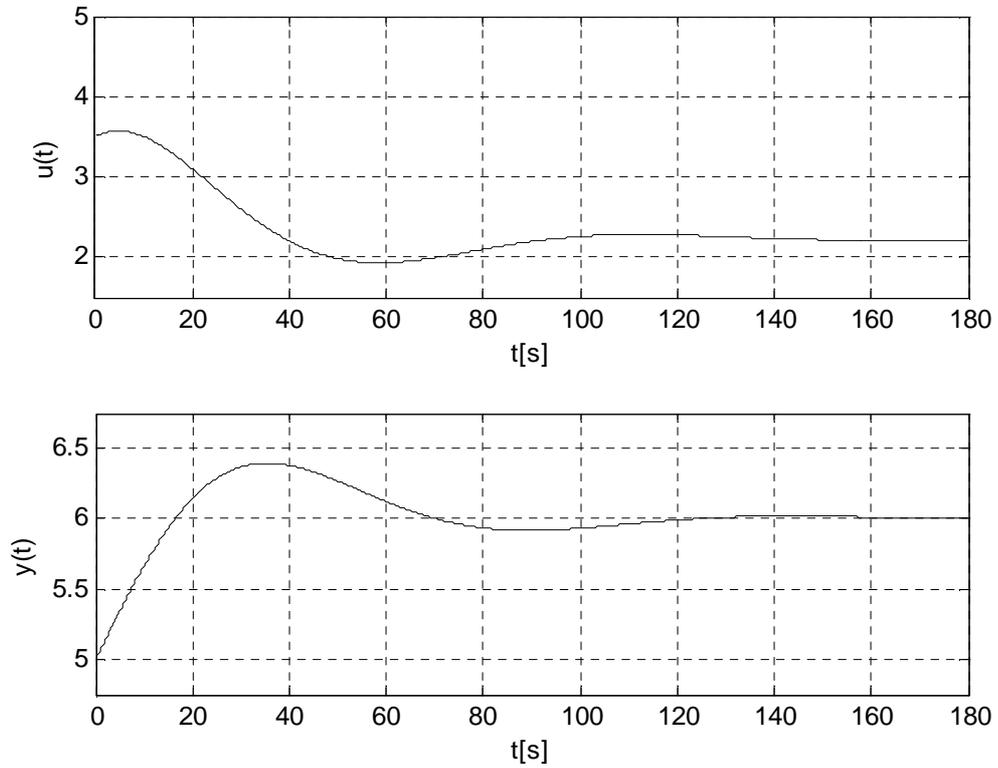


Figura 4.7 – Simulação da malha de controle considerando o modelo ARX

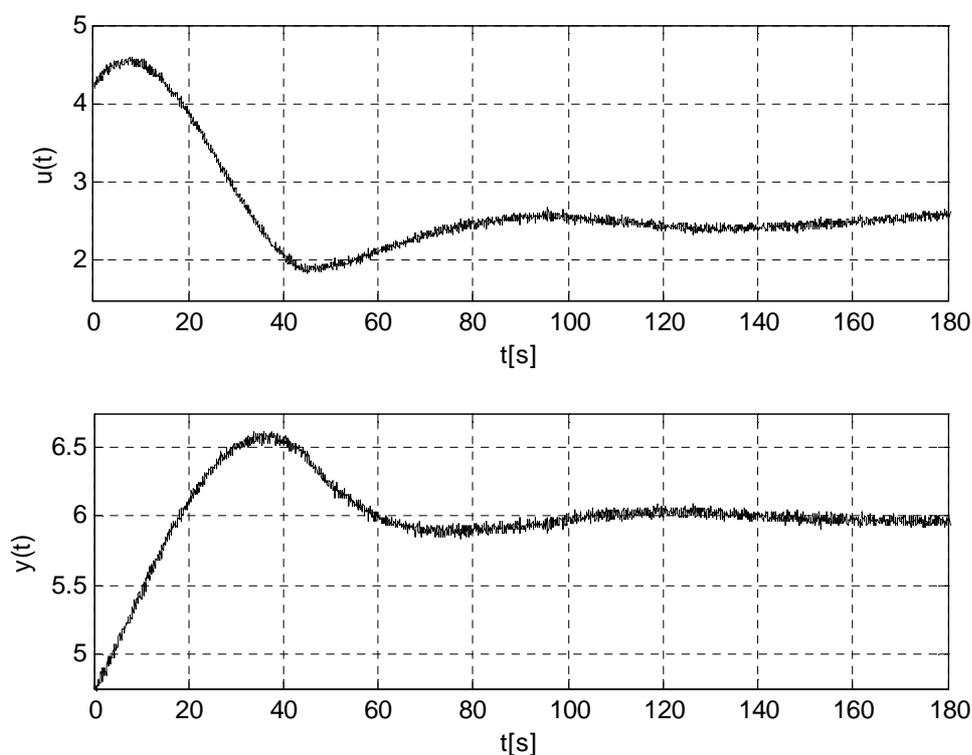


Figura 4.8 – Ensaio experimental da malha de controle

### 4.3.1 Simulações da Malha de Controle com o Modelo NARMAX e com o Controlador PI Linear Discreto para Diversos Valores de Referência

Nesta seção é considerada a resposta da malha de controle assumindo agora o modelo não-linear (NARMAX) do processo de nível e o controlador linear discreto projetado. A Figura 4.9 ilustra as respostas normalizadas ( $y/sp$ ) obtidas, considerando diferentes valores de referência de entrada ( $sp$ ) nas simulações realizadas (programa disponível no Apêndice F). Os gráficos resultantes indicam que as respostas dinâmicas obtidas se alteram com o valor de operação do sistema, ou seja, em função do valor de referência (*set point*) utilizado. Este comportamento é esperado, pois o processo possui um comportamento não-linear e o controlador utilizado possui uma ação de controle linear. Na Tabela 4.1 são indicados os valores típicos relacionados com as dinâmicas resultantes, ou seja, valores de máximo pico ( $M_p$ ) e de tempo de acomodação ( $T_a$ ).

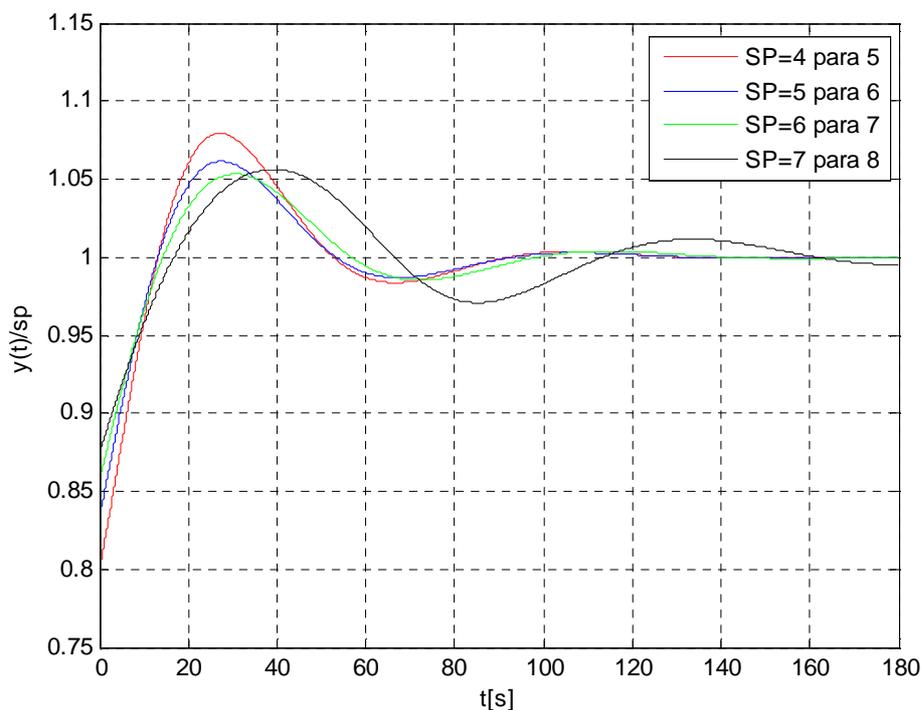


Figura 4.9 – Simulação da malha de controle com o controlador linear discreto

Tabela 4.1 – Dinâmica da malha de controle com o controlador PI linear discreto

<b>sp</b>	<b>M<sub>p</sub> (%)</b>	<b>T<sub>a</sub> (s)</b>
De 4 para 5	7,90	70,17
De 5 para 6	6,10	67,15
De 6 para 7	5,30	71,07
De 7 para 8	5,60	97,84

Com o objetivo de tentar uma melhor visualização das dinâmicas resultantes, efetuou-se outra normalização nos dados obtidos na forma  $(y - y_{min})/(sp - y_{min})$  para referenciar os gráficos em um mesmo ponto inicial, como mostrado na Figura 4.10. Porém, nesta figura como nas outras que serão mostradas neste capítulo e no próximo, este tipo normalização altera a relação real dos dados se comparadas com a normalização anterior ( $y/sp$ ) que reflete bem os valores de máximo pico e de tempo de acomodação dos dados originais.

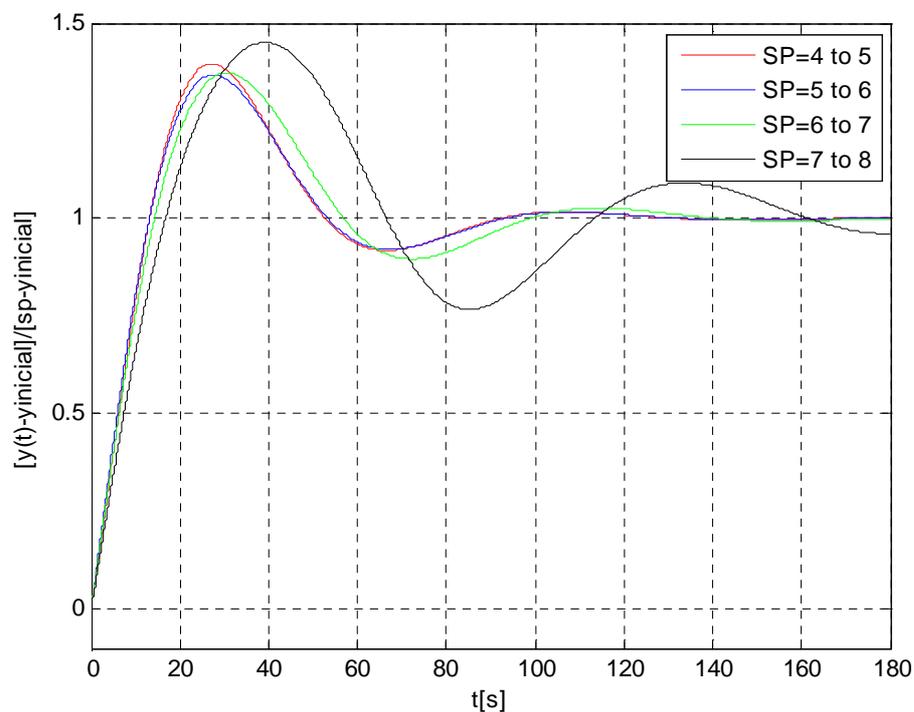


Figura 4.10 – Simulação da malha de controle com o controlador linear discreto com a outra padronização citada

Na Figura 4.11 estão ilustrados os dados associados com a ação de comando “ $u$ ”, o erro “ $e$ ” da malha de controle associada e a integração “ $Ie$ ”. Para valores de referência de entrada decrescentes, as magnitudes dos dados possuem valores invertidos em relação aos gráficos mostrados. Estas informações servirão para definir as funções de pertinência dos conjuntos difusos do controlador *fuzzy* considerado neste trabalho.

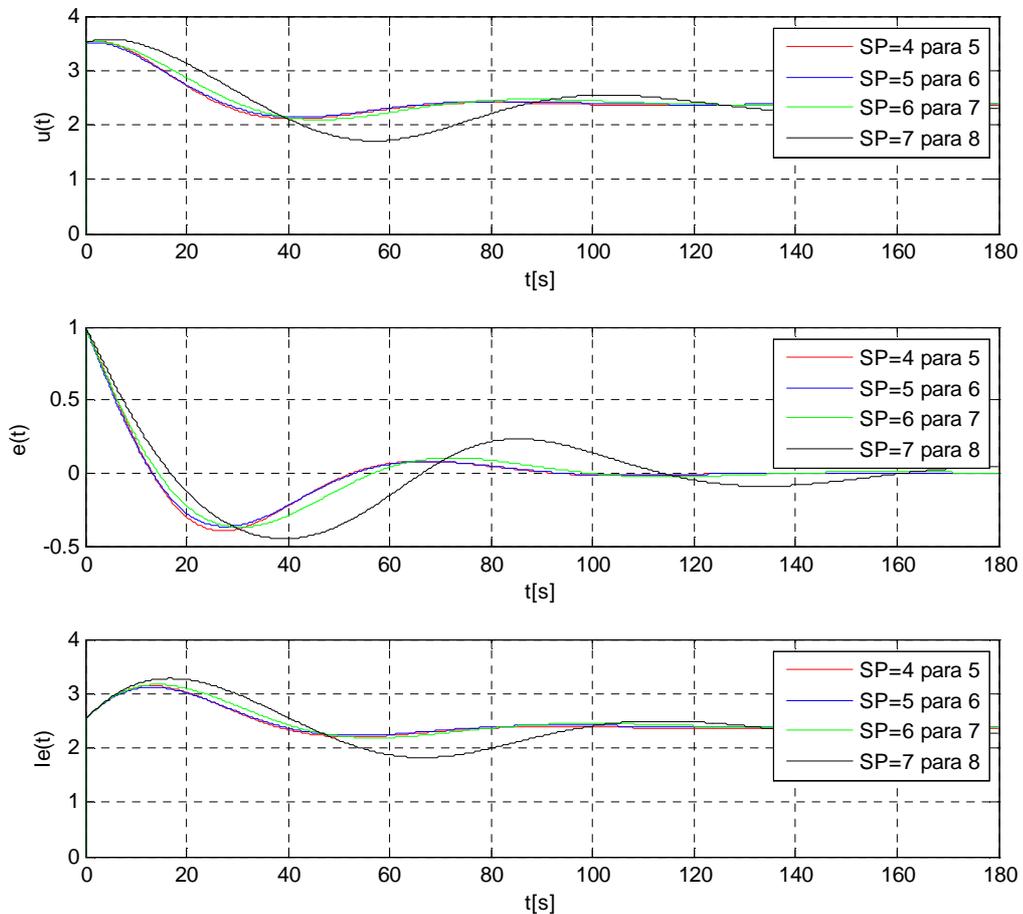


Figura 4.11 – Informações da ação de comando, do erro da malha de controle e da sua integração

#### 4.4 Projeto de um Controlador *Fuzzy* com Ação PI

Com o propósito de melhorar as respostas dinâmicas relacionadas com o controle do processo em questão, é considerado agora um controlador *fuzzy* com ação proporcional e integral. Foi mencionado no Capítulo 3 que um controlador *fuzzy* com funções de pertinência não igualmente espaçadas pode apresentar um comportamento adaptativo em relação ao erro da malha de controle associada, possibilitando um melhor desempenho nas respostas dinâmicas obtidas no controle de processos com características não-lineares ou com parâmetros variantes no tempo.

Os procedimentos de ajuste das funções de pertinência e de análise de estabilidade utilizados em controladores *fuzzy* estão disponíveis nas referências citadas do Capítulo 3, que aplicados no contexto deste trabalho resultou nos dados descritos a seguir. Para a variável de entrada “ $e$ ” associada com o erro da malha de controle, os valores modais das funções de pertinência foram ajustados em  $C_n = 5$ ,  $b = 1,25$ ,  $C_o = 0$  (Figura 3.5), e para a variável “ $v$ ” associada com a integração do erro os valores foram ajustados em  $C_n = 5$ ,  $b = 5$ ,  $C_o = 0$ . Para a variável de saída  $V_o$  associada com a ação de comando “ $u$ ” da malha de controle, os valores foram definidos por  $C_m = 10$ ,  $C_i = 5$ ,  $C_c = 0$  (Figura 3.6). Os limites dos universos de discurso adotados para as funções de pertinência definidas acima estão associados com os dados das variáveis “ $e$ ”, “ $Ie$ ” e “ $u$ ” resultantes de simulações da malha de controle em questão, cujos valores típicos estão ilustrados nos gráficos da Figura 4.10. Os fatores de escala das variáveis de entrada do controlador *fuzzy* adotado (Figura 3.4), correspondem aos valores dos ganhos calculados para a função (4.4) de compensação do PI discreto, ou seja,  $g_e = K_p = 1$  e  $g_v = K_i = 0,1$ . As regras *fuzzy* utilizadas são as mesmas expressas em (3.22).

#### **4.4.1 Simulações da Malha de Controle com o Modelo NARMAX e com o Controlador *Fuzzy* para Diversos Valores de Referência**

Para a realização das simulações desta seção foi desenvolvido um programa (código disponível no Apêndice G) que utiliza a composição produto no processamento dos valores associados aos antecedentes das regras, e o método do centro de área na etapa de defuzzificação de dados. O gráfico na Figura 4.12 ilustra os resultados obtidos, cujos dados relacionados com as dinâmicas resultantes estão sumarizados na Tabela 4.2 (verifica-se que os valores de máximo pico e de tempo de acomodação foram menores em relação às informações da malha de controle com o PI convencional). Na Figura 4.13 seguem os gráficos relacionados com a outra forma de normalização adotada.

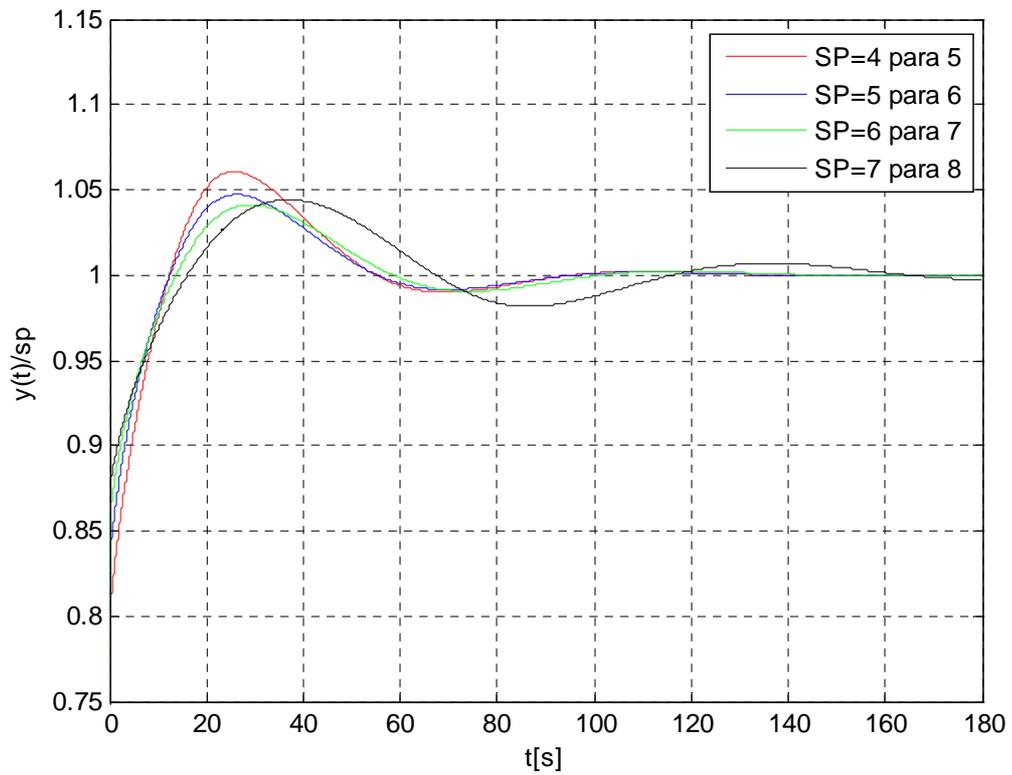


Figura 4.12 – Simulação da malha de controle com o controlador *fuzzy* com ação PI

Tabela 4.2 – Dinâmica do controlador *fuzzy* com ação PI

<b>sp</b>	<b>M<sub>p</sub> (%)</b>	<b>T<sub>a</sub> (s)</b>
De 4 para 5	6,10	67,70
De 5 para 6	4,70	68,43
De 6 para 7	4,10	72,70
De 7 para 8	4,40	85,44

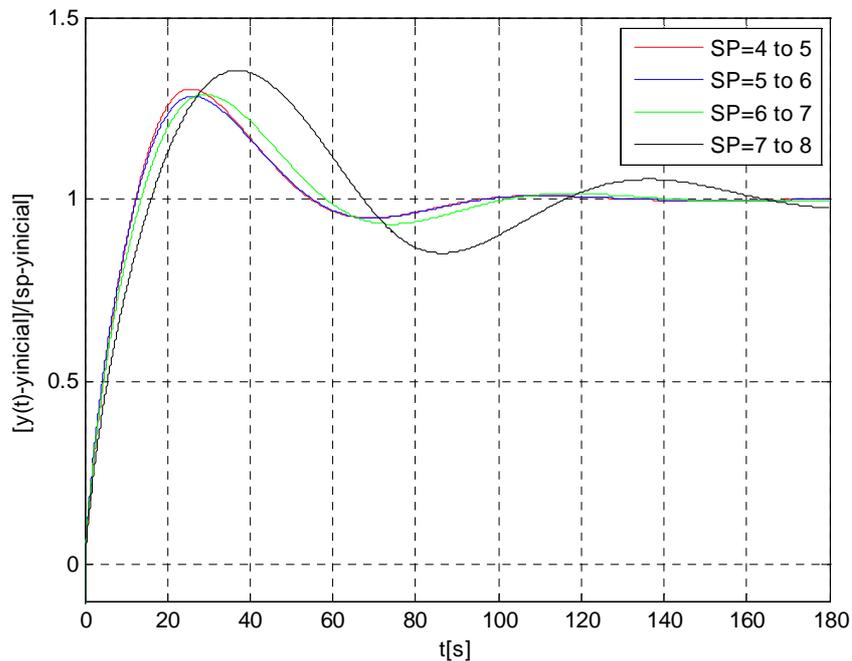


Figura 4.13 – Simulação da malha de controle com o controlador *fuzzy* com a outra padronização citada

## 4.5 Projeto de um Controlador *Rough* com Ação PI

Como alternativa para a utilização do controlador *fuzzy*, é proposto o desenvolvimento e a aplicação de um controlador *rough*, cuja vantagem consiste na ausência de procedimentos de fuzificação e defuzificação de dados, constituindo uma característica importante em aplicações em tempo real.

A metodologia empregada para a conversão de um modelo *fuzzy* em um conjunto de regras aproximadas (*rough*) foi abordada no Capítulo 3. Considerando o controlador *fuzzy* representado pelas regras em (3.22) e os valores modais definidos na seção 4.4 deste Capítulo para as funções de pertinência utilizadas, foram definidas as regiões de decomposição dos conjuntos *fuzzy* em conjuntos *crisp*, definindo-se o sistema de informação apresentado na Tabela 4.3 (onde os valores  $F(x)$  foram inferidos diretamente do modelo *fuzzy* em questão). A variável “ $e$ ” (erro) está associada à informação  $x_1$  do controlador aproximado correspondente, a variável “ $v$ ” ( $Ie$  ou integral do erro) com a informação  $x_2$ , e “ $y$ ” associado com a informação de saída do modelo em questão.

Tabela 4.3 – Sistema de informação obtido a partir de um sistema *fuzzy*

$o_i$	$x_1=e$	$x_2=Ie \text{ ou } v$	$F(x)=u$
$o_1$	[-5,00; -1,25)	[-5,00; -2,50)	[-8,40; -5,96]
$o_2$	[-1,25; 1,25)	[-5,00; -2,50)	[-7,85; 2,50]
$o_3$	[1,25; 5,00]	[-5,00; -2,50)	[0,00; 2,50]
$o_4$	[-5,00; -1,25)	[-2,50; 2,50)	[-6,05; -2,50]
$o_5$	[-1,25; 1,25)	[-2,50; 2,50)	[-5,96; 5,96]
$o_6$	[1,25; 5,00]	[-2,50; 2,50)	[2,50; 6,05]
$o_7$	[-5,00; -1,25)	[2,50; 5,00]	[-2,50; 0,00]
$o_8$	[-1,25; 1,25)	[2,50; 5,00]	[-2,50; 7,85]
$o_9$	[1,25; 5,00]	[2,50; 5,00]	[5,96; 8,40]

Denominando os intervalos dos conjuntos *crisp* como  $\alpha_1 = [-5; -1,25)$ ,  $\alpha_2 = [-1,25; 1,25)$ ,  $\alpha_3 = [1,25; 5]$ ,  $\beta_1 = [-5; -2,5)$ ,  $\beta_2 = [-2,5; 2,5)$  e  $\beta_3 = [2,5; 5]$ , pode-se escrever as regras de decisão (4.5) relacionadas com o sistema de informação correspondente (dados da Tabela 4.4 estão associados aos coeficientes das regras do controlador *rough*).

$$\begin{aligned}
r_1: & \text{IF } x_1 = \alpha_1 \text{ AND } x_2 = \beta_1 \text{ THEN } y = c_{10} + c_{11}x_1 + c_{12}x_2; \\
r_2: & \text{IF } x_1 = \alpha_2 \text{ AND } x_2 = \beta_1 \text{ THEN } y = c_{20} + c_{21}x_1 + c_{22}x_2; \\
r_3: & \text{IF } x_1 = \alpha_3 \text{ AND } x_2 = \beta_1 \text{ THEN } y = c_{30} + c_{31}x_1 + c_{32}x_2; \\
r_4: & \text{IF } x_1 = \alpha_1 \text{ AND } x_2 = \beta_2 \text{ THEN } y = c_{40} + c_{41}x_1 + c_{42}x_2; \\
r_5: & \text{IF } x_1 = \alpha_2 \text{ AND } x_2 = \beta_2 \text{ THEN } y = c_{50} + c_{51}x_1 + c_{52}x_2; \\
r_6: & \text{IF } x_1 = \alpha_3 \text{ AND } x_2 = \beta_2 \text{ THEN } y = c_{60} + c_{61}x_1 + c_{62}x_2; \\
r_7: & \text{IF } x_1 = \alpha_1 \text{ AND } x_2 = \beta_3 \text{ THEN } y = c_{70} + c_{71}x_1 + c_{72}x_2; \\
r_8: & \text{IF } x_1 = \alpha_2 \text{ AND } x_2 = \beta_3 \text{ THEN } y = c_{80} + c_{81}x_1 + c_{82}x_2; \\
r_9: & \text{IF } x_1 = \alpha_3 \text{ AND } x_2 = \beta_3 \text{ THEN } y = c_{90} + c_{91}x_1 + c_{92}x_2
\end{aligned} \tag{4.5}$$

Tabela 4.4 – Coeficientes das regras *rough*

<b>i</b>	<b>c<sub>i0</sub></b>	<b>c<sub>i1</sub></b>	<b>c<sub>i2</sub></b>
1	-4,3374	0,3253	0,4880
2	5,0875	2,0700	2,0700
3	2,0875	0,3333	0,5000
4	-2,80175	0,4733	0,3550
5	0,0000	2,3840	1,1920
6	2,80175	0,4733	0,3550
7	-2,0875	0,3333	0,5000
8	-5,0875	2,0700	2,0700
9	4,3374	0,3253	0,4880

A Figura 3.4 também serve como ilustração da estrutura do controlador aproximado proposto, basicamente a mesma do controlador *fuzzy*, porém agora na forma de regras *rough*. Os fatores de escala  $g_e$  e  $g_v$  correspondem aos valores dos ganhos  $K_p$  e  $K_i$  das ações proporcional e integral, respectivamente, e representam ampliações ou atenuações nos valores das variáveis de entrada  $E$  e  $V$ .

Com o objetivo de verificar a similaridade entre as inferências resultantes das regras difusas e das aproximadas, foram realizadas simulações com os dois modelos de controladores baseados em regras utilizando as mesmas informações de entradas, valores aleatórios dentro da faixa dos universos de discurso definidos. O resultado obtido é ilustrado na Figura 4.14, onde se verifica uma razoável similaridade entre os dados inferidos pelos modelos computacionais dos dois controladores estabelecidos. O código do programa está disponível no Apêndice H.

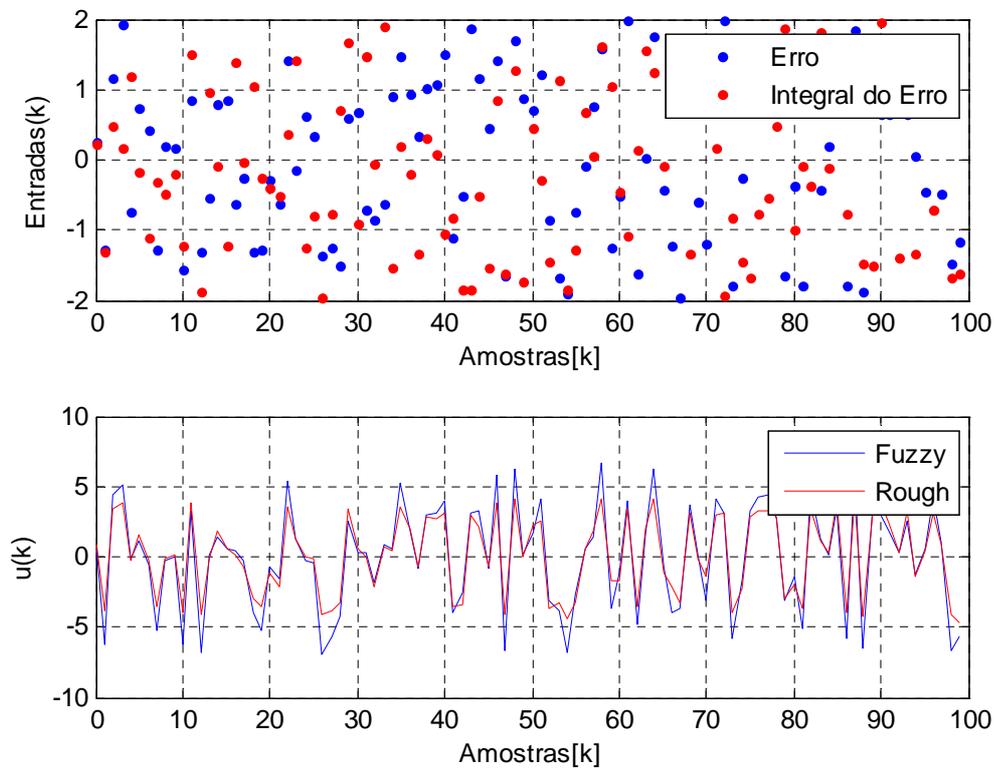


Figura 4.14 – Simulação dos modelos dos controladores (*fuzzy* e *rough*) para entradas aleatórias

Foi mensurada a relação dos tempos de processamento dos algoritmos computacionais dos controladores correspondentes, onde se verificou que o algoritmo do controlador aproximado foi na ordem de grandeza de 100 vezes mais rápido em relação ao algoritmo do controlador difuso, contexto pertinente para aplicações em tempo real. O motivo desta diferença está relacionado com a ausência de procedimentos de fuzificação e defuzificação de dados no processamento das regras *rough*.

#### 4.5.1 Simulações da Malha de Controle com o Modelo NARMAX e com o Controlador *Rough* para Diversos Valores de Referência

Para a realização das simulações desta seção foi desenvolvido um programa (código disponível no Apêndice I). Os gráficos na Figura 4.15 ilustram os resultados obtidos, cujos dados relacionados com as dinâmicas resultantes estão resumidos na Tabela 4.5 (se verifica que os valores de máximo pico de tempo de acomodação são menores em relação às

informações da malha de controle com o controlador *fuzzy*). Na Figura 4.16 seguem os gráficos relacionados com a outra forma de normalização adotada.

No próximo Capítulo serão mostrados os resultados dos ensaios práticos realizados no processo de nível real com a utilização dos controladores citados: convencional; *fuzzy*; e *rough*.

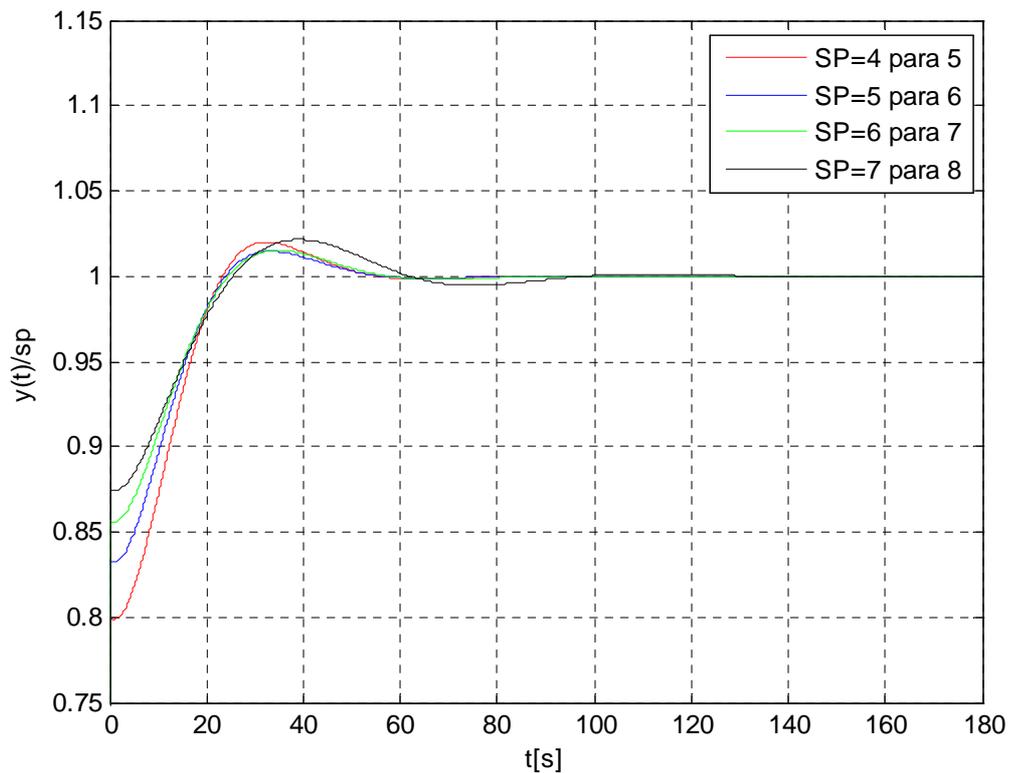


Figura 4.15 – Simulação da malha de controle com o controlador *rough* com ação PI

Tabela 4.5 – Dinâmica da malha de controle com o controlador *rough* com ação PI

<b>sp</b>	<b>M<sub>p</sub> (%)</b>	<b>T<sub>a</sub> (s)</b>
De 4 para 5	2,00	63,80
De 5 para 6	1,50	64,20
De 6 para 7	1,50	66,60
De 7 para 8	2,10	73,10

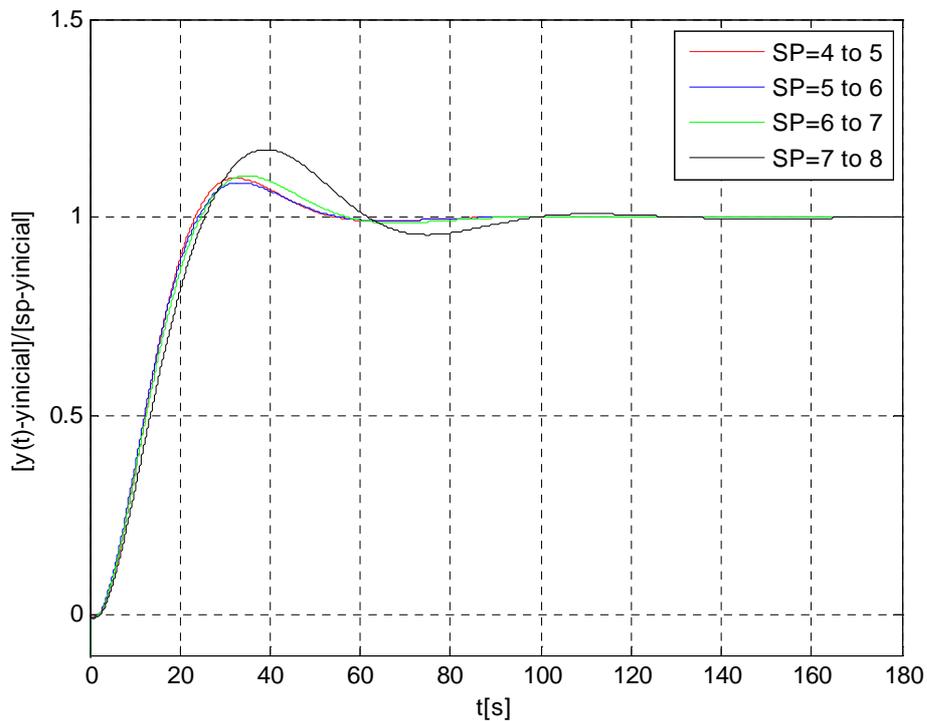


Figura 4.16 – Simulação da malha de controle com o controlador *rough* com a outra padronização incorporada

## 4.6 Análises de Estabilidade

Nesta seção é analisada a questão de estabilidade das malhas de controle consideradas no trabalho. A partir dos dados experimentais, Figuras 4.3 e 4.4, foram obtidos os modelos definidos por (4.1) e (4.2), respectivamente, um modelo ARX e outro NARMAX. A modelagem linear foi obtida com o conjunto de dados associado com a condição operacional do sistema de nível, relacionada com a alteração do estado de operação do processo de um valor nominal de 5,5 unidades de nível para 6,5 (cuja média representa a condição nominal operacional do processo de nível considerado neste trabalho). A partir do modelo (4.1) e as especificações definidas para a malha de controle com o controlador linear discreto, calculou-se os ganhos do PI associado. A condição de estabilidade nesta condição operacional é assegurada pelos valores dos polos em malha fechada. A comprovação deste contexto pode ser demonstrada através do modelo da função de transferência em malha fechada (4.6) do sistema de controle em questão, com os parâmetros indicados de forma literal.

$$\frac{y(z)}{r(z)} = \frac{K_p b_1 (z - 1) + K_i b_1 T}{z^2 + (K_p b_1 - a_1 - 1)z + a_1 + b_1 (K_i T - K_p)} \quad (4.6)$$

Substituindo na equação (4.6) os valores dos parâmetros ( $a_1 = 0,999$  e  $b_1 = 0,0027$ ) do modelo ARX (4.1), os dados dos ganhos ( $K_p = 1$  e  $K_i = 0,1$ ) do controlador PI, o valor do tempo de varredura ( $T = 0,065[s]$ ) e fatorando o denominador resultante de (4.6), têm-se os valores dos polos ( $0,9981 \pm j0,0038$ ) em malha fechada, que estão dentro do círculo unitário no domínio “z”, garantido deste modo a estabilidade para a condição operacional citada. Uma conclusão correspondente é obtida no caso de se representar a malha de controle associada em variáveis de estado, onde a modelagem do processo em questão é expressa por (4.7) para  $x(n) = y(n)$ , com a lei de controle dada por (4.8), considerando  $e(z) = r(z) - y(z)$  e a correspondência  $K_I = K_p$ .

$$x(n + 1) = a_1 x(n) + b_1 u(n) \quad (4.7)$$

$$u(z) = K_i \frac{T}{z - 1} e(z) - K_1 y(z) \quad (4.8)$$

A função de transferência em malha fechada é expressa por (4.9), onde se verifica que o denominador desta expressão é igual ao da função (4.6). Deste modo para os mesmos valores de parâmetros do modelo e de ganhos do controlador, se tem os mesmos dados dos polos em malha fechada que garantem a condição de estabilidade no contexto considerado.

$$\frac{y(z)}{r(z)} = \frac{K_i b_1 T}{z^2 + (K_p b_1 - a_1 - 1)z + a_1 + b_1 (K_i T - K_p)} \quad (4.9)$$

A condição de estabilidade também pode ser verificada com os dados dos autovalores da representação em variáveis de estado do modelo considerado em malha fechada, cujos valores são calculados via (4.10) e com os termos (expandidos para incluir a ação integral) dados por (4.11) para a modelagem considerada, sendo “ $T$ ” uma matriz identidade da mesma dimensão de “ $A$ ”.

$$|zI - A + BK| = 0 \quad (4.10)$$

$$A = \begin{bmatrix} a_1 & 0 \\ -T & 1 \end{bmatrix}; \quad B = \begin{bmatrix} b_1 \\ 0 \end{bmatrix}; \quad K = [K_1 - K_i]. \quad (4.11)$$

Para os mesmos valores de parâmetros do modelo e de ganhos do controlador, tem-se os mesmos dados dos polos em malha fechada ( $0,9981 \pm j0,0038$ ) que garantem a estabilidade. Outro modo de verificação neste tipo de representação é realizado pelo critério de Lyapunov expresso pela inequação (4.12), que pode ser reescrita com a igualdade dada por (4.13) com “ $T$ ” denotando matriz transposta e  $\alpha$  simbolizando um escalar de valor tão pequeno quanto possível. A condição de estabilidade é verificada se os autovalores da matriz “ $P$ ” (definida positiva) tiverem valores positivos.

$$[A - BK]^T P [A - BK] - P < 0 \quad (4.12)$$

$$[A - BK]^T P [A - BK] - P = -\alpha I \quad (4.13)$$

Para os mesmos valores dos parâmetros e ganhos considerados anteriormente, e assumindo  $\alpha = 0,01$ , encontra-se a matriz definida positiva que garante a estabilidade:

$$P = \begin{bmatrix} 1,36 & 0,12 \\ 0,12 & 329,29 \end{bmatrix}. \quad (4.14)$$

Para verificar a condição de estabilidade da malha de controle em questão para outras condições operacionais do sistema de nível, podem ser utilizados modelos ARX correspondentes aos pontos de operação do processo considerado. A modelagem resultante é conhecida como linear por partes (*piecewise*) e é utilizada para modelar um determinado sistema não-linear na forma de um conjunto de modelos lineares chaveados (LIN E ANTSAKLIS, 2009). Neste tipo de abordagem a equação de Lyapunov na forma de (4.12) ou (4.13) é utilizada para determinar a estabilidade do sistema estudado. Os dados relacionados com a Figura 4.4 foram utilizados na obtenção dos parâmetros da modelagem *piecewise* para

as correspondentes mudanças de estados do processo com os valores de nível em  $4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 7 \rightarrow 6 \rightarrow 5$ , obtendo-se:  $a_l = [0,9983; 0,9990; 0,9886; 0,9986; 0,9985; 0,9982; 0,9982]$ ;  $b_l = [0,0036; 0,0027; 0,0036; 0,0039; 0,0038; 0,0041; 0,0036]$ . Calculando via uma tabela de transformada  $Z$  (PHILLIPS E NAGLE, 1995) as correspondentes funções de transferência de modelagem na forma de (4.15) com “ $s$ ” simbolizando o operador de Laplace, têm-se as seguintes faixas de valores do ganho ( $K_g$ ) e da constante de tempo ( $\tau_g$ ) do modelo do processo associado:  $1,6 \leq K_g \leq 2,8$ ;  $29,5 \leq \tau_g \leq 64,9$ .

$$\frac{y(s)}{u(s)} = \frac{K_g}{\tau_g s + 1} \quad (4.15)$$

Estes valores distintos na função de modelagem do sistema de nível explicam as diferentes respostas dinâmicas da malha de controle com a utilização do controlador linear discreto. Considerando os valores dos ganhos fixos  $K_p = K_I = 1$  e  $K_i = 0,1$  nas leis de controle associadas, é possível verificar a condição de estabilidade para os diferentes pontos de operação da malha de controle em questão por meio da equação (4.13) de Lyapunov. Utilizando os dados dos parâmetros ( $a_l$  e  $b_l$  listados acima) para a modelagem linear por partes correspondente, obtém-se a matriz “ $P$ ” com dados que resultam em autovalores positivos (caracterizando a condição de estabilidade da malha de controle). Uma exemplificação é ilustrada com um par dos dados listados, onde para  $a_l = 0,999$  e  $b_l = 0,0027$  tem-se (4.14) que define a estabilidade para o sistema de controle associado.

Para a malha de controle com o controlador *fuzzy* o procedimento utilizado no PI linear discreto também pode ser aplicado. No Capítulo 3 foi citado que um controlador difuso com ação proporcional e integral possui um comportamento adaptativo em função do valor do erro (e/ou da sua integração) na malha de controle associada. Também, afirmou-se que se forem utilizadas funções de pertinência triangulares igualmente espaçadas, e considerando os valores dos fatores de escala destas funções iguais aos ganhos de um controlador linear (ou seja,  $g_e = K_p$  e  $g_v = K_i$ ), as leis de controle correspondentes são similares. Assim, utilizando os mesmos valores dos parâmetros ( $a_l$  e  $b_l$ ) da modelagem ARX linear por partes e os dados dos ganhos equivalentes do controlador *fuzzy*, é factível a obtenção de uma matriz definida

positiva “ $P$ ” comum que garanta a condição de estabilidade para os estados de operação considerados no sistema de nível.

Para a malha de controle com o controlador aproximado podem ser consideradas duas abordagens para análise de estabilidade. Na primeira delas pode-se assumir que como a conversão do modelo do controlador *fuzzy* para o *rough* corresponde a uma modelagem com ação similar e de determinada exatidão, então a análise mencionada no parágrafo anterior pode ser atribuída à malha de controle que utiliza o controlador aproximado. A segunda abordagem é aquela referenciada em Machado e Pinheiro (2013), considerando um modelo aproximado (*rough*) do processo de nível que pode ser interpretado como um conjunto de modelos lineares chaveados. Empregando técnicas de alocação de polos se calcula os ganhos ( $K_I$  e  $K_i$ ) das leis de controle correspondentes. A condição de estabilidade é verificada com a utilização dos parâmetros da modelagem do processo e dos ganhos calculados, com a obtenção de uma matriz definida positiva comum (equação de Lyapunov).

## 5 Resultados Experimentais

Neste Capítulo são apresentados os resultados experimentais referentes às estruturas dos controladores. O processo controlado é constituído pelo sistema de nível indicado, Figura 4.2, cuja estrutura correspondente está ilustrada na Figura 4.1. Espera-se que os dados resultantes dos ensaios práticos apresentem similaridades com as informações provenientes das simulações computacionais obtidas anteriormente (modelo de processo não-linear considerado), e com as respectivas modelagens dos controladores projetados: linear discreto; *fuzzy*; aproximado (todos com ações PI). As malhas de controle ensaiadas possuem uma estrutura similar ao diagrama de blocos ilustrado na Figura 5.1, onde os conversores A/D e D/A fazem parte do sistema de aquisição de dados utilizado (NI USB-6008 da National Instruments), sendo que a função de subtração e os algoritmos de controle correspondentes são executados no computador utilizado na bancada de ensaios. A função “FE” corresponde a um ajuste de escala entre os dados associados com a informação de entrada “*sp*” e o dado de referência “*r*” correspondente a variável “*y*” controlada do processo, cujo dado associado “*pv*” é adquirido pelo conversor A/D da malha de controle em questão.

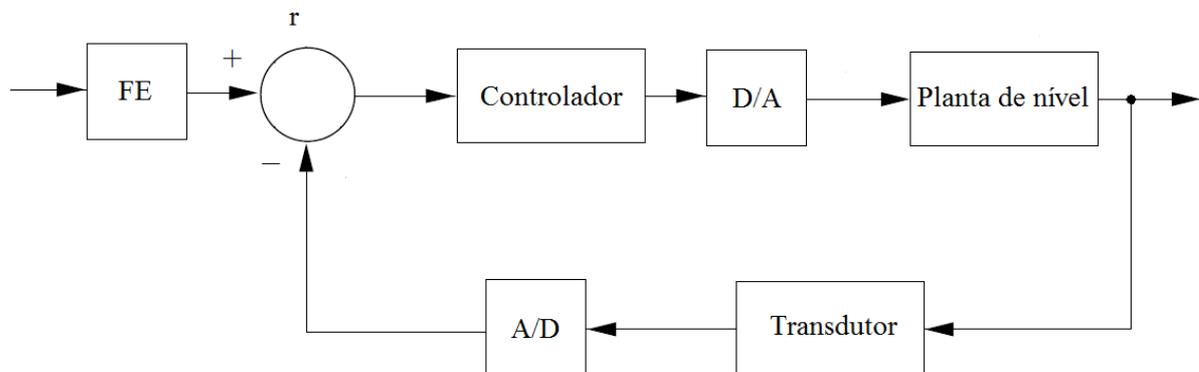


Figura 5.1–Diagrama de blocos das malhas de controle consideradas nos ensaios práticos

## 5.1 Programas de Controle e Ajustes de Escalas

Nos Apêndices E, J e L da dissertação estão disponíveis os códigos fontes dos programas desenvolvidos para as realizações práticas dos algoritmos de controle associados (linear, *fuzzy* e *rough*, respectivamente). Os algoritmos utilizam praticamente os mesmos códigos dos programas utilizados nas simulações computacionais do Capítulo anterior, onde foram inseridas instruções adicionais (específicas do MATLAB) para a leitura e escrita de dados nos conversores A/D–D/A, e também para se ler o período de tempo envolvido em cada laço de varredura de cada algoritmo utilizado. A interface D/A empregada possui uma taxa de atualização de dados em 150[Hz], resultando em um tempo real de varredura dos algoritmos dos controladores (linear, *fuzzy* e *rough*) em torno de 65[ms], inclui-se também o tempo de processamento das instruções dos programas correspondentes (que dependem do modelo do computador utilizado e das versões dos softwares empregados), e dos tempos de amostragem e conversão de dados do conversor A/D do sistema de aquisição. Neste contexto os tempos de processamentos dos três algoritmos dos controladores utilizados (linear, *fuzzy* e aproximado) apresentaram valores semelhantes (65[ms]), sendo que a diferença indicada no Capítulo 4 entre o controlador *fuzzy* e o *rough* somente é evidenciada com a utilização de uma interface D/A com maior taxa de atualização de dados.

A Tabela 5.1 contém três conjuntos de dados provenientes da interface A/D em valores de tensão do transdutor utilizado e as correspondentes informações da grandeza controlada do processo de nível (que neste caso têm indicações em *centímetros*, e cujos valores são visualizados em uma escala milimetrada fixada no reservatório do sistema). Aplicando-se o método de regressão linear é possível estabelecer a função (5.1) que fornece a relação de escala “FE” indicada na Figura 5.1, sendo que esta relação também é utilizada para associar o dado “*p<sub>v</sub>*” coletado pelo conversor A/D e a informação de nível “*y*” do processo controlado. Este fator de escala é empregado nos códigos dos controladores utilizados nos experimentos práticos, cujos valores dos resultados obtidos são apresentados posteriormente

Tabela 5.1 – Dados do sistema de nível

Dados de Entrada do A/D	Valor de Nível do Processo
2,20	2,9
4,60	4,6
6,40	6,0

$$r = -1,70581 + 1,356846 * sp \quad (5.1)$$

## 5.2 Ensaio Práticos com o Controlador PI Linear Discreto

Nesta seção é considerada a atuação da malha de controle com o controlador linear discreto projetado no Capítulo anterior, que é empregado para controlar o processo de nível em escala reduzida utilizado na bancada de ensaios. As Figuras 5.2 e 5.3 ilustram as respostas normalizadas ( $y/sp$ ) obtidas considerando diferentes valores de referência de entrada ( $sp$ ) para os experimentos realizados. Nota-se que as respostas resultantes apresentam dinâmicas diferenciadas em relação aos valores das referências utilizadas, sendo um comportamento esperado, dado que o processo possui comportamento não-linear e o controlador possui atuações lineares (proporcional e integral). A Tabela 5.2 contém valores associados com as dinâmicas resultantes, indicando que a ação do controlador linear convencional resultou em valores de máximo pico e de tempo de acomodação distintos em relação ao valor nominal da referência de entrada de operação do sistema. Nas Figuras 5.4 e 5.5 estão os gráficos relacionados com a normalização referenciada a uma origem comum. Comparando os gráficos da Figura 5.2 (resultantes dos ensaios práticos) com os gráficos da Figura 4.9 (obtidos via simulações), nota-se uma similaridade nos valores de máximo pico e de tempo de acomodação, a menos da condição da referência de entrada mudando de 7 para 8 unidades de nível, atribuída à diferença do modelo NARMAX considerado e o comportamento real do processo de nível.

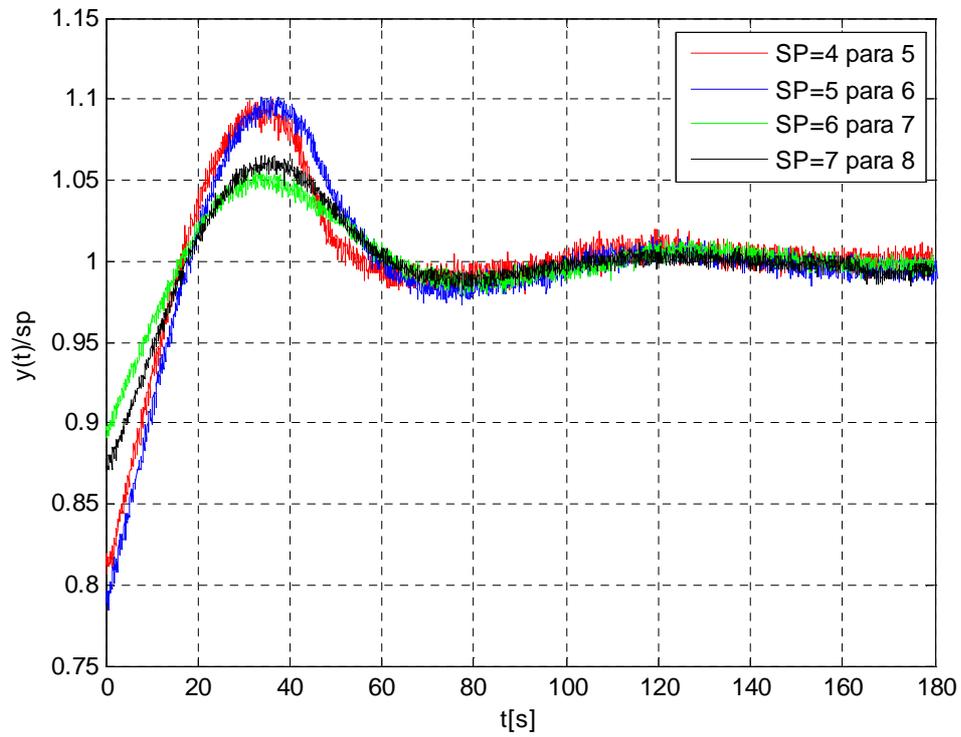


Figura 5.2 – Dados de ensaios práticos com  $sp$  crescente na malha de controle via controlador linear discreto

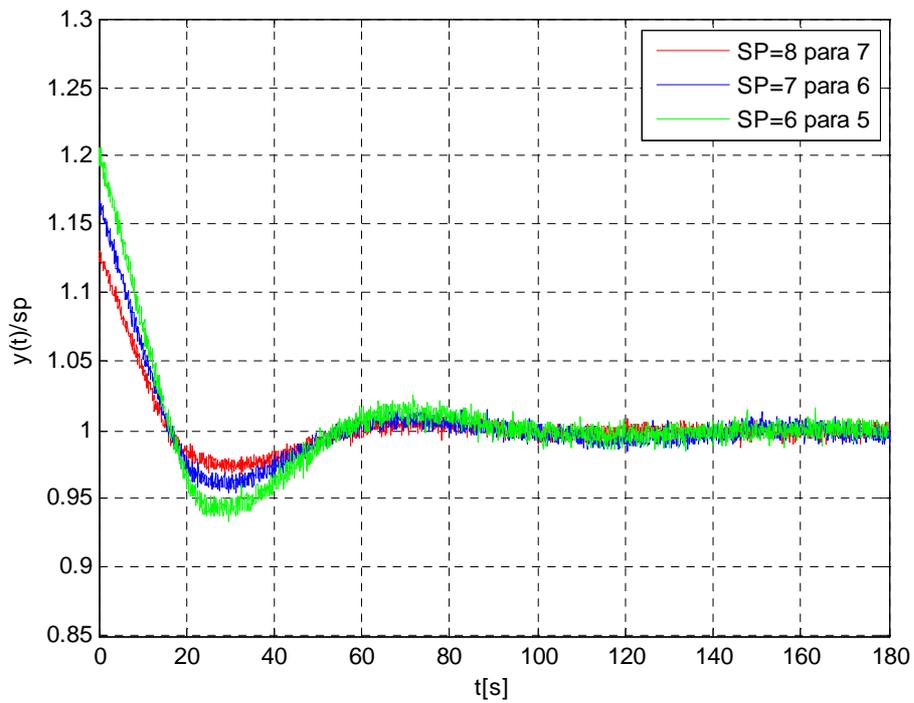


Figura 5.3 – Dados de ensaios práticos com  $sp$  decrescente na malha de controle via controlador linear discreto

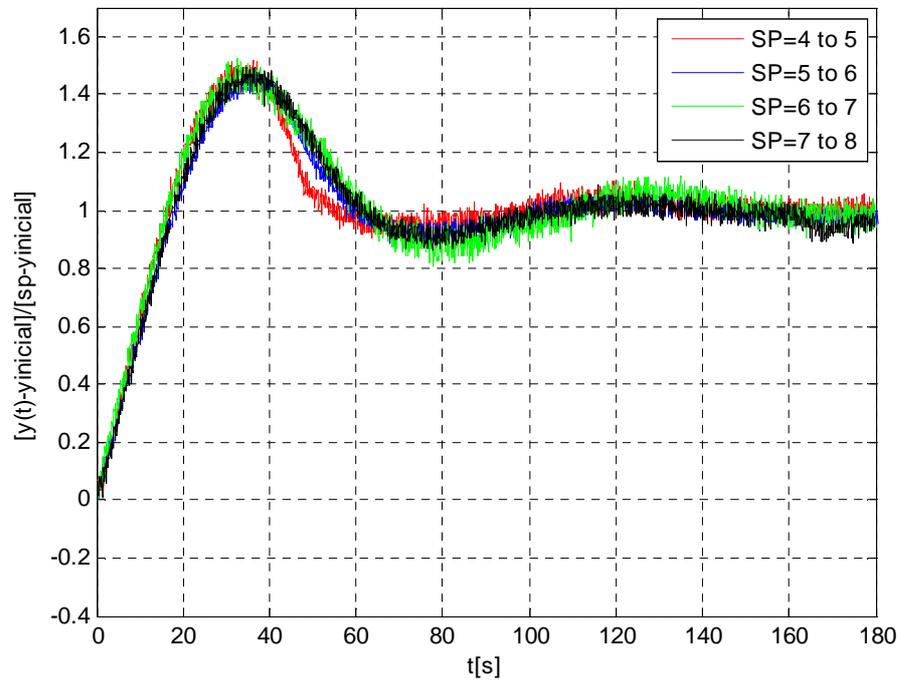


Figura 5.4 – Dados de ensaios práticos com  $sp$  crescente do controlador linear com a outra normalização citada

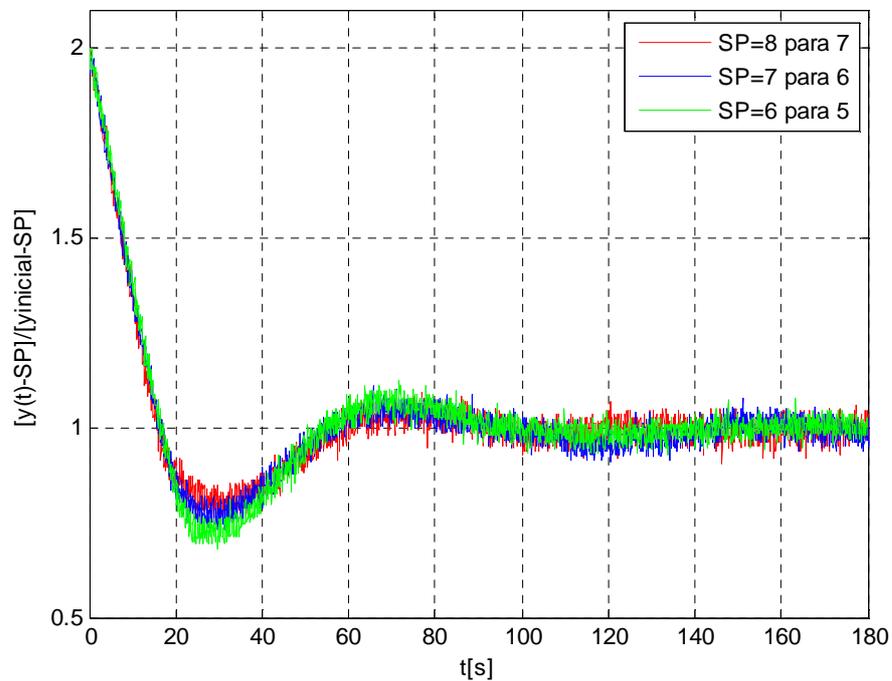


Figura 5.5 – Dados de ensaios práticos com  $sp$  decrescente do controlador linear com a origem padronizada

Tabela 5.2 – Dinâmica da malha de controle com o controlador PI linear discreto

<b>sp (crescente)</b>	<b>M<sub>p</sub> (%)</b>	<b>T<sub>a</sub> (s)</b>
De 4 para 5	9,30	70,00
De 5 para 6	9,80	73,27
De 6 para 7	5,10	72,59
De 7 para 8	6,10	76,57
<b>sp (decrecente)</b>	<b>Under-shoot (%)</b>	<b>T<sub>a</sub> (s)</b>
De 8 para 7	2,65	69,07
De 7 para 6	3,93	70,68
De 6 para 5	5,87	69,84

### 5.3 Ensaio Práticos com o Controlador *Fuzzy*

Nesta seção é utilizado o controlador *fuzzy* empregado na malha de controle do processo de nível em escala reduzida da bancada de ensaios. Os gráficos nas Figuras 5.6 e 5.7 ilustram os resultados obtidos. Os dados relacionados com as dinâmicas resultantes estão indicados na Tabela 5.3. Nas Figuras 5.8 e 5.9 estão os gráficos relacionados com a normalização referenciada a uma origem comum. Comparando os dados da Figura 5.6 (resultantes dos ensaios práticos) com os gráficos da Figura 4.12 (obtidos via simulações), nota-se uma similaridade nos valores de máximo pico e de tempo de acomodação, a menos da condição da referência de entrada mudando de 7 para 8 unidades de nível, que é atribuída à diferença do modelo não-linear adotado. Comparando os gráficos da Figura 5.6 (via controlador *fuzzy* prático com ação proporcional e integral) com os gráficos da Figura 5.2 (utilizando o controlador PI convencional), nota-se que os valores de máximo pico e de tempo de acomodação (com o compensador difuso) são em média menores em relação às respostas da malha de controle com o PI linear discreto. Os resultados práticos resultantes estão coerentes com os dados obtidos via simulações computacionais, indicando que a atuação do controlador *fuzzy* apresenta uma ação adaptativa em relação ao erro da malha de controle em questão, resultando em um desempenho melhor em relação à utilização de controlador convencional na compensação de processos com características não-lineares.

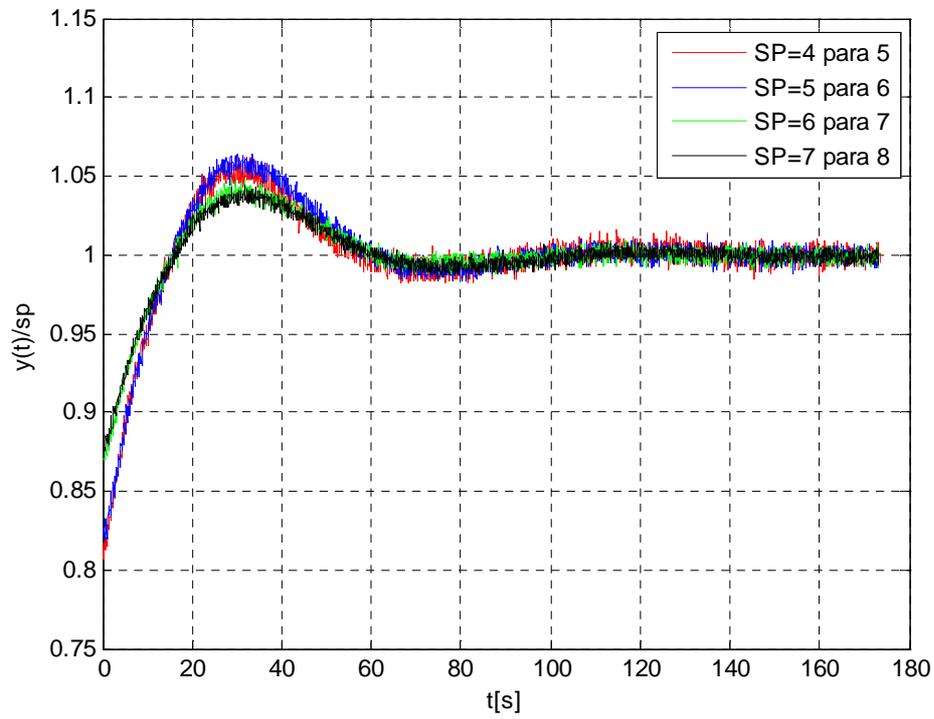


Figura 5.6 – Dados de ensaios práticos com  $sp$  crescente na malha de controle via controlador *fuzzy*

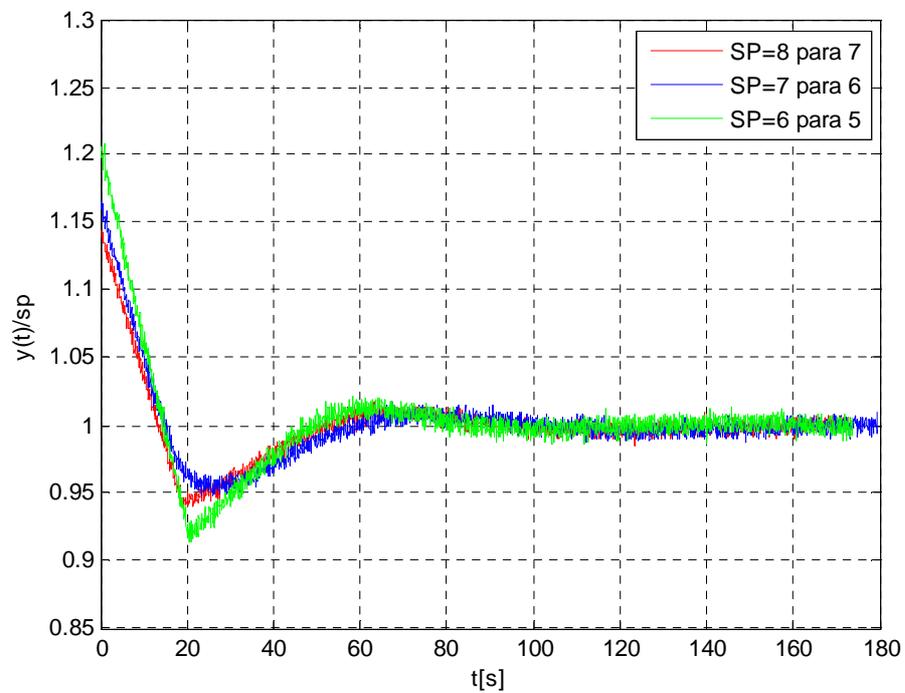


Figura 5.7 – Dados de ensaios práticos com  $sp$  decrescente na malha de controle via controlador *fuzzy*

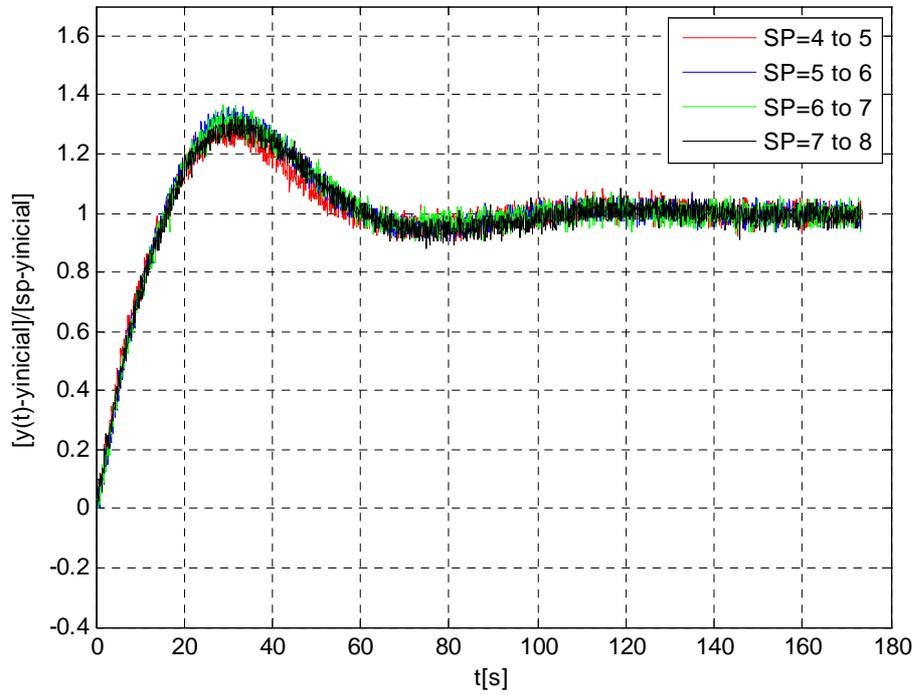


Figura 5.8 – Dados de ensaios práticos com  $sp$  crescente do controlador *fuzzy* com a origem normalizada

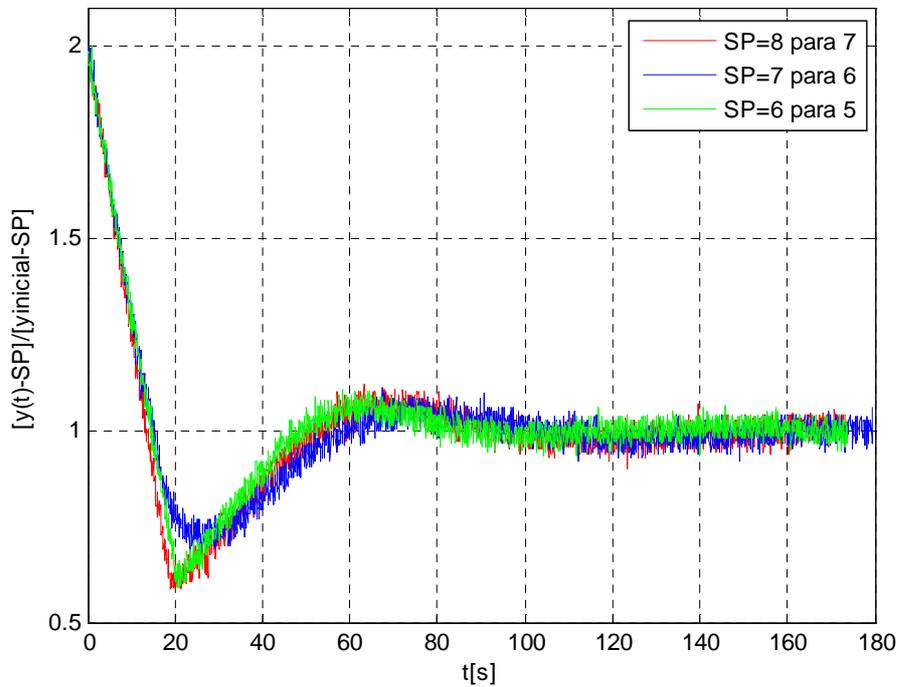


Figura 5.9 – Dados de ensaios práticos com  $sp$  decrescente do controlador *fuzzy* com a origem normalizada

Tabela 5.3 – Dinâmica do controlador *fuzzy* com ação PI

<b>sp (crescente)</b>	<b>M<sub>p</sub> (%)</b>	<b>T<sub>a</sub> (s)</b>
De 4 para 5	5,40	70,96
De 5 para 6	5,80	73,46
De 6 para 7	3,90	72,68
De 7 para 8	3,70	74,87
<b>sp (decrecente)</b>	<b>Under-shoot (%)</b>	<b>T<sub>a</sub> (s)</b>
De 8 para 7	5,65	67,79
De 7 para 6	4,68	72,57
De 6 para 5	7,82	64,71

## 5.4 Ensaios Práticos com o Controlador *Rough*

Nesta seção é utilizado o controlador aproximado empregado no controle do processo de nível em escala reduzida da bancada de ensaios. Os gráficos nas Figuras 5.10 e 5.11 ilustram os resultados obtidos. Os dados relacionados com as dinâmicas resultantes estão resumidos na Tabela 5.4. Nas Figuras 5.12 e 5.13 estão os gráficos relacionados com a normalização referenciada a uma origem comum Comparando os dados da Figura 5.10 (resultantes dos ensaios práticos) com os gráficos da Figura 4.15 (obtidos via simulações numéricas), nota-se uma boa similaridade nos valores de máximo pico e de tempo de acomodação. Comparando os gráficos da Figura 5.10 (via controlador *rough* prático com ação proporcional e integral) com os gráficos da Figura 5.6 (utilizando o controlador *fuzzy*), nota-se que os valores de máximo pico e de tempo de acomodação são menores em relação aos dados obtidos com a malha de controle com o controlador *fuzzy*. Estes resultados estão coerentes com aqueles obtidos anteriormente, entretanto, os tempos de processamento dos algoritmos de controle utilizados nos programas em tempo real apresentaram valores semelhantes devido ao tempo de atualização de dados do conversor D/A utilizado. Para que a diferença de tempos seja evidenciada entre os algoritmos do controlador *fuzzy* e o aproximado, é necessário utilizar um sistema de aquisição com maior velocidade de atualização de dados.

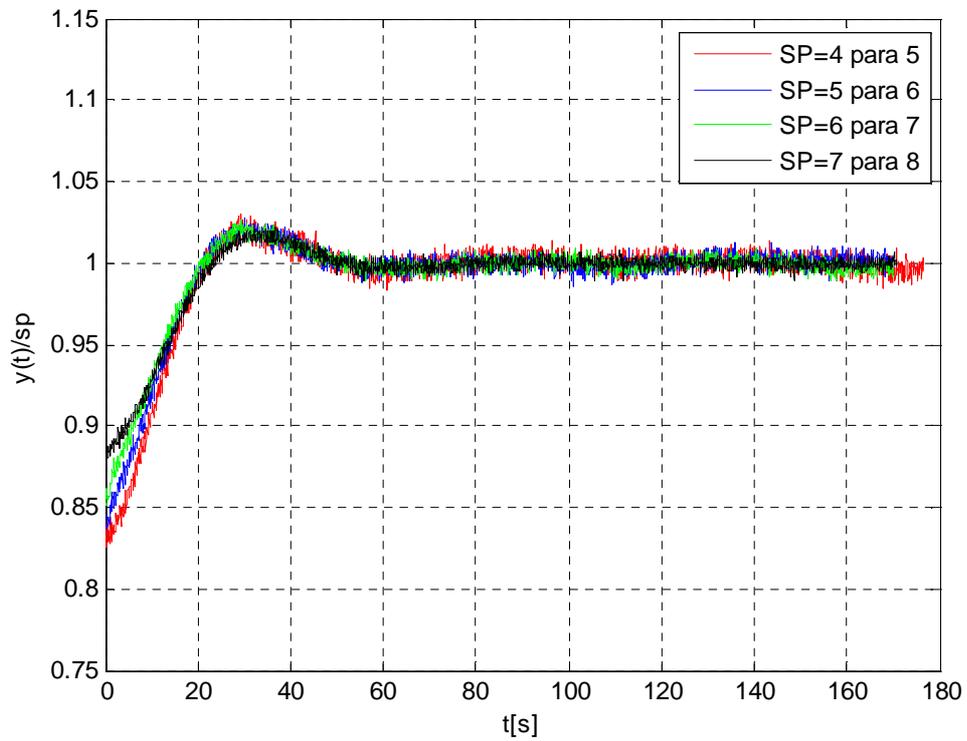


Figura 5.10 – Dados de ensaios práticos com  $sp$  crescente da malha de controle via controlador *rough*

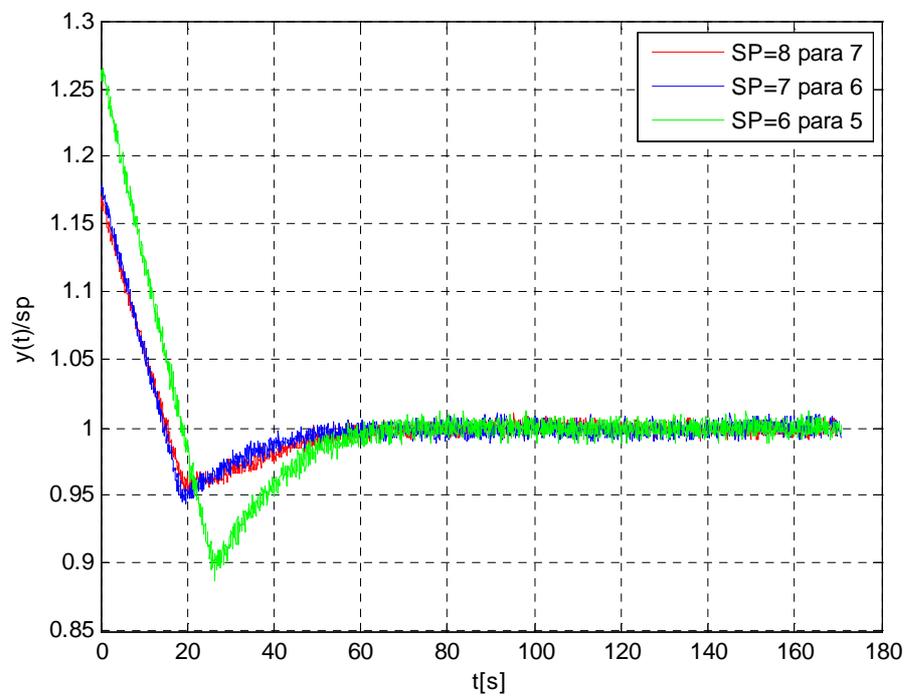


Figura 5.11 – Dados de ensaios práticos com  $sp$  decrescente da malha de controle via controlador *rough*

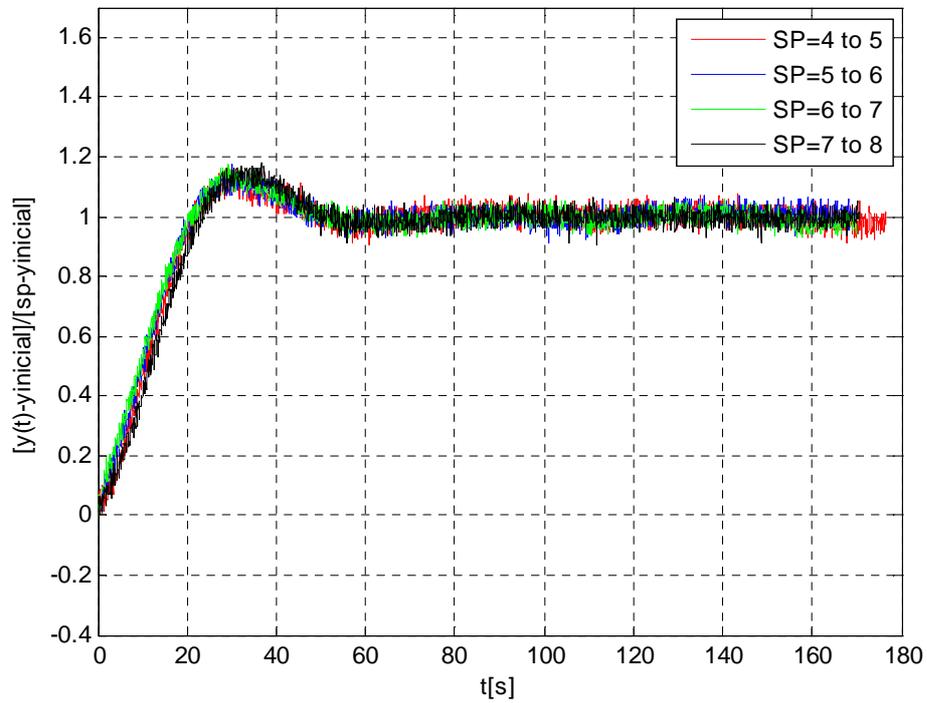


Figura 5.12 – Dados de ensaios práticos com  $sp$  crescente do controlador *rough* com a normalização adotada

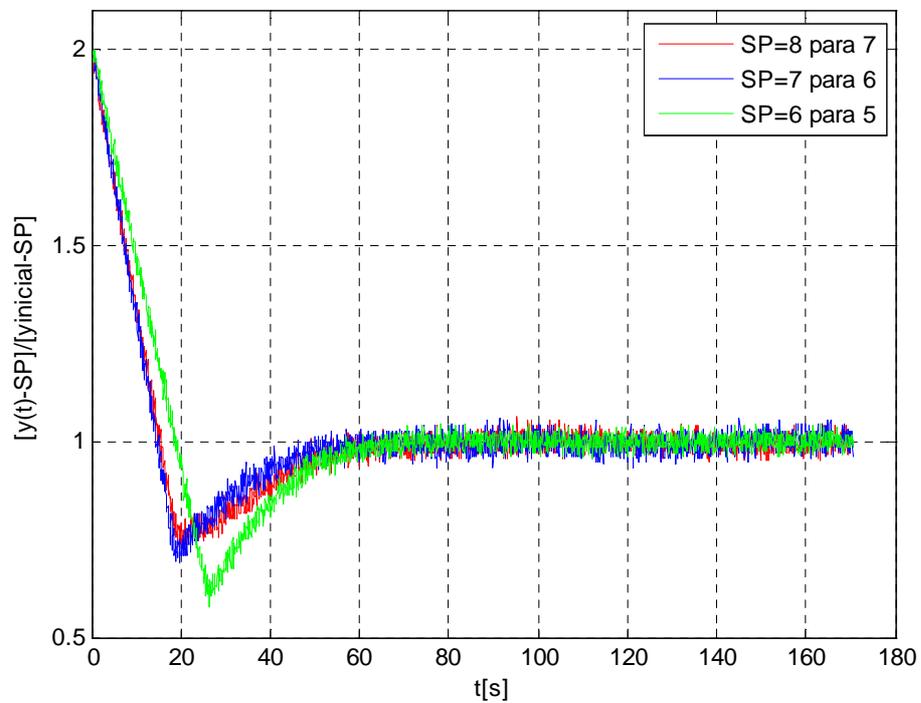


Figura 5.13 – Dados de ensaios práticos com  $sp$  decrescente do controlador *rough* com a normalização adotada

Tabela 5.4 – Dinâmica do controlador *rough* com ação PI

<b>sp (crescente)</b>	<b>M<sub>p</sub> (%)</b>	<b>T<sub>a</sub> (s)</b>
De 4 para 5	2,20	58,55
De 5 para 6	2,00	55,98
De 6 para 7	1,70	59,84
De 7 para 8	1,60	58,17
<b>sp (decrecente)</b>	<b>Under-shoot (%)</b>	<b>T<sub>a</sub> (s)</b>
De 8 para 7	4,47	50,00
De 7 para 6	5,05	50,00
De 6 para 5	9,92	50,00

## 5.5 Considerações Finais

Usualmente os modelos *fuzzy* apresentam uma boa exatidão na modelagem de funções não-lineares devido às sobreposições dos conjuntos *fuzzy* associados. Neste trabalho o modelo *rough* do controlador resultante apresentou um melhor desempenho em relação ao controlador *fuzzy* considerado, tanto nos resultados práticos como nos dados oriundos das simulações computacionais realizadas, o que foi atribuído a determinadas condições na conversão das regras associadas.

Para os valores de *under-shoot* relacionados com as respostas dinâmicas das malhas de controle consideradas nas condições de variações decrescentes de dados de referência de entrada (particularmente para magnitudes menores), não houve melhoria de desempenho do controlador aproximado em relação ao difuso e ao compensador linear discreto. Este contexto pode ser atribuído à atuação do servomecanismo do processo de nível, onde as Figuras 5.14-5.16 mostram as informações relacionadas com o dado de comando do atuador considerado que opera com valores entre 0 e 5[V]. Comparando os gráficos da Figura 5.14 (do controlador linear) com os dados das Figuras 5.15 e 5.16 (controladores *fuzzy* e *rough*, respectivamente), nota-se que não ocorreu nenhuma saturação na informação do comando “*u*” do controlador linear, mas nas ações iniciais dos controladores *fuzzy* e aproximado ocorreram saturações no nível inferior de atuação (em 0[V], e com maior intervalo de duração para o controlador *rough*), provocando atrasos nas correções de nível do processo em questão (esta condição

poderia ser minimizada usando uma tubulação de saída de maior diâmetro na planta em escala reduzida empregada).

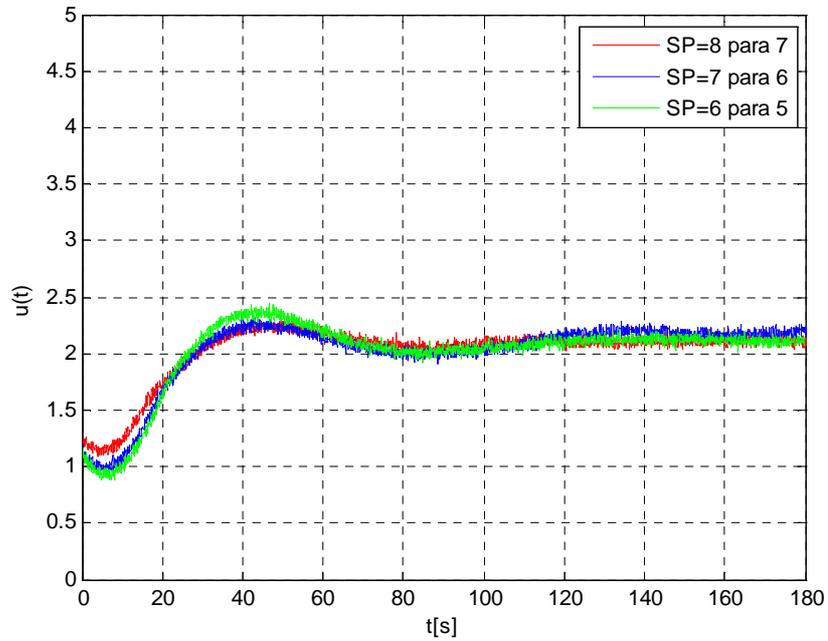


Figura 5.14 – Dados de ensaios práticos com  $sp$  decrescente com a malha de controle linear discreto

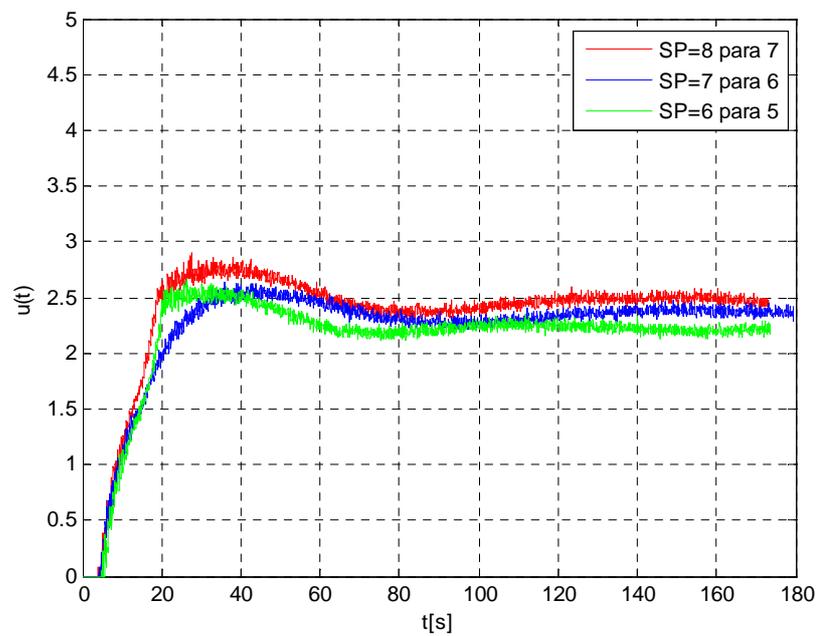


Figura 5.15 – Dados de ensaios práticos com  $sp$  decrescente na malha de controle *fuzzy*

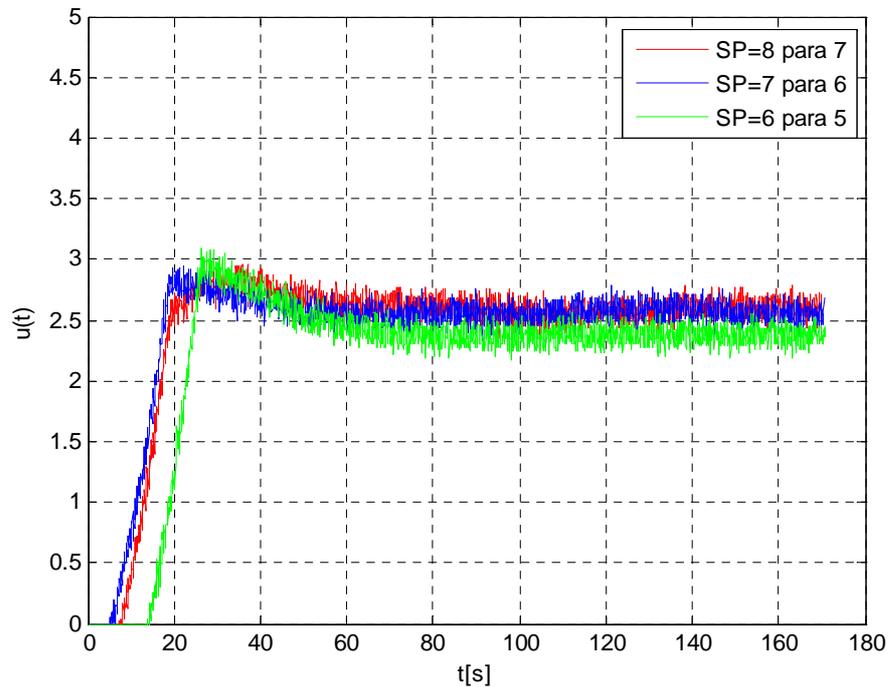


Figura 5.16 – Dados de ensaios práticos com  $sp$  decrescente da malha de controle *rough*

Um modo usual de se comparar o desempenho de malhas de controle é o índice ITAE – *Integral of Time multiplied by Absolute Error* (DORF E BISHOP, 2001), cujo valor é obtido por meio da computação da integral expressa em (5.2) em tempo contínuo, ou pela somatória em tempo discreto ( $dt = T$ ). Utilizando os dados das Figuras 5.2, 5.3, 5.6, 5.7, 5.10 e 5.11 na equação (5.2) com os valores subtraídos da unidade (referência de entrada normalizada), têm-se na Tabela 5.5 os valores calculados do índice ITAE para cada gráfico das figuras citadas, e na Figura 5.17 estes dados são apresentados graficamente. Nota-se da Tabela 5.5 e da Figura 5.17 que o controlador *rough* apresentou na maioria dos ensaios os menores índices em relação aos controladores *fuzzy* e linear discreto, uma vez que quanto menor o valor ITAE melhor é o desempenho da malha de controle associada.

$$ITAE = \int_0^{Te} t * |e(n)| dt \quad (5.2)$$

Tabela 5.5 – Índice de desempenho ITAE

sp	PI Discreto	Fuzzy	Rough
De 4 para 5	155	115	82
De 5 para 6	187	116	72
De 6 para 7	127	88	65
De 7 para 8	132	88	55
De 8 para 7	69	92	77
De 7 para 6	102	98	76
De 6 para 5	123	114	131

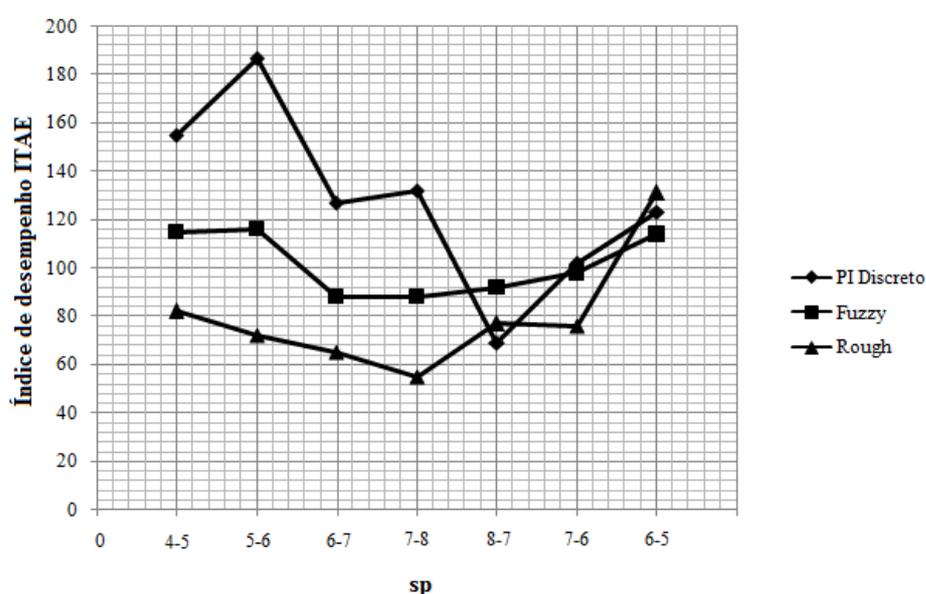


Figura 5.17 – Índice ITAE em relação às variações de referência de entrada

A contribuição principal deste trabalho permitiu mostrar a potencialidade da aplicação de controladores baseados em regras aproximadas em aplicações práticas. Particularmente na proposta apresentada nesta dissertação da utilização de um controlador *rough* com ação proporcional e integral, que neste trabalho foi obtido via conversão de um modelo *fuzzy* associado. Os resultados obtidos indicaram que o desempenho resultante foi igualmente satisfatório em relação à utilização de um controlador baseado em regras com ação de avanço ou atraso de fase, aplicado na compensação de um estabilizador de sistema de potência estudado por Silva, Pinheiro e Mendes (2014).

No próximo Capítulo encontra-se a conclusão desta pesquisa e a sugestão de trabalhos futuros no contexto apresentado nesta dissertação.

## 6 Conclusões

Como alternativa aos procedimentos convencionais utilizados em sistemas de controle típicos, este trabalho teve como finalidade realizar comparações de desempenho de malhas de controle associadas a um processo não-linear, por meio da aplicação de três tipos de controladores: linear discreto; *fuzzy*; aproximado (*rough*). O objetivo principal foi o desenvolvimento de um controlador baseado em regras aproximadas com ação proporcional e integral, cuja finalidade foi controlar um sistema com comportamento não-linear com um desempenho melhor ou similar em relação a um controlador *fuzzy* correspondente.

Na metodologia adotada inicialmente foram obtidos modelos matemáticos que representaram em determinados contextos o processo não-linear considerado, um processo de nível em escala reduzida, cujos dados oriundos de ensaios práticos possibilitaram o desenvolvimento das modelagens utilizadas neste trabalho. O modelo (ARX) linear obtido serviu como base para o cálculo dos ganhos do controlador linear discreto utilizado, e o modelo não-linear (NARMAX) resultante serviu para estimar as respostas dinâmicas a serem obtidas em condições mais próximas da realidade. As abordagens de projeto e desenvolvimento dos controladores baseados em regras utilizados nesta pesquisa foram realizadas através de simulações computacionais com a finalidade de validar os procedimentos desenvolvidos e as análises efetuadas. Para o controlador difuso foram alteradas características das funções de pertinência utilizadas com o objetivo de se obter ações de controle mais efetivas. Posteriormente, utilizando um procedimento apropriado foram obtidas as regras do controlador aproximado a partir do modelo do compensador *fuzzy*. Na etapa final da metodologia foram realizados experimentos práticos no sistema de nível controlado em tempo real via os algoritmos de controle desenvolvidos. Os dados obtidos experimentalmente comprovaram as análises teóricas realizadas. E com os resultados práticos coletados verificou-se um melhor desempenho na aplicação real da abordagem proposta.

Resumindo, o controlador linear discreto apresentou como esperado um desempenho inferior em relação aos controladores baseados em regras, pois a sua aplicação em processos com comportamentos não-lineares resulta em desempenhos não adequados para contextos operacionais distintos de um determinado valor nominal de projeto. Os compensadores

baseados em regras, utilizando os mesmos valores dos ganhos do controlador linear discreto foram mais eficientes em suas ações de controle. Os valores de máximo pico e de tempo de acomodação associados com as respostas dinâmicas da malha de controle com o controlador *fuzzy*, em geral, foram menores em comparação às respostas da malha de controle com o PI linear discreto. Os resultados práticos foram coerentes com os dados obtidos via simulações computacionais, indicando que a atuação do controlador *fuzzy* apresenta uma ação adaptativa em relação ao erro da malha de controle em questão, resultando em um desempenho melhor quando comparado à utilização de um controlador convencional na compensação de processos com características não-lineares. O controlador *rough* (derivado do modelo difuso) e aplicado no controle do processo de nível, apresentou valores de máximo pico e de tempo de acomodação menores na maioria dos ensaios realizados, onde foram realizadas variações repentinas na referência de entrada da malha de controle associada. O melhor desempenho geral do controlador aproximado em relação ao *fuzzy* é atribuído ao mapeamento da função de controle original e as regras resultantes dos controladores *rough*, as quais apresentam uma adequada capacidade de interpolação dos dados provenientes das ações de comando oriundas da informação dos sistemas real. Foi utilizado o índice ITAE para comparar o desempenho das malhas de controle associadas aos três tipos de controladores considerados, comprovando os resultados comentados nos parágrafos anteriores.

Nesta pesquisa também foi comparado o tempo de processamento das regras *fuzzy* e as aproximadas, onde constatou que o algoritmo do controlador *rough* apresentou um tempo de processamento, aproximadamente, da ordem de grandeza 100 vezes menor em relação ao algoritmo do controlador *fuzzy*. Esse valor consideravelmente menor no tempo de processamento é devido à ausência de procedimentos de fuzificação e defuzificação de dados no modelo do controlador aproximado, o que constitui uma característica importante em aplicações de tempo real. Na realização computacional prática dos algoritmos de controle esta diferença não foi considerável, pois o conversor D/A utilizado nos experimentos práticos possuía uma taxa de atualização de dados de 150[Hz].

Essa pesquisa destacou a potencialidade da aplicação de controladores baseados em regras aproximadas em aplicações práticas.

Com o intuito de dar continuidade aos estudos realizados nesta dissertação, são apresentadas algumas sugestões para trabalhos futuros:

- Aplicação dos conceitos utilizados nesta pesquisa no controle de processos reais de maior complexidade, por exemplo, em sistemas com múltiplas entradas e saídas que apresentem não linearidades.

- A utilização de processadores digitais de sinais ou sistemas com estruturas programáveis em FPGA para a realização dos algoritmos de controle desenvolvidos, ao invés de utilizar um computador acoplado a uma placa de aquisição de dados de velocidade limitada. Deste modo será possível testar a diferença dos tempos de computação dos algoritmos de controle considerados em plataformas de alto desempenho e com boas relações de custos/benefícios.

## Apêndice A – Telas de Controle e Supervisão do Programa de Controle Linear com Ação PI

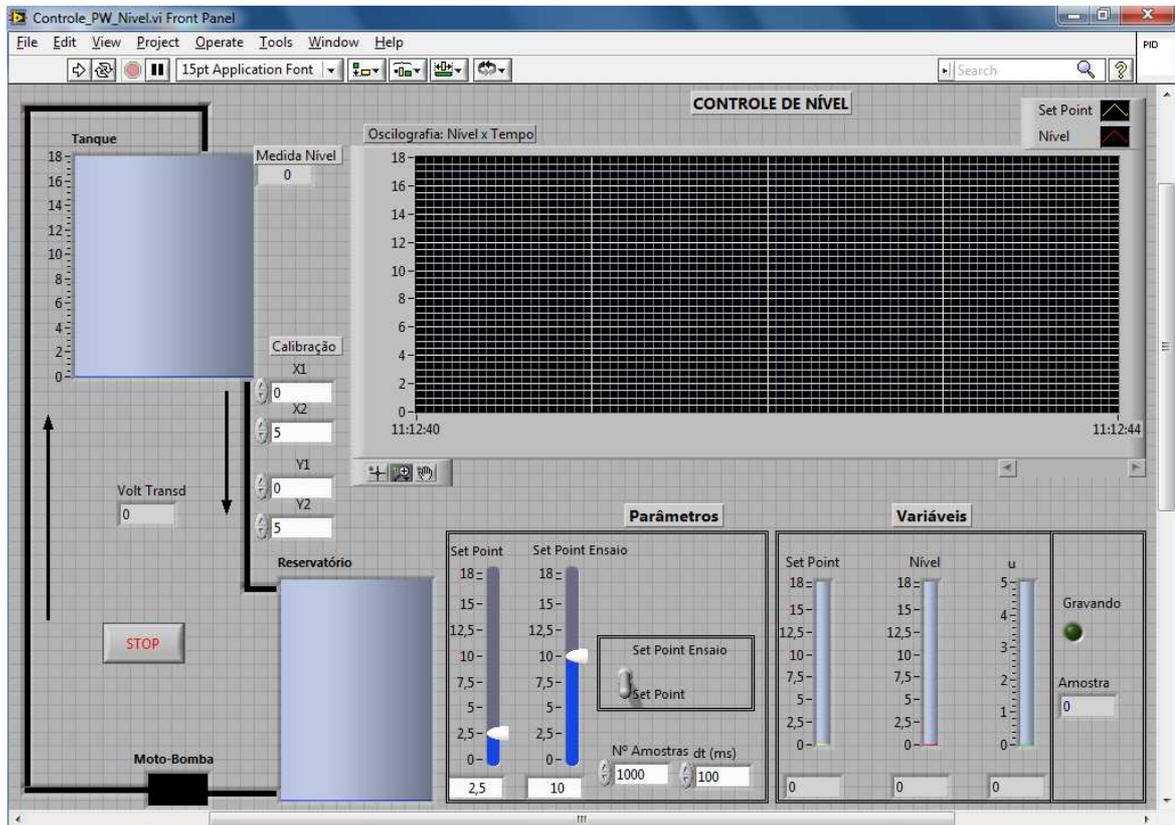


Figura A1 – Painel frontal

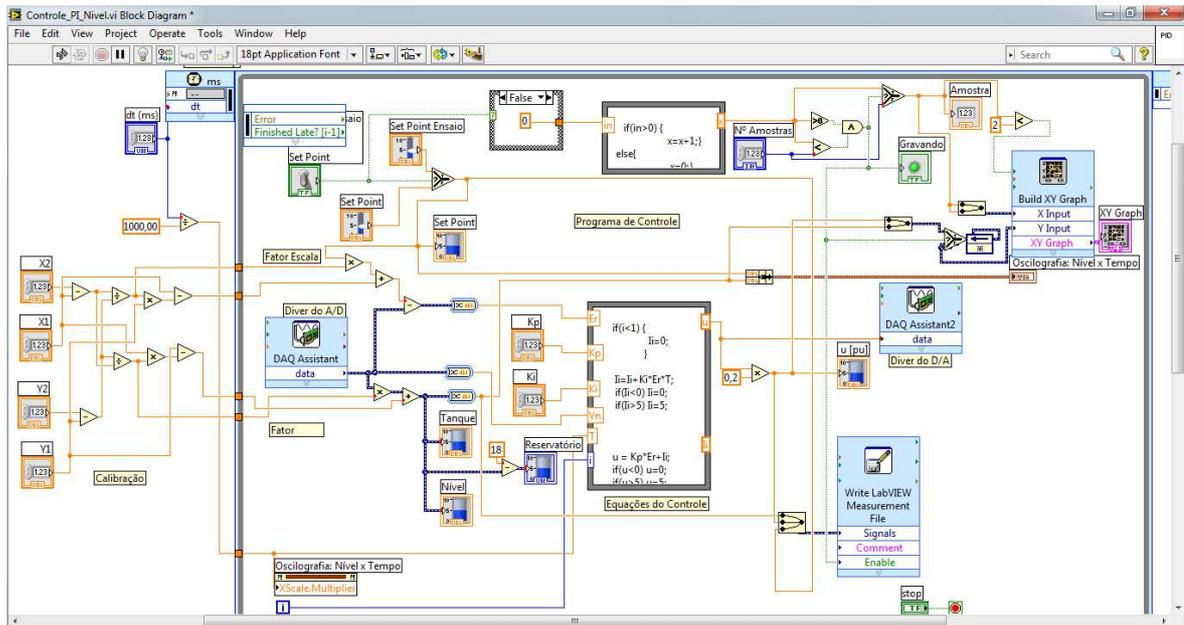


Figura A2 – Diagrama de blocos

## Apêndice B – Programa de Estimação do Modelo ARX

```

% Obtenção modelo ARX

clear all

load ensaio_55_65.xls;
t = ensaio_55_65(:,1); ym = ensaio_55_65(:,2); um = ensaio_55_65(:,4);

u=um(1:750); y=ym(1:750);

r = [y u];
modelo = arx(r,[1 1 1]);
[b,a] = th2tf(modelo)

% Teste modelo ARX
m=2768;

%Dados para validação
iv=um(1:2768); %Entrada
ov=ym(1:2768); %Saída

%Validação
S1(1)=ov(1); S1(2)=ov(2); S1(3)=ov(3);
for k=4:m
    S1(k)=0.999*S1(k-1) + 0.0027*iv(k-1);
end
S1=S1'; %Dados de saída simulado
plot(1:m,S1,'k-.',1:m,ov,'k'); grid; xlabel('Amostras [k]');
ylabel('y(k)'); legend('Modelo ARX','Dados Medidos');
axis([0 2768 5.25 7.25]);

%Autocorrelação do sinal simulado com as medidas de saída
rr=xcorr(S1,ov); %AutoCorrelação dos dados simulados e medidas de saída
Crr=corrcoef(S1,ov);
crr=Crr(1,2)*100; %Coeficiente de correlação entre os dados simulados
e os medidos
'Coeficiente de Correlação do Modelo ARX'
crr
figure;
plot(-2767:2767,rr); grid; xlabel('Número de Amostras[k]');
ylabel('AutoCorrelação do sinal medido com modelo ARX');
title('Autocorrelação');

```

## Apêndice C - Programa de Estimação do Modelo NARMAX

```

%Estimação de modelo NARMAX

clear all

n=4996; %Número de amostras
m=n+3; %Indexador

%Dados para identificação
load ensaio.xls;
t = ensaio(:,1); ym = ensaio(:,2); um = ensaio(:,4);

in=um(1:4999); %Entrada
out=ym(1:4999); %Saída

%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 parâmetros por MQ
P = inv(F'*F);
Q = F'*Y;
T = P*Q; %Vetor de parâmetros estimados
Teta = T;
'Vetor de Parametros Estimados'
format short g
Teta

%Dados para validação

iv=um(5000:9998); %Entrada
ov=ym(5000:9998); %Saída

%Validação
S1(1)=ov(1); S1(2)=ov(2); S1(3)=ov(3);
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 saída simulado
plot(1:m,S1,'k-.',1:m,ov,'k'); grid; xlabel('Amostras [k]');
ylabel('y(k)'); legend('Modelo NARMAX','Dados Medidos');

%Autocorrelação do sinal simulado com as medidas de saída
rr=xcorr(S1,ov); %AutoCorrelação dos dados simulados e medidas de saída
Crr=corrcoef(S1,ov);
crr=Crr(1,2)*100;      %Coeficiente de correlação entre os dados simulados
e os medidos
'Coeficiente de Correlação do Modelo NARMAX'
crr
figure;
plot(-4998:4998,rr); grid; xlabel('Número de Amostras[k]');
ylabel('AutoCorrelação do sinal medido com modelo NARMAX');
title('Autocorrelação');

```

## Apêndice D – Programa de Simulação da Malha de Controle com Ação Linear

```

% Teste malha de controle com modelo ARX e PI discreto

clear all

sp=input('sp=');

% Condições iniciais
y(1)=sp-1;y(2)=sp-1;y(3)=sp-1;
u(1)=2.5;u(2)=2.5;u(3)=2.5;
I(1)=2.5;I(2)=2.5;I(3)=2.5;
np = 2804;

% Ganhos da lei de controle
Kp = 1; Ki = 0.1;
T=0.0642; % Tempo de varredura
t=0;

% Simulação da malha de controle com o modelo ARX
for n = 4:np

    vt(n)=t;
    t=t+T;

    r(n) = sp;
    e(n-1) = sp - y(n-1);    % Erro da malha

    I(n) = I(n-1) + Ki*e(n-1)*T; % Ação integral e anti-windUp
    if (I(n) > 5)
        I(n) = 5;
    end
    if (I(n)< 0)
        I(n)= 0;
    end

    u(n)= Kp*e(n-1) + I(n-1); % Lei de controle e saturação
    if (u(n) < 0)
        u(n) = 0;
    end
    if (u(n) > 5)
        u(n) = 5;
    end

    % Modelo ARX
    y(n) = 0.999*y(n-1)+0.0027*u(n-1);

    % Atualização de variáveis
    y(n-1) = y(n);

```

```
u(n-1) = u(n);

% Armazenamento de valores
vy(n)= y(n);
vu(n)= u(n);
vn(n) = n;
rf(n)=r(n);

end

% Gráficos resultantes da simulação
figure(1)
plot(vt,rf); grid;
xlabel('t'); ylabel('r');
figure(2)
subplot(211);
plot(vt,vu,'k'); grid;
xlabel('t[s]'); ylabel('u(t)');axis([0 180 1.5 5]);
subplot(212);
plot(vt,vy,'k'); grid;
xlabel('t[s]'); ylabel('y(t)');axis([0 180 4.75 6.75]);
```

## Apêndice E – Programa do Experimento Prático da Malha de Controle via Controlador PI Linear Discreto

```

% Programa PIZ NIDAQ

clear all;

% Configurações do sistema de aquisição de dados
ai = analoginput('nidaq','Dev1');
addchannel(ai, 0);
ai.SampleRate = 300;
ai.SamplesPerTrigger = 2;
set(ai, 'InputType', 'SingleEnded');
ao = analogoutput('nidaq','Dev1');
addchannel(ao,0);

% Condições iniciais
In=2.5;
Na = 2804; % Número de amostras do gráfico

% Parâmetros da lei de controle
Kp = 1; Ki = 0.1;
T=0.0642;

% Laço do algoritmo de controle
n=1; % Indexador

sp=input('sp=');

% Conversão SP(n_tanque) para v_gráfico(v)
spv = -1.70581 + 1.356846*sp;

while n <= Na,
    tic

    % Comandos para leitura de dados
    start(ai)
    wait(ai,0.04);
    PV = getdata(ai);
    pv=Pv(1);

    % Conversão v_gráfico(v) para SP(n_tanque)
    pvh = 1.259459 + 0.736486*pv;

    vd(n) = pvh; % Registro dos dados coletados

    en = spv - pv; % Erro da malha

```

```

In = In + Ki*en*T; % Ação integral e anti-windUp
if (In > 5)
    In = 5;
end
if (In < 0)
    In = 0;
end

un = Kp*en + In; % Lei de controle e saturação
if (un < 0)
    un = 0;
end
if (un > 5)
    un = 5;
end

putsample(ao,un);

vn(n) = n-1;
vu(n) = un;

n = n + 1; % Incrementa indexador dos dados
T = toc;
vtoc(n)=T;

%Armazena variáveis
vP(n)= Kp*en;
vIn(n)= In;

end

vt=vtoc(1:Na);
vt(1)=vtoc(2);

t=0;
for k=1:Na
    vT(k) = t;
    dt = vt(k);
    t = t + dt;
end

vP=vP(1:Na);
vIn=vIn(1:Na);

% Gráficos
figure(1)
plot(vT,vd,'b');grid;
xlabel('t'); ylabel('y');
figure(2)
plot(vT,vu,'b');grid;
xlabel('t'); ylabel('u');

figure(3)

```

```
plot(vT,vP,'b');grid;  
xlabel('t'); ylabel('P');  
figure(4)  
plot(vT,vIn,'b');grid;  
xlabel('t'); ylabel('I');
```

## Apêndice F – Programa de Simulação da Malha de Controle com o PI Linear Discreto

```

% Simulação da malha de controle com modelo NARMAX e PI discreto

clear all

sp=input('sp=');

% Condições iniciais
y(1)=sp-1;y(2)=sp-1;y(3)=sp-1;
u(1)=2.5;u(2)=2.5;u(3)=2.5;
I(1)=2.5;I(2)=2.5;I(3)=2.5;
np = 2804;

% Ganhos da lei de controle
Kp = 1; Ki = 0.1;
T=0.0642; % Tempo de varredura
t=0;

% Simulação da malha de controle com o modelo NARMAX
for n = 4:np

    vt(n)=t;
    t=t+T;

    r(n) = sp;
    e(n-1) = sp - y(n-1);    % Erro da malha

    I(n) = I(n-1) + Ki*e(n-1)*T; % Ação integral e anti-windUp
    if (I(n) > 5)
        I(n) = 5;
    end
    if (I(n)< 0)
        I(n)= 0;
    end

    u(n)= Kp*e(n-1) + I(n-1); % Lei de controle e saturação
    if (u(n) < 0)
        u(n) = 0;
    end
    if (u(n) > 5)
        u(n) = 5;
    end

    % Modelo NARMAX
    y(n) = 0.51469*y(n-1)+0.48182*y(n-2)+0.00042547*u(n-2)-0.0030307*u(n-
1)+2.8506e-005*(y(n-1)^2)*u(n-1)-2.3374e-005*(u(n-1)^2)*u(n-3)+4.0015e-
005*(y(n-3)^3)+0.0030301*y(n-2)*u(n-2)-0.00031422*(y(n-3)^2)*u(n-3);

```

```
% Atualização de variáveis
y(n-1) = y(n);
y(n-2) = y(n-1);
y(n-3) = y(n-2);
u(n-1) = u(n);
u(n-2) = u(n-1);
u(n-3) = u(n-2);

% Armazenamento de valores
vy(n)= y(n);
vu(n)= u(n);
vn(n) = n;
rf(n)=r(n);
vP(n)= Kp*e(n-1);
vIn(n)= I(n-1);

end

% Gráficos resultantes da simulação
figure(1)
plot(vt,rf); grid;
xlabel('t'); ylabel('r');
figure(2)
plot(vt,vu); grid;
xlabel('t'); ylabel('u');
figure(3)
plot(vt,vy/sp); grid;
xlabel('t'); ylabel('y');

figure(4)
subplot(211);
plot(vt,vP,'k');grid;
xlabel('t[s]'); ylabel('e(t)');
subplot(212);
plot(vt,vIn,'k');grid;
xlabel('t[s]'); ylabel('Ie(t)');
```

## Apêndice G – Programa de Simulação da Malha de Controle com o PI *Fuzzy*

```

% Simulação da malha de controle com modelo NARMAX e PI-Fuzzy

clear all

sp=input('sp=');

T=0.0667; % Tempo de varredura

% Definição das funções de pertinência triangulares: cxyz=centro,
% bxyz=base, N=negative, Z=zero, P=positive, S=small, M=medium, L=large
ceN1=-5; ceZ1=0; ceP1=5;
beN1=5; beZ1=1.25; beP1=5;
cieN2=-5; cieZ2=0; cieP2=5;
bieN2=5; bieZ2=5; bieP2=5;
caNL=-10; caNM=-5; caZE=0; caPM=5; caPL=10;

% Condições iniciais
y(1)=sp-1;y(2)=sp-1;y(3)=sp-1;
u(1)=2.5;u(2)=2.5;u(3)=2.5;
I(1)=2.5;I(2)=2.5;I(3)=2.5;
np = 2699;
t=0;

% Dados do controlador
ge=1;%Kp em w da malha discreta
gv=0.1;%Ki em w da malha discreta

% Simulação da malha de controle com o modelo NARMAX
for n = 4:np

    vt(n)=t;
    t=t+T;

    r(n) = sp;
    e(n-1) = sp - y(n-1);    % Erro da malha

    I(n) = I(n-1) + gv*e(n-1)*T; % Ação integral e anti-windUp
    if (I(n) > 5)
        I(n) = 5;
    end
    if (I(n)< 0)
        I(n)= 0;
    end
end

```

```

    e(n) = e(n-1);

% Equação do controlador
    ie = I(n-1);
    e = ge*e(n-1);

% Variável de entrada "e" e = x1n
% Variável de entrada "ie" ie = x2n

% Fuzificação de dados:
if (((ceN1 - beN1) <= e) & (e <= (ceN1 + beN1))),
    ueN1 = 1 - (abs(ceN1 - e)) / beN1;
else
    ueN1 = 0;
end
if e < ceN1,
    ueN1 = 1;
end

if (((ceZ1 - beZ1) <= e) & (e <= (ceZ1 + beZ1))),
    ueZ1 = 1 - (abs(ceZ1 - e)) / beZ1;
else
    ueZ1 = 0;
end

if (((ceP1 - beP1) <= e) & (e <= (ceP1 + beP1))),
    ueP1 = 1 - (abs(ceP1 - e)) / beP1;
else
    ueP1 = 0;
end
if e > ceP1,
    ueP1 = 1;
end

if (((cieN2 - bieN2) <= ie) & (ie <= (cieN2 + bieN2))),
    uieN2 = 1 - (abs(cieN2 - ie)) / bieN2;
else
    uieN2 = 0;
end
if ie < cieN2,
    uieN2 = 1;
end

if (((cieZ2 - bieZ2) <= ie) & (ie <= (cieZ2 + bieZ2))),
    uieZ2 = 1 - (abs(cieZ2 - ie)) / bieZ2;
else
    uieZ2 = 0;
end

if (((cieP2 - bieP2) <= ie) & (ie <= (cieP2 + bieP2))),
    uieP2 = 1 - (abs(cieP2 - ie)) / bieP2;
else
    uieP2 = 0;
end

```

```

end
if ie > cieP2,
    uieP2 = 1;
end

% Implicações lógicas: operador min ou produto
ur1 = ueN1 * uieN2;
ur2 = ueZ1 * uieN2;
ur3 = ueP1 * uieN2;
ur4 = ueN1 * uieZ2;
ur5 = ueZ1 * uieZ2;
ur6 = ueP1 * uieZ2;
ur7 = ueN1 * uieP2;
ur8 = ueZ1 * uieP2;
ur9 = ueP1 * uieP2;

% Defuzificação (método da altura):
Nd =
caNL*ur1+caNM*ur2+caZE*ur3+caNM*ur4+caZE*ur5+caPM*ur6+caZE*ur7+caPM*ur8+caP
L*ur9;

Dd = ur1+ur2+ur3+ur4+ur5+ur6+ur7+ur8+ur9;

d = Nd / Dd;

Defuz = d;

u(n) = Defuz; % Armazenamento de valores

% Saturação
if (u(n) < 0)
    u(n) = 0;
end
if (u(n) > 5)
    u(n) = 5;
end

% Modelo NARMAX
y(n) = 0.51469*y(n-1)+0.48182*y(n-2)+0.00042547*u(n-2)-0.0030307*u(n-
1)+2.8506e-005*(y(n-1)^2)*u(n-1)-2.3374e-005*(u(n-1)^2)*u(n-3)+4.0015e-
005*(y(n-3)^3)+0.0030301*y(n-2)*u(n-2)-0.00031422*(y(n-3)^2)*u(n-3);

% Atualização de variáveis
y(n-1) = y(n);
y(n-2) = y(n-1);
y(n-3) = y(n-2);
u(n-1) = u(n);
u(n-2) = u(n-1);
u(n-3) = u(n-2);

```

```
% Armazenamento de valores
vy(n)= y(n);
vu(n)= u(n);
rf(n)=r(n);

end

% Gráficos resultantes da simulação
figure(2)
plot(vt,rf); grid;
xlabel('t'); ylabel('r');
figure(3)
plot(vt,vu); grid;
xlabel('t'); ylabel('u');
figure(4)
plot(vt,vy/sp); grid;
xlabel('t'); ylabel('y');
```

## Apêndice H – Programa de Avaliação do Tempo de Processamento das Regras *Fuzzy versus Rough*

```

% Programa de avaliação do tempo de processamento das regras do fuzzy
versus rough

clear all;      % Zerar variáveis

% Definição das funções de pertinência triangulares: cxyz=centro,
% bxyz=base, N=negative, Z=zero, P=positive, S=small, M=medium, L=large
ceN1=-5; ceZ1=0; ceP1=5;
beN1=5; beZ1=1.25; beP1=5;
cieN2=-5; cieZ2=0; cieP2=5;
bieN2=5; bieZ2=5; bieP2=5;
caNL=-10; caNM=-5; caZE=0; caPM=5; caPL=10;

% Valores extremos dos conjuntos crisp do modelo rough:
A1k=-5;      A1m=-1.25;   B1k=-5;      B1m=-2.5;
A2k=-1.25;   A2m=1.25;    B2k=-2.5;   B2m=2.5;
A3k= 1.25;   A3m=5;        B3k=2.5;    B3m=5;

% Coeficientes do modelo rough:
c01=-4.3374;   c11=0.3253;   c21=0.488;
c02=5.0875;   c12=2.07;    c22=2.07;
c03=2.0875;   c13=0.3333;   c23=0.5;
c04=-2.80175; c14=0.4733;   c24=0.355;
c05=0;        c15=2.384;    c25=1.192;
c06=2.80175;   c16=0.4733;   c26=0.355;
c07=-2.0875;   c17=0.3333;   c27=0.5;
c08=-5.0875;   c18=2.07;    c28=2.07;
c09=4.3374;   c19=0.3253;   c29=0.488;

Np = 100; % Número de amostras da simulação

for n=1:Np % Geração de dados para teste

    dado = rand;
    if dado < 0.5
        sg = -1;
    else
        sg = 1;
    end
    x1 = 2*sg*rand;
    vx1(n) = x1;

    dado = rand;
    if dado < 0.5
        sg = -1;
    else
        sg = 1;
    end

```

```

    end
    x2 = 2*sg*rand;
    vx2(n) = x2;

end

% Simulação das regras fuzzy:
tic
for n=1:Np

    x1 = vx1(n);
    x2 = vx2(n);

    e = x1; % Variável de entrada "e"
    ie = x2; % Variável de entrada "ie"

    % Fuzificação de dados:
    if (((ceN1 - beN1) <= e) & (e <= (ceN1 + beN1))),
        ueN1 = 1 - (abs(ceN1 - e)) / beN1;
    else
        ueN1 = 0;
    end
    if e < ceN1,
        ueN1 = 1;
    end

    if (((ceZ1 - beZ1) <= e) & (e <= (ceZ1 + beZ1))),
        ueZ1 = 1 - (abs(ceZ1 - e)) / beZ1;
    else
        ueZ1 = 0;
    end

    if (((ceP1 - beP1) <= e) & (e <= (ceP1 + beP1))),
        ueP1 = 1 - (abs(ceP1 - e)) / beP1;
    else
        ueP1 = 0;
    end
    if e > ceP1,
        ueP1 = 1;
    end

    if (((cieN2 - bieN2) <= ie) & (ie <= (cieN2 + bieN2))),
        uieN2 = 1 - (abs(cieN2 - ie)) / bieN2;
    else
        uieN2 = 0;
    end
    if ie < cieN2,
        uieN2 = 1;
    end

    if (((cieZ2 - bieZ2) <= ie) & (ie <= (cieZ2 + bieZ2))),
        uieZ2 = 1 - (abs(cieZ2 - ie)) / bieZ2;
    else

```

```

    uieZ2 = 0;
end

if (((cieP2 - bieP2) <= ie) & (ie <= (cieP2 + bieP2))),
    uieP2 = 1 - (abs(cieP2 - ie)) / bieP2;
else
    uieP2 = 0;
end
if ie > cieP2,
    uieP2 = 1;
end

% Implicações lógicas: operador min ou produto
ur1 = ueN1 * uieN2;
ur2 = ueZ1 * uieN2;
ur3 = ueP1 * uieN2;
ur4 = ueN1 * uieZ2;
ur5 = ueZ1 * uieZ2;
ur6 = ueP1 * uieZ2;
ur7 = ueN1 * uieP2;
ur8 = ueZ1 * uieP2;
ur9 = ueP1 * uieP2;

% Defuzificação (método da altura):
Nd =
caNL*ur1+caNM*ur2+caZE*ur3+caNM*ur4+caZE*ur5+caPM*ur6+caZE*ur7+caPM*ur8+caP
L*ur9;

Dd = ur1+ur2+ur3+ur4+ur5+ur6+ur7+ur8+ur9;

d = Nd / Dd;

Defuz = d;

    vyf(n) = Defuz; % Armazenamento de valores
    vn(n) = n-1;

end
toc

% Gráfico da simulação do controlador fuzzy
subplot(211);
plot(vn,vx1,'b.',vn,vx2,'r. '); grid; ylabel('erro e integral do erro');
xlabel('Amostras[k]');
title('Gráfico da simulação do controlador Fuzzy');
subplot(212);
plot(vn,vyf,'b '); grid; ylabel('u(k)'); xlabel('Amostras[k]');
figure

```

```

% Simulação das regras rough:
tic
for n=1:Np

    x1 = vx1(n);
    x2 = vx2(n);

    % Processamento das regras do modelo rough
    ya=0;

    if (((x1 >= A1k) & (x1 < A1m)) & ((x2 >= B1k) & (x2 < B1m)))
        ya = c01+c11*x1+c21*x2;
    end
    if (((x1 >= A2k) & (x1 < A2m)) & ((x2 >= B1k) & (x2 < B1m)))
        ya = c02+c12*x1+c22*x2;
    end
    if (((x1 >= A3k) & (x1 <= A3m)) & ((x2 >= B1k) & (x2 < B1m)))
        ya = c03+c13*x1+c23*x2;
    end

    if (((x1 >= A1k) & (x1 < A1m)) & ((x2 >= B2k) & (x2 < B2m)))
        ya = c04+c14*x1+c24*x2;
    end
    if (((x1 >= A2k) & (x1 < A2m)) & ((x2 >= B2k) & (x2 < B2m)))
        ya = c05+c15*x1+c25*x2;
    end
    if (((x1 >= A3k) & (x1 <= A3m)) & ((x2 >= B2k) & (x2 < B2m)))
        ya = c06+c16*x1+c26*x2;
    end

    if (((x1 >= A1k) & (x1 < A1m)) & ((x2 >= B3k) & (x2 <= B3m)))
        ya = c07+c17*x1+c27*x2;
    end
    if (((x1 >= A2k) & (x1 < A2m)) & ((x2 >= B3k) & (x2 <= B3m)))
        ya = c08+c18*x1+c28*x2;
    end
    if (((x1 >= A3k) & (x1 <= A3m)) & ((x2 >= B3k) & (x2 <= B3m)))
        ya = c09+c19*x1+c29*x2;
    end

    vyr = ya;

    vya(n) = vyr; % Armazenamento de valores
    vn(n) = n-1;

end
toc

% Gráfico da simulação do controlador rough
subplot(211);
plot(vn,vx1,'b.',vn,vx2,'r.');
```

grid; ylabel('erro e integral do erro');

xlabel('Amostras[k]');

title('Gráfico da simulação do controlador Rough');

```
subplot(212);
plot(vn,vya,'b'); grid; ylabel('u(k)'); xlabel('Amostras[k]');
figure

% Gráfico de Comparação
subplot(211);
plot(vn,vx1,'b.',vn,vx2,'r. '); grid; ylabel('Entradas(k)');
xlabel('Amostras[k]'); legend('Erro','Integral do Erro');
subplot(212);
plot(vn,vyf,'b',vn,vya,'r'); grid; ylabel('u(k)'); xlabel('Amostras[k]');
legend('Fuzzy','Rough');
```

## Apêndice I – Programa de Simulação da Malha de Controle com o PI *Rough*

```

% Programa de simulação da malha de controle com modelo NARMAX e PI-
aproximado

clear all

sp=input('sp=');

T=0.0679; % Tempo de varredura

% Condições iniciais
y(1)=sp-1;y(2)=sp-1;y(3)=sp-1;
u(1)=2.5;u(2)=2.5;u(3)=2.5;
I(1)=2.5;I(2)=2.5;I(3)=2.5;
np = 2651;
t=0;

% Dados do controlador
ge=1;%Kp em w da malha discreta
gv=0.1;%Ki em w da malha discreta

% Valores extremos dos conjuntos crisp do modelo rough:
A1k=-5;      A1m=-1.25;   B1k=-5;      B1m=-2.5;
A2k=-1.25;   A2m=1.25;    B2k=-2.5;   B2m=2.5;
A3k= 1.25;   A3m=5;       B3k=2.5;   B3m=5;

% Coeficientes do modelo rough:
c01=-4.3374;  c11=0.3253;   c21=0.488;
c02=5.0875;  c12=2.07;     c22=2.07;
c03=2.0875;  c13=0.3333;   c23=0.5;
c04=-2.80175; c14=0.4733;  c24=0.355;
c05=0;       c15=2.384;    c25=1.192;
c06=2.80175; c16=0.4733;  c26=0.355;
c07=-2.0875; c17=0.3333;  c27=0.5;
c08=-5.0875; c18=2.07;    c28=2.07;
c09=4.3374;  c19=0.3253;  c29=0.488;

% Simulação da malha de controle com o modelo NARMAX
for n = 4:np

    vt(n)=t;
    t=t+T;

    r(n) = sp;
    e(n-1) = sp - y(n-1); % Erro da malha

```

```

I(n) = I(n-1) + gv*e(n-1)*T; % Ação integral e anti-windUp
if (I(n) > 5)
    I(n) = 5;
end
if (I(n) < 0)
    I(n) = 0;
end

e(n) = e(n-1);

% Equação do controlador
x2n = I(n-1);
x1n = ge*e(n-1);

% Regras de decisão

% Processamento das regras do modelo rough
u(n)=0;

if (((x1n >= A1k) & (x1n < A1m)) & ((x2n >= B1k) & (x2n < B1m)))
    u(n) = c01+c11*x1n+c21*x2n;
end
if (((x1n >= A2k) & (x1n < A2m)) & ((x2n >= B1k) & (x2n < B1m)))
    u(n) = c02+c12*x1n+c22*x2n;
end
if (((x1n >= A3k) & (x1n <= A3m)) & ((x2n >= B1k) & (x2n < B1m)))
    u(n) = c03+c13*x1n+c23*x2n;
end

if (((x1n >= A1k) & (x1n < A1m)) & ((x2n >= B2k) & (x2n < B2m)))
    u(n) = c04+c14*x1n+c24*x2n;
end
if (((x1n >= A2k) & (x1n < A2m)) & ((x2n >= B2k) & (x2n < B2m)))
    u(n) = c05+c15*x1n+c25*x2n;
end
if (((x1n >= A3k) & (x1n <= A3m)) & ((x2n >= B2k) & (x2n < B2m)))
    u(n) = c06+c16*x1n+c26*x2n;
end

if (((x1n >= A1k) & (x1n < A1m)) & ((x2n >= B3k) & (x2n <= B3m)))
    u(n) = c07+c17*x1n+c27*x2n;
end
if (((x1n >= A2k) & (x1n < A2m)) & ((x2n >= B3k) & (x2n <= B3m)))
    u(n) = c08+c18*x1n+c28*x2n;
end
if (((x1n >= A3k) & (x1n <= A3m)) & ((x2n >= B3k) & (x2n <= B3m)))
    u(n) = c09+c19*x1n+c29*x2n;
end

% Saturação
if (u(n) < 0)
    u(n) = 0;

```

```

end
if (u(n) > 5)
    u(n) = 5;
end

% Modelo NARMAX e limitações
y(n) = 0.51469*y(n-1)+0.48182*y(n-2)+0.00042547*u(n-2)-0.0030307*u(n-
1)+2.8506e-005*(y(n-1)^2)*u(n-1)-2.3374e-005*(u(n-1)^2)*u(n-3)+4.0015e-
005*(y(n-3)^3)+0.0030301*y(n-2)*u(n-2)-0.00031422*(y(n-3)^2)*u(n-3);

% Atualização de variáveis
y(n-1) = y(n);
y(n-2) = y(n-1);
y(n-3) = y(n-2);
u(n-1) = u(n);
u(n-2) = u(n-1);
u(n-3) = u(n-2);

% Armazenamento de valores
vy(n)= y(n);
vu(n)= u(n);
rf(n)=r(n);

end

% Gráficos resultantes da simulação
figure(2)
plot(vt,rf); grid;
xlabel('t'); ylabel('r');
figure(3)
plot(vt,vu); grid;
xlabel('t'); ylabel('u');
figure(4)
plot(vt,vy/sp); grid;
xlabel('t'); ylabel('y');

```

## Apêndice J – Programa do Experimento Prático da Malha de Controle com o PI *Fuzzy*

```

% Algoritmo de controle PI_FUZZY NIDAQ

clear all;

% Configurações do sistema de aquisição de dados
ai = analoginput('nidaq','Dev1');
addchannel(ai, 0);
ai.SampleRate = 300;
ai.SamplesPerTrigger = 2;
set(ai, 'InputType', 'SingleEnded');
ao = analogoutput('nidaq','Dev1');
addchannel(ao,0);

% Condições iniciais
In=2.5;

% Tempo de varredura e número de amostras
T=0.0667;
Na = 2699;

% Definição das funções de pertinência triangulares: cxyz=centro,
% bxyz=base, N=negative, Z=zero, P=positive, S=small, M=medium, L=large
ceN1=-5; ceZ1=0; ceP1=5;
beN1=5; beZ1=1.25; beP1=5;
cieN2=-5; cieZ2=0; cieP2=5;
bieN2=5; bieZ2=5; bieP2=5;
caNL=-10; caNM=-5; caZE=0; caPM=5; caPL=10;

% Laço do teste
n=1; % Indexador

% Dados do controlador
ge=1;%Kp em w da malha discreta
gv=0.1;%Ki em w da malha discreta

sp=input('sp=');

% Conversão SP(n_tanque) para v_gráfico(v)
spv = -1.70581 + 1.356846*sp;

while n <= Na,
    tic

    % Comandos para leitura de dado do A/D
    start(ai)
    wait(ai,0.04);
    PV = getdata(ai);

```

```

pv=Pv(1);

% Conversão v_gráfico(v) para SP(n_tanque)
pvh = 1.259459 + 0.736486*pv;

vd(n) = pvh; % Registro dos dados coletados

en = spv - pv; % Erro da malha

In = In + gv*en*T; % Ação integral e anti-windUp
if (In > 5)
    In = 5;
end
if (In < 0)
    In = 0;
end

% Equação do controlador
ie = In;
e = ge*en;

% Variável de entrada "e" e = x1n
% Variável de entrada "ie" ie = x2n

% Fuzificação de dados:
if (((ceN1 - beN1) <= e) & (e <= (ceN1 + beN1))),
    ueN1 = 1 - (abs(ceN1 - e)) / beN1;
else
    ueN1 = 0;
end
if e < ceN1,
    ueN1 = 1;
end

if (((ceZ1 - beZ1) <= e) & (e <= (ceZ1 + beZ1))),
    ueZ1 = 1 - (abs(ceZ1 - e)) / beZ1;
else
    ueZ1 = 0;
end

if (((ceP1 - beP1) <= e) & (e <= (ceP1 + beP1))),
    ueP1 = 1 - (abs(ceP1 - e)) / beP1;
else
    ueP1 = 0;
end
if e > ceP1,
    ueP1 = 1;
end

if (((cieN2 - bieN2) <= ie) & (ie <= (cieN2 + bieN2))),
    uieN2 = 1 - (abs(cieN2 - ie)) / bieN2;
else
    uieN2 = 0;
end

```

```

end
if ie < cieN2,
    uieN2 = 1;
end

if (((cieZ2 - bieZ2) <= ie) & (ie <= (cieZ2 + bieZ2))),
    uieZ2 = 1 - (abs(cieZ2 - ie)) / bieZ2;
else
    uieZ2 = 0;
end

if (((cieP2 - bieP2) <= ie) & (ie <= (cieP2 + bieP2))),
    uieP2 = 1 - (abs(cieP2 - ie)) / bieP2;
else
    uieP2 = 0;
end
if ie > cieP2,
    uieP2 = 1;
end

% Implicações lógicas: operador min ou produto.
ur1 = ueN1 * uieN2;
ur2 = ueZ1 * uieN2;
ur3 = ueP1 * uieN2;
ur4 = ueN1 * uieZ2;
ur5 = ueZ1 * uieZ2;
ur6 = ueP1 * uieZ2;
ur7 = ueN1 * uieP2;
ur8 = ueZ1 * uieP2;
ur9 = ueP1 * uieP2;

% Defuzificação (método da altura):
Nd =
caNL*ur1+caNM*ur2+caZE*ur3+caNM*ur4+caZE*ur5+caPM*ur6+caZE*ur7+caPM*ur8+caP
L*ur9;

Dd = ur1+ur2+ur3+ur4+ur5+ur6+ur7+ur8+ur9;

d = Nd / Dd;

Defuz = d;

un = Defuz; % Armazenamento de valores

% Saturação    if (un < 0)
    un = 0;
end
if (un > 5)
    un = 5;
end

putsample(ao,un); % Escrita de dado no D/A

```

```

vn(n) = n-1;
vu(n) = un;

n = n + 1; % Incrementa indexador dos dados
T = toc;
vtoc(n)=T;

%Armazena variáveis
vP(n)= e;
vIn(n)= ie;
end

vt=vtoc(1:Na);
vt(1)=vtoc(2);

t=0;
for k=1:Na
    vT(k) = t;
    dt = vt(k);
    t = t + dt;
end

vP=vP(1:Na);
vIn=vIn(1:Na);

% Gráficos
figure(1)
plot(vT,vd, 'b');grid;
xlabel('t'); ylabel('y');
figure(2)
plot(vT,vu, 'b');grid;
xlabel('t'); ylabel('u');

figure(3)
plot(vT,vP, 'b');grid;
xlabel('t'); ylabel('P');
figure(4)
plot(vT,vIn, 'b');grid;
xlabel('t'); ylabel('I');

```

## Apêndice L – Programa do Experimento Prático da Malha de Controle com o PI *Rough*

```

% Algoritmo do PI-aproximado NIDAQ

clear all;

% Configurações do sistema de aquisição de dados
ai = analoginput('nidaq','Dev1');
addchannel(ai, 0);
ai.SampleRate = 300;
ai.SamplesPerTrigger = 2;
set(ai, 'InputType', 'SingleEnded');
ao = analogoutput('nidaq','Dev1');
addchannel(ao,0);

% Condições iniciais
In=2.5;

% Tempo de varredura e número de amostras
T=0.0679;
Na = 2651;

% Valores extremos dos conjuntos crisp do modelo rough:
A1k=-5;      A1m=-1.25;   B1k=-5;      B1m=-2.5;
A2k=-1.25;   A2m=1.25;    B2k=-2.5;   B2m=2.5;
A3k= 1.25;   A3m=5;       B3k=2.5;    B3m=5;

% Coeficientes do modelo rough:
c01=-4.3374;   c11=0.3253;   c21=0.488;
c02=5.0875;   c12=2.07;    c22=2.07;
c03=2.0875;   c13=0.3333;   c23=0.5;
c04=-2.80175; c14=0.4733;   c24=0.355;
c05=0;        c15=2.384;    c25=1.192;
c06=2.80175;   c16=0.4733;   c26=0.355;
c07=-2.0875;   c17=0.3333;   c27=0.5;
c08=-5.0875;   c18=2.07;    c28=2.07;
c09=4.3374;   c19=0.3253;   c29=0.488;

% Laço do teste
n=1; % Indexador

% Dados do controlador
ge=1; %Kp em w da malha discreta
gv=0.1; %Ki em w da malha discreta

sp=input('sp=');

% Conversão SP(n_tanque) para v_gráfico(v)
spv = -1.70581 + 1.356846*sp;

```

```

while n <= Na,
    tic

    % Algoritmo comandos para leitura de dados do A/D
    start(ai)
    wait(ai,0.04);
    PV = getdata(ai);
    pv=PV(1);

    % Conversão v_gráfico(v) para SP(n_tanque)
    pvh = 1.259459 + 0.736486*pv;

    vd(n) = pvh; % Registro dos dados coletados

    en = spv - pv; % Erro da malha

    In = In + gv*en*T; % Ação integral e anti-windUp
    if (In > 5)
        In = 5;
    end
    if (In < 0)
        In = 0;
    end

    % Equação do controlador
    x2n = In;
    x1n = ge*en;

    % Regras de decisão

    % Processamento das regras do modelo rough
    un=0;

    if (((x1n >= A1k) & (x1n < A1m)) & ((x2n >= B1k) & (x2n < B1m)))
        un = c01+c11*x1n+c21*x2n;
    end
    if (((x1n >= A2k) & (x1n < A2m)) & ((x2n >= B1k) & (x2n < B1m)))
        un = c02+c12*x1n+c22*x2n;
    end
    if (((x1n >= A3k) & (x1n <= A3m)) & ((x2n >= B1k) & (x2n < B1m)))
        un = c03+c13*x1n+c23*x2n;
    end

    if (((x1n >= A1k) & (x1n < A1m)) & ((x2n >= B2k) & (x2n < B2m)))
        un = c04+c14*x1n+c24*x2n;
    end
    if (((x1n >= A2k) & (x1n < A2m)) & ((x2n >= B2k) & (x2n < B2m)))
        un = c05+c15*x1n+c25*x2n;
    end
    if (((x1n >= A3k) & (x1n <= A3m)) & ((x2n >= B2k) & (x2n < B2m)))
        un = c06+c16*x1n+c26*x2n;
    end
end

```

```

if (((x1n >= A1k) & (x1n < A1m)) & ((x2n >= B3k) & (x2n <= B3m)))
    un = c07+c17*x1n+c27*x2n;
end
if (((x1n >= A2k) & (x1n < A2m)) & ((x2n >= B3k) & (x2n <= B3m)))
    un = c08+c18*x1n+c28*x2n;
end
if (((x1n >= A3k) & (x1n <= A3m)) & ((x2n >= B3k) & (x2n <= B3m)))
    un = c09+c19*x1n+c29*x2n;
end

% Saturação
if (un < 0)
    un = 0;
end
if (un > 5)
    un = 5;
end

putsample(ao,un); % Comando para escrita de dado no D/A

vn(n) = n-1;
vu(n) = un;

n = n + 1; % Incrementa indexador dos dados
T = toc;
vtoc(n)=T;

    %Armazena variáveis
vP(n)= x1n;
vIn(n)= x2n;
end

vt=vtoc(1:Na);
vt(1)=vtoc(2);

t=0;
for k=1:Na
    vT(k) = t;
    dt = vt(k);
    t = t + dt;
end

vP=vP(1:Na);
vIn=vIn(1:Na);

% Gráficos
figure(1)
plot(vT,vd, 'b');grid;
xlabel('t'); ylabel('y');
figure(2)
plot(vT,vu, 'b');grid;
xlabel('t'); ylabel('u');

```

```
figure(3)
plot(vT,vP,'b');grid;
xlabel('t'); ylabel('P');
figure(4)
plot(vT,vIn,'b');grid;
xlabel('t'); ylabel('I');
```

## Referências

- AGUIRRE, Luis Antônio. **Introdução à identificação de sistemas, técnicas lineares e não-lineares aplicadas a sistemas reais**. 2. ed. Belo Horizonte: UFMG, 2004.
- ASTROM, K. J., WITTENMARK, B. **Computer-Controlled Systems**. Prentice-Hall, 1990.
- BRAAE, M., RUTHERFORD, D. A. **Theoretical and linguistic aspects of the fuzzy logic controllers**. *Automatica*, v.15, p.553-577, 1979.
- CAMATTA, Ulisses Pessin. **Projeto e implementação de um regulador de velocidade e corrente baseado na técnica de conjuntos aproximados (rough sets)**. 2009. 86p. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Itajubá, Itajubá, 2009.
- DORF, Richard C.; BISHOP, Robert H. **Sistemas de controle modernos**. 8. ed. Rio de Janeiro: LTC, 2001.
- DRIANKOV, D., HELLENDORRN, H., REINFRANK, M. **An introduction to fuzzy control**. Springer-Verlag, Berlin. 1996.
- GUARACY, Fernando Henrique Duarte. **Contribuições a aplicações práticas de sistemas de controle por modos deslizantes**. 2013. 144p. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Itajubá, Itajubá, 2013.
- GUARACY, Fernando; PINHEIRO, Carlos; FERREIRA, Luís. **Converting Fuzzy Models to Rough Models**. IEEE Xplore Biblioteca Digital, 2014. Disponível em: < <http://ieeexplore.ieee.org/Xplore/home.jsp> >. Acesso em: 06 abril. 2015.
- GUPTA, M. M., SINHA, N. K. **“Intelligent Control Systems: Theory and Applications”**, IEEE Press, 1996.
- LIMA, Isaías; PINHEIRO, Carlos A. M; SANTOS, Flávia A. Oliveira. **Inteligência Artificial**. Rio de Janeiro: Elsevier, 2014.
- LIN, Hai; ANTSAKLIS, Panos J. **Stability and Stabilizability of Switched Linear Systems: A Survey of Recent Results**. *IEEE Transactions on Automatic Control*, vol. 54, no. 2, February, 2009.
- MACHADO, Kascilene; PINHEIRO, Carlos. Rough controllers with state feedback. **Engineering Applications of Artificial Intelligence, Elsevier, Science Direct**, 30 ago. 2013, p. 2467-2449. Disponível em: < [www.elsevier.com/locate/engappai](http://www.elsevier.com/locate/engappai) >. Acesso em: 04 março. 2015.

MAMDANI, E. H. **Applications of fuzzy algorithms for control of simple dynamic plant.** Proc IEE, v 121(12), 1.585-1.588, 1974.

NELLES, Oliver. **Nonlinear system identification: from classical approaches to neural networks and fuzzy models.** New York: Springer-Verlag Berlin Heidelberg, 2001.

OGATA, Katsuhiko. **Engenharia de controle moderno.** 5. ed. São Paulo: Pearson Prentice Hall, 2010.

PAWLAK, Z. **Rough Sets.** International Journal of Computer and Information Sciences, v. 11, n. 5, p.341-56, 1982.

PHILLIPS, Charles L.; NAGLE, H. Troy. **Digital control system analysis and design.** 3. ed. New Jersey: Pearson Prentice Hall, 1995.

PINHEIRO, Carlos Alberto Murari. **Análise e projeto de sistemas de controle fuzzy: uma abordagem no domínio da frequência.** 2000. 133p. Tese (Doutorado em Engenharia Elétrica) – Universidade Estadual de Campinas, Campinas, 2000.

PINHEIRO, Carlos; CAMATTA, Ulisses; REZEK, Angelo. **Rough Controller Synthesis.** Fuzzy logic controls, concepts, theories and applications, Edited by Elmer P. Dadios. Croatia: INTECH, 2012.

PINHEIRO, C. *et al.* **Granular Synthesis of Rule-Based Models and Function Approximation using Rough Sets.** In: Yao, J.T. (ed.), Novel Developments in Granular Computing, p. 408-425. Information Science Publishing, New York, 2010.

PINHEIRO, Carlos; GOMIDE, Fernando. Equivalência entre controladores fuzzy e compensadores clássicos. **Simpósio Brasileiro de Automação Inteligente**, São Paulo, 08-10 de setembro de 1999a.

PINHEIRO, C. A. M., GOMIDE, F. A. C. **Fuzzy control systems by learning method and frequency response.** FUZZ-IEEE Conference, v.1, p. 444-448, 1999b.

PINHEIRO, Carlos Alberto Murari; SOUZA, Antonio Carlos de; ROSA, Paulo César; LIMA, Isaías. **Sistemas de controle: projetos, simulações e experiências de laboratório.** Itajubá: ESETec, 2012.

SILVA, José Roberto Pinto da. **Proposta de compensador baseado em regras para pss aplicação da técnica IMC em AVR.** 2014. 145p. Tese (Doutorado em Engenharia Elétrica) – Universidade Federal de Itajubá, Itajubá, 2014.

SILVA, J. R. P.; PINHEIRO, C. A. M.; MENDES, P. P. C. **Proposta de um novo tipo de controlador baseado em regras para aplicações em estabilizadores de sistemas de potência.** XXI Congresso Internacional de Ingeniería Electrónica, Eléctrica y Computación

INTERCON, 2014.

SLOTINE, J. J. E.; LI, W. **Applied nonlinear control**. Prentice Hall, 1991.

SOUZA, Antonio Carlos de; PINHEIRO, Carlos Alberto Murari. **Introdução à modelagem, análise e simulação de sistemas dinâmicos**. Rio de Janeiro: Interciência, 2008.

ZADEH, L. A. **Fuzzy sets**. Information and Control. n. 8, p.338-53, 1965.

## Anexo A – Datasheet NI USB-6008

### USER GUIDE AND SPECIFICATIONS

## NI USB-6008/6009

### Bus-Powered Multifunction DAQ USB Device

[Français](#)   [Deutsch](#)   [日本語](#)   [한국어](#)   [简体中文](#)  
[ni.com/manuals](http://ni.com/manuals)

This user guide describes how to use the National Instruments USB-6008 and National Instruments USB-6009 data acquisition (DAQ) devices and lists specifications.

The NI USB-6008/6009 provides connection to eight single-ended analog input (AI) channels, two analog output (AO) channels, 12 digital input/output (DIO) channels, and a 32-bit counter with a full-speed USB interface. Table 1 compares the devices.

Table 1. NI USB-6008 and NI USB-6009 Comparison

Feature	NI USB-6008	NI USB-6009
AI resolution	12 bits differential, 11 bits single-ended	14 bits differential, 13 bits single-ended
Maximum AI sample rate, single channel*	10 kS/s	48 kS/s
Maximum AI sample rate, multiple channels (aggregate)*	10 kS/s	48 kS/s
DIO configuration	Open collector <sup>†</sup>	Each channel individually programmable as open collector or active drive <sup>†</sup>
* System-dependent. <sup>†</sup> This document uses NI-DAQmx naming conventions. Open-drain is called open collector and push-pull is called active drive.		

## Cables and Accessories

Table 4 contains information about cables and accessories available for the NI USB-6008/6009. For a complete list of accessories and ordering information, refer to the pricing section of the NI USB-6008 or NI USB-6009 product page at [ni.com](http://ni.com).

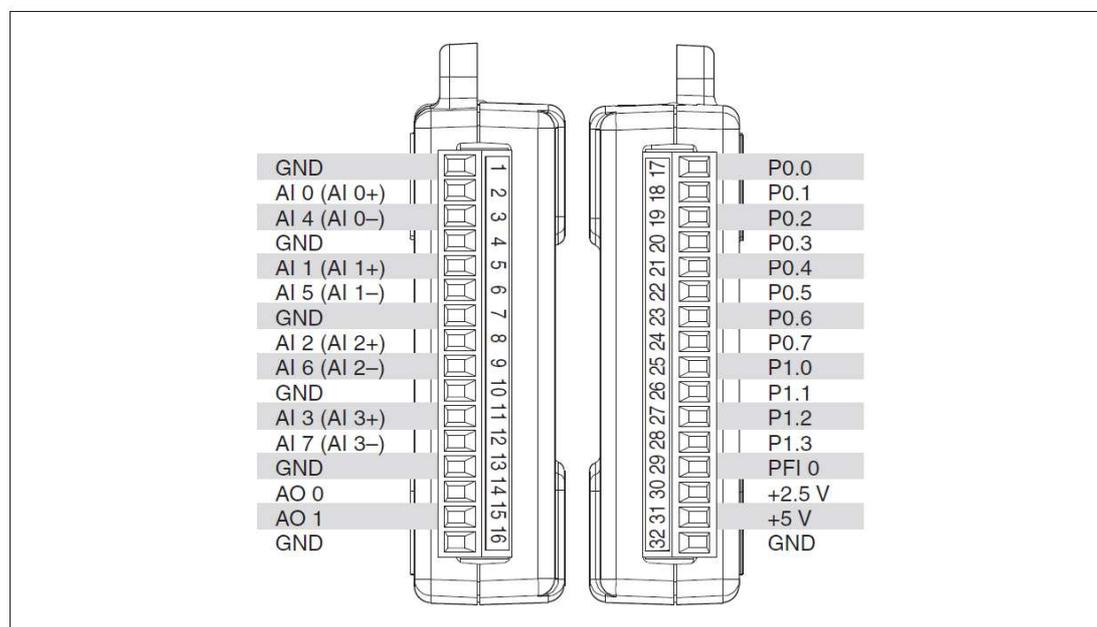
**Table 4.** NI USB-6008/6009 Cables and Accessories

Accessory	Part Number	Description
USB-6008/6009 Accessory Kit	779371-01	Four additional screw-terminal connectors, connector labels, and a screwdriver
USB-6000 Series Prototyping Accessory	779511-01	Unshielded breadboarding accessory for custom-defined signal conditioning and prototyping. You can use up to two accessories per device.
Hi-Speed USB Cable	184125-01 184125-02	1 m and 2 m lengths

**Caution:** For compliance with Electromagnetic Compatibility (EMC) requirements, this product must be operated with shielded cables and accessories. If unshielded cables or accessories are used, the EMC specifications are no longer guaranteed unless all unshielded cables and/or accessories are installed in a shielded enclosure with properly designed and shielded input/output ports.

## Pinout and Signal Descriptions

Figure 6 shows the pinout of the NI USB-6008/6009. Analog input signal names are listed as single-ended analog input name, AI  $x$ , and then differential analog input name, (AI  $x+/-$ ). Refer to Table 5 for a detailed description of each signal.



**Figure 6.** NI USB-6008/6009 Pinout

**Table 5.** Signal Descriptions

Signal Name	Reference	Direction	Description
GND	—	—	<b>Ground</b> —The reference point for the single-ended analog input measurements, analog output voltages, digital signals, +5 VDC supply, and +2.5 VDC at the I/O connector, and the bias current return point for differential mode measurements.
AI <0..7>	Varies	Input	<b>Analog Input Channels 0 to 7</b> —For single-ended measurements, each signal is an analog input voltage channel. For differential measurements, AI 0 and AI 4 are the positive and negative inputs of differential analog input channel 0. The following signal pairs also form differential input channels: AI<1, 5>, AI<2, 6>, and AI<3, 7>. Refer to the <i>Analog Input</i> section for more information.
AO <0, 1>	GND	Output	<b>Analog Output Channels 0 and 1</b> —Supplies the voltage output of AO channel 0 or AO channel 1. Refer to the <i>Analog Output</i> section for more information.
P0.<0..7>	GND	Input or Output	<b>Port 0 Digital I/O Channels 0 to 7</b> —You can individually configure each signal as an input or output. Refer to the <i>Digital I/O</i> section for more information.
P1.<0..3>	GND	Input or Output	<b>Port 1 Digital I/O Channels 0 to 3</b> —You can individually configure each signal as an input or output. Refer to the <i>Digital I/O</i> section for more information.
PFI 0	GND	Input	<b>PFI 0</b> —This pin is configurable as either a digital trigger or an event counter input. Refer to the <i>PFI 0</i> section for more information.
+2.5 V	GND	Output	<b>+2.5 V External Reference</b> —Provides a reference for wrap-back testing. Refer to the <i>+2.5 V External Reference</i> section for more information.
+5 V	GND	Output	<b>+5 V Power Source</b> —Provides +5 V power up to 200 mA. Refer to the <i>+5 V Power Source</i> section for more information.

## External Reference and Power Source

The NI USB-6008/6009 creates an external reference and supplies a power source. All voltages are relative to ground (GND).

### +2.5 V External Reference

The NI USB-6008/6009 creates a high-purity reference voltage supply for the ADC using a multi-state regulator, amplifier, and filter circuit. You can use the resulting +2.5 V reference voltage as a signal for self-test.

### +5 V Power Source

The NI USB-6008/6009 supplies a 5 V, 200 mA output. You can use this source to power external components.



**Note** When the device is in USB suspend, the output is disabled.

## Specifications

The following specifications are typical at 25 °C, unless otherwise noted.

### Analog Input

Analog inputs

Differential .....	4
Single-ended .....	8, software-selectable

Input resolution

NI USB-6008	
Differential .....	12 bits
Single-ended .....	11 bits
NI USB-6009	
Differential .....	14 bits
Single-ended .....	13 bits

Max sample rate (aggregate)<sup>1</sup>

NI USB-6008 .....	10 kS/s
NI USB-6009 .....	48 kS/s

Converter type .....

Successive approximation

AI FIFO .....

512 bytes

Timing resolution .....

41.67 ns (24 MHz timebase)

Timing accuracy .....

100 ppm of actual sample rate

Input range

Differential .....	$\pm 20\text{ V}^2$ , $\pm 10\text{ V}$ , $\pm 5\text{ V}$ , $\pm 4\text{ V}$ , $\pm 2.5\text{ V}$ , $\pm 2\text{ V}$ , $\pm 1.25\text{ V}$ , $\pm 1\text{ V}$
Single-ended .....	$\pm 10\text{ V}$

<sup>1</sup> System-dependent.

<sup>2</sup>  $\pm 20\text{ V}$  means that  $|AI+ - (AI-)| \leq 20\text{ V}$ . However, AI+ and AI- must both be within  $\pm 10\text{ V}$  of GND. Refer to the [Taking Differential Measurements](#) section for more information.

Working voltage .....	±10 V
Input impedance .....	144 kΩ
Overvoltage protection .....	±35 V
Trigger source .....	Software or external digital trigger
System noise <sup>1</sup>	
Differential	
±20 V range .....	5 mVrms
±1 V range .....	0.5 mVrms
Single-ended	
±10 V range .....	5 mVrms
Absolute accuracy at full scale, differential <sup>2</sup>	

Range (V)	Typical at 25 °C (mV)	Maximum over Temperature (mV)
±20	14.7	138
±10	7.73	84.8
±5	4.28	58.4
±4	3.59	53.1
±2.5	2.56	45.1
±2	2.21	42.5
±1.25	1.70	38.9
±1	1.53	37.5

Absolute accuracy at full scale, single-ended

Range (V)	Typical at 25 °C (mV)	Maximum over Temperature (mV)
±10	14.7	138

## Analog Output

Analog outputs .....	2
Output resolution .....	12 bits
Maximum update rate .....	150 Hz, software-timed
Output range .....	0 to +5 V
Output impedance .....	50 Ω
Output current drive .....	5 mA
Power-on state .....	0 V

<sup>1</sup> System noise measured at maximum sample rate.

<sup>2</sup> Input voltages may not exceed the working voltage range.

Slew rate.....	1 V/ $\mu$ s
Short circuit current .....	50 mA
Absolute accuracy (no load)	
Typical.....	7 mV
Maximum at full scale .....	36.4 mV

## Digital I/O

Digital I/O lines	
P0.<0..7>.....	8 lines
P1.<0..3>.....	4 lines
Direction control .....	Each channel individually programmable as input or output
Output driver type <sup>1</sup>	
NI USB-6008 .....	Open collector
NI USB-6009 .....	Each channel individually programmable as open collector or active drive
Compatibility .....	TTL, LVTTTL, CMOS
Absolute maximum voltage range .....	-0.5 to 5.8 V with respect to GND
Pull-up resistor .....	4.7 k $\Omega$ to 5 V
Power-on state.....	Input
Digital logic levels	

Level	Min	Max
Input low voltage	-0.3 V	0.8 V
Input high voltage	2.0 V	5.8 V
Input leakage current	—	50 $\mu$ A
Output low voltage (I = -8.5 mA)	—	0.8 V
Output high voltage		
Active drive, I = -8.5 mA	2.0 V	3.5 V
Open collector, I = -0.6 mA, nominal	2.0 V	5.0 V
Open collector, I = -8.5 mA, with external pull-up resistor	2.0 V	—

## External Voltage

+5 V output (200 mA maximum)	
Minimum .....	+4.85 V
Typical.....	+5 V
+2.5 V output (1 mA maximum) .....	+2.5 V
+2.5 V accuracy .....	0.25% maximum
Reference temperature drift .....	50 ppm/ $^{\circ}$ C maximum

<sup>1</sup> This document uses NI-DAQmx naming conventions. Open-drain is called open collector and push-pull is called active drive.