



TRABALHO DE GRADUAÇÃO

**IMPLEMENTAÇÃO DE CONTROLE FUZZY  
EM CLP INDUSTRIAL**

**Jhonantans Moraes Rocha**



**ENGENHARIA  
MECATRÔNICA**  
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA  
Faculdade de Tecnologia  
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

**IMPLEMENTAÇÃO DE CONTROLE FUZZY  
EM CLP INDUSTRIAL**

**Jhonantans Moraes Rocha**

*Trabalho de Graduação submetido como requisito parcial de obtenção do grau de Engenheiro de Controle e Automação.*

Banca Examinadora

Prof. Eduardo Stockler Tognetti, ENE/UnB  
*Orientador*

Prof. Eugênio Fortaleza, ENM/UnB  
*Examinador Externo*

Dr. Luis Felipe Cruz Figueiredo, ENE/UnB  
*Examinador Externo*

**Brasília, 08 de dezembro de 2016**

## **FICHA CATALOGRÁFICA**

**ROCHA, JHONANTANS MORAES**

Implementação de Controle Fuzzy Em CLP Industrial [Distrito Federal] 2016.

xi, 69p., 210 x 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2016).

Trabalho de Graduação – Universidade de Brasília, Faculdade de Tecnologia.

1. Fuzzy

2. Controle

3. Quatro-Tanques

4. CLP

I. Mecatrônica/FT/UnB

II. Título (série)

## **REFERÊNCIA BIBLIOGRÁFICA**

ROCHA, J.M. (2016). Implementação de Controle Fuzzy Em CLP Industrial, Trabalho de Graduação em Engenharia de Controle e Automação, Publicação TG-037/2016, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 69p.

## **CESSÃO DE DIREITOS**

AUTOR: Jhonantans Moraes Rocha

TÍTULO: Implementação de Controle Fuzzy Em CLP Industrial.

GRAU: Engenheiro ANO: 2016

É concedida à Universidade de Brasília permissão para reproduzir cópias desta trabalho de graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa trabalho de graduação pode ser reproduzida sem autorização por escrito do autor.

---

Jhonantans Moraes Rocha

Faculdade de Tecnologia - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

*Aos meus irmãos, Jefferson e Jéssica,  
são a alegria da minha vida.*

## **AGRADECIMENTOS**

*Agradecimentos*

---

## RESUMO

Plantas descritas matematicamente por sistemas não-lineares apresentam desafios para modelagem de sistemas e para aplicação de técnicas de controle convencionais. O procedimento mais simples nestas situações é aproximar os sistemas a um estado local, linearizado, e assumir este comportamento pontual como o global do sistema. Sabe-se que esta abordagem fornece resultados que se afastam dos reais à medida que o estado do sistema distoa daquele ponto de referência. Abordagens fuzzy, como os modelos propostos por Takagi-Sugeno, são alternativas geralmente mais eficazes para solução deste problema, uma vez que fazem uma interpolação de várias modelagens em diversos pontos locais.

Este trabalho faz uso da lógica fuzzy, utilizando modelos Takagi-Sugeno, para desenvolver controladores para uma planta de quatro-tanques, um sistema não-linear com acoplamento entre suas variáveis. O objetivo final é implementar este controlador em um CLP industrial e observar seu desempenho.

---

## ABSTRACT

Plants mathematically described by non-linear systems present challenges for system modeling and for applying conventional control techniques. The simplest procedure in these situations is to approximate these systems to a linear local state and assume this punctual behavior as a global system. It is known that this approach yields results that deviate from the real ones as the state of the system moves away from that reference point. Fuzzy approaches, such as the models proposed by Takagi-Sugeno, are generally more effective alternatives to solve this problem, since they interpolate several models in several local points.

This work makes use of fuzzy logic, using Takagi-Sugeno models, to develop controllers for a four-tank plant, a nonlinear system with degrees of coupling between its variables. The final goal is to implement this controller in an industrial PLC and observe its performance.

<b>1 INTRODUÇÃO .....</b>	<b>1</b>
1.1 OBJETIVOS .....	1
1.2 ORGANIZAÇÃO DO TRABALHO .....	2
<b>2 DESCRIÇÃO DO SISTEMA.....</b>	<b>3</b>
2.1 SISTEMA DE QUATRO-TANQUES .....	3
2.2 CLP ROCKWELL 1756-L62.....	4
2.2.1 INSTALAÇÃO .....	5
2.2.2 INTEGRAÇÃO .....	7
2.3 RSLINX E RSLOGIX .....	8
2.3.1 LINGUAGEM LADDER .....	8
2.3.2 BLOCOS FUNCIONAIS .....	9
2.3.3 TEXTO ESTRUTURADO .....	10
<b>3 SISTEMAS FUZZY .....</b>	<b>11</b>
3.1 CONJUNTOS FUZZY .....	11
3.1.1 VARIÁVEIS LINGUÍSTICAS .....	11
3.2 FUNÇÕES DE PERTINÊNCIA .....	12
3.3 APLICAÇÃO .....	14
3.3.1 REGRAS SE-ENTÃO .....	14
3.4 MODELO FUZZY TAKAGI-SUGENO .....	15
<b>4 CONTROLE FUZZY .....</b>	<b>17</b>
4.1 SISTEMA AUMENTADO .....	17
4.2 MÉTODO DE LYAPUNOV .....	17
4.3 ESTABILIDADE DE SISTEMAS .....	18
4.4 ESTABILIDADE FUZZY .....	19
<b>5 MODELAGEM.....</b>	<b>20</b>
5.1 MODELO NÃO LINEAR .....	20
5.2 LINEARIZAÇÃO .....	20
5.3 MODELAGEM FUZZY TAKAGI-SUGENO .....	21
5.3.1 VARIÁVEIS LINGUÍSTICAS .....	21
5.3.2 PERTINÊNCIA .....	21
5.3.3 REGRAS SE-ENTÃO .....	22
5.3.4 ATIVAÇÃO .....	23
5.3.5 MODELO FINAL .....	23
5.3.6 CONTROLADOR FUZZY.....	24
<b>6 IMPLEMENTAÇÃO .....</b>	<b>25</b>

6.1 IDENTIFICAÇÃO.....	25
6.1.1 IDENTIFICAÇÃO CAIXA-PRETA .....	25
6.1.2 IDENTIFICAÇÃO CAIXA-CINZA .....	30
6.2 CLP .....	32
<b>7 RESULTADOS.....</b>	<b>34</b>
7.1 SIMULAÇÕES.....	34
7.1.1 FASE MÍNIMA .....	34
7.1.2 FASE NÃO-MÍNIMA.....	42
7.2 IMPLEMENTAÇÃO NA BANCADA .....	48
<b>8 CONCLUSÃO.....</b>	<b>51</b>
8.1 OBJETIVOS .....	51
8.2 TRABALHOS FUTUROS .....	51
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>52</b>
<b>A CÓDIGOS UTILIZADOS.....</b>	<b>53</b>
A.1 MATLAB .....	53
A.1.1 PLANTA SIMULADA.....	53
A.1.2 MODELO TS COM 4 REGRAS .....	53
A.1.3 FUNÇÃO DE CÁLCULO DOS "R" SISTEMAS .....	55
A.1.4 FUNÇÃO DE CÁLCULO DA PERTINÊNCIAS.....	56
A.1.5 SIMULINK DE COMUNICAÇÃO OPC .....	58
A.2 TEXTO ESTRUTURADO .....	58
A.2.1 SUBROTINA: FUZZYINPUTCALC .....	58

## LISTA DE FIGURAS

2.1	DIAGRAMA ESQUEMÁTICO DO SISTEMA DE QUATRO TANQUES E PLANTA DIDÁTICA.	3
2.2	PLANTA DE QUATRO-TANQUES NO LARA.....	4
2.3	ESTAÇÃO DE TRABALHO.....	5
2.4	INTERIOR DO PAINEL.....	6
2.5	MÓDULOS DO PAINEL.....	7
2.6	INTERFACE DO SOFTWARE RSLINX.....	8
2.7	EXEMPLO DE DIAGRAMA LADDER.....	9
2.8	EXEMPLO DE DIAGRAMA DE BLOCOS.....	9
2.9	EXEMPLO DE TEXTO ESTRUTURADO.....	10
3.1	FUNCÕES DE PERTINÊNCIA.....	12
3.2	FUNCÕES DE PERTINÊNCIA.....	13
4.1	SISTEMA EM MALHA FECHADA .....	17
5.1	FUNÇÕES DE PERTINÊNCIA.....	22
5.2	ESPAÇO DE ESTADOS DA PLANTA CONTROLADA .....	24
6.1	IDENTIFICAÇÃO 1 - 4000 SEGUNDOS .....	26
6.2	IDENTIFICAÇÃO 2 - 6000 SEGUNDOS .....	26
6.3	IDENTIFICAÇÃO 3 - 6000 SEGUNDOS .....	27
6.4	IDENTIFICAÇÃO 1 - 4000 SEGUNDOS .....	28
6.5	IDENTIFICAÇÃO 2 - 6000 SEGUNDOS .....	29
6.6	IDENTIFICAÇÃO 3 - 6000 SEGUNDOS .....	30
6.7	PROGRAMA PRINCIPAL .....	32
6.8	VARIÁVEIS DO PROGRAMA .....	33
7.1	NÍVEL DO TANQUE 1 .....	35
7.2	NÍVEL DO TANQUE 2 .....	36
7.3	COMPARAÇÃO COM 25 REGRAS .....	36
7.4	COMPARAÇÃO COM 225 REGRAS.....	37
7.5	ESPAÇO DE ESTADOS DA PLANTA CONTROLADA .....	38
7.6	NÍVEL H1 CONTROLADO .....	38
7.7	NÍVEL H2 CONTROLADO .....	39
7.8	NÍVEL H1 CONTROLADO - COM SATURAÇÃO DO CONTROLADOR .....	40
7.9	NÍVEL H2 CONTROLADO - COM SATURAÇÃO DO CONTROLADOR .....	40
7.10	NÍVEL H1 CONTROLADO - Com <i>Anti-Windup</i> .....	41
7.11	NÍVEL H2 CONTROLADO - Com <i>Anti-Windup</i> .....	42
7.12	NÍVEL DO TANQUE 1 .....	43
7.13	NÍVEL DO TANQUE 2 .....	44
7.14	COMPARAÇÃO COM 25 REGRAS TS .....	44

***LISTA DE FIGURAS***

v

7.15 COMPARAÇÃO COM 225 REGRAS TS .....	45
7.16 NÍVEL H1 CONTROLADO .....	46
7.17 NÍVEL H2 CONTROLADO .....	46
7.18 NÍVEL H1 CONTROLADO - COM SATURAÇÃO DO CONTROLADOR .....	47
7.19 NÍVEL H2 CONTROLADO - COM SATURAÇÃO DO CONTROLADOR .....	47
7.20 NÍVEL H1 CONTROLADO - COM <i>Anti-Windup</i> .....	48
7.21 NÍVEL H2 CONTROLADO - COM <i>Anti-Windup</i> .....	48
7.22 NÍVEL H1 CONTROLADO - REAL .....	49
7.23 NÍVEL H2 CONTROLADO - REAL .....	49
A.1 COMUNICAÇÃO OPC SIMULINK-RSLINX .....	58

## LISTA DE TABELAS

2.1	ESPECIFICAÇÕES INICIAIS DA PLANTA.....	4
2.2	MÓDULOS 1756 INSTALADOS. ....	6
2.3	IPs DOS DISPOSITIVOS.....	7
3.1	TABELA DE EXEMPLOS .....	13
6.1	TENSÕES ESCOLHIDAS .....	31
6.2	PONTOS DE OPERAÇÃO DA PLANTA INSTALADA .....	31
6.3	GANHOS APLICADOS À PLANTA .....	31
7.1	PARÂMETROS DA PLANTA EM FASE MÍNIMA.....	34
7.2	PONTOS DE OPERAÇÃO .....	35
7.3	GANHOS OBTIDOS .....	37
7.4	PARÂMETROS DA PLANTA EM FASE NÃO-MÍNIMA.....	42
7.5	PONTOS DE OPERAÇÃO .....	43
7.6	GANHOS OBTIDOS .....	45

**Símbolos Latinos**

$v_i$	Tensão aplicada à Bomba $i$
$k_i$	Constante de Fluxo da Bomba $i$
$g$	Constante Aceleração da Gravidade
$H_i$	Altura do Tanque $i$
$h_i$	Altura do fluído do Tanque $i$
$A_i$	Área da secção transversal do Tanque $i$
$a_i$	Área do orifício de saída do Tanque $i$

**Símbolos Gregos**

$\gamma_i$	Abertura da válvula $i$
------------	-------------------------

## NOTAÇĀO

Neste trabalho utiliza-se as denominações lógica fuzzy, lógica nebulosa e lógica difusa como sinônimos.

MIMO Múltiplas Entradas Múltiplas Saídas (*Multiple Inputs Multiple Outputs*)

$M'$  Transposta da matriz  $M_i$

LMI Linear Matricial Inequalities (Desigualdade Linear Matricial)

TS Sistema Takagi Sugeno

CLP Controlador Lógico Programável

# 1 INTRODUÇÃO

Não há assunto tão velho que não possa ser dito algo de novo sobre ele.

---

Fiodor Dostoievski

Desenvolver controladores para sistemas não-lineares é quase sempre uma tarefa dispendiosa e complexa. Para plantas multivariáveis não-lineares este desafio é ainda maior. É por este motivo que é prática comum recorrer-se à linearização das equações que as descrevem, obtendo uma aproximação do sistema inicial num formato que se encaixa às teorias de controle convencionais para sistemas lineares.

Esta linearização simples, realizada por meio da série de Taylor, resulta numa aproximação local do sistema não-linear. No entanto, à medida que as variáveis controladas e manipuladas se afastam deste ponto de operação, condição na qual foi realizada a linearização, o modelo passa a se afastar da planta real.

Neste cenário, a abordagem fuzzy figura como excelente ferramenta para solução destes desvios. Aparecendo pela primeira vez nos trabalhos de Zadeh [1], foi desenvolvida para aplicações em modelagem de sistemas nos trabalhos de Takagi e Sugeno [2]. Seus métodos consistem na linearização convencional do sistema em múltiplos pontos escolhidos criteriosamente, baseados em um conjunto de métricas relevantes para o problema em questão. A partir daí desenvolve-se um conjunto de regras para determinar o grau de conformidade de cada estado do sistema à cada um dos pontos pré-modelados. Utiliza-se então como modelo a soma (uma interpolação) dos múltiplos modelos iniciais ponderada por estes coeficiente de pertinência.

O objeto de estudo deste trabalho é o sistema de quatro tanques, desenvolvido por Karl Johansson [3] com o objetivo didático de demonstrar de forma ilustrativa conceitos e propriedades de sistemas com múltiplas entradas e saídas (MIMO, do inglês *Multiple Input, Multiple Output*). Ele consiste em quatro tanques interconectados, um reservatório inferior, duas válvulas esféricas e duas bombas de corrente contínua que bombeiam o fluido do reservatório inferior para os tanques de forma cruzada, de acordo com a razão entre os fluxos definida pela posição das válvulas. O sistema de quatro tanques é não linear. Seu modelo linearizado apresenta um zero multivariável que pode estar localizado tanto no semi-plano esquerdo quanto no direito dependendo da configuração das válvulas. A abertura delas determina se o sistema é de fase mínima ou de fase não-mínima afetando a dinâmica geral entre entradas e saídas.

## 1.1 OBJETIVOS

O objetivo geral é desenvolver um controlador fuzzy, baseado no modelo Takagi-Sugeno da planta, capaz de controlar os níveis do fluido nos tanques inferiores 1 e 2. As variáveis manipuladas do processo são somente as tensões de entrada das bombas, que influenciam de maneira proporcional-

mente direta no fluxo.

Como objetivos específicos apresentam-se a instalação do CLP e sua integração. Segue-se então a modelagem e linearização da planta de quatro tanques. Em seguida, a identificação do modelo Takagi-Sugeno proposto juntamente com as funções de pertinência a serem utilizadas, das regras e dos modelos que as compõem. Após a simulação e validação, propõe-se o desenvolvimento de ganhos que estabilizem o sistema em malha fechada, seguindo as referências desejadas. Por fim, objetiva-se a identificação da planta real em 4 pontos a serem utilizados para determinação de seu modelo TS e de ganhos. Finaliza-se com sua implementação e observação dos resultados.

## 1.2 ORGANIZAÇÃO DO TRABALHO

Os capítulos iniciais deste trabalho tratam da teoria fuzzy e sua aplicação em sistemas controlados. Já os capítulos finais aplicam essa teoria diretamente sobre a bancada de quatro-tanques e por meio de LMIs são desenvolvidos controladores para ela. No Capítulo 2 são apresentados a planta estudada e o CLP Rockwell onde os algoritmos são implementados. Em seguida, o Capítulo 3 apresenta uma introdução aos conceitos da lógica e modelagem fuzzy e como aplicá-los. No Capítulo 5 são realizadas as três formas de modelagem do sistema abordadas neste trabalho: não-linear, linear e o modelo Takagi-Sugeno. No Capítulo 4 o projeto do controlador é desenvolvido, seguindo os conceitos de estabilidade baseados em desigualdade lineares matriciais. O Capítulo 6 apresenta a implementação dos algoritmos no CLP, utilizando as linguagens de programação aceitas por este. No Capítulo 7 são apresentados os resultados das simulações e do sistema real. Por fim as considerações finais são apresentadas no Capítulo 8.

## 2 DESCRIÇÃO DO SISTEMA

É um erro terrível teorizar antes de termos informação.

Sir Arthur Conan Doyle

Sistemas controlados são constituídos essencialmente por uma ou mais plantas as quais se deseja controlar por meio de um dispositivo que implementará os algoritmos de controle aplicados à ela. O objeto de estudo neste trabalho é uma planta de Quatro-Tanques, descrita na seção a seguir. Apresenta-se por fim o dispositivo controlador utilizado, é um CLP Rockwell 1756-L62, apresentado na seção homônima logo em seguida.

### 2.1 SISTEMA DE QUATRO-TANQUES

Em 1999, Karl Henrick Johansson publicou o artigo "Theaching Multivariable Control Using the Quadruple-Tank System" [3], onde são apresentadas as ideias do sistema de quatro-tanques estudado neste trabalho. Trata-se de um laboratório didático de processo multivariável capaz de demonstrar dinâmicas de zeros alocáveis em fases mínima e não-mínima. O sistema não linear ilustra claramente os problemas de controle MIMO (Multiples Inputs Multiples Outputs) com acoplamento.

Seu diagrama esquemático é apresentado na Figura 2.1 a seguir. Consiste em quatro tanques interconectados, um reservatório inferior, quatro válvulas esféricas e duas bombas de corrente contínua.

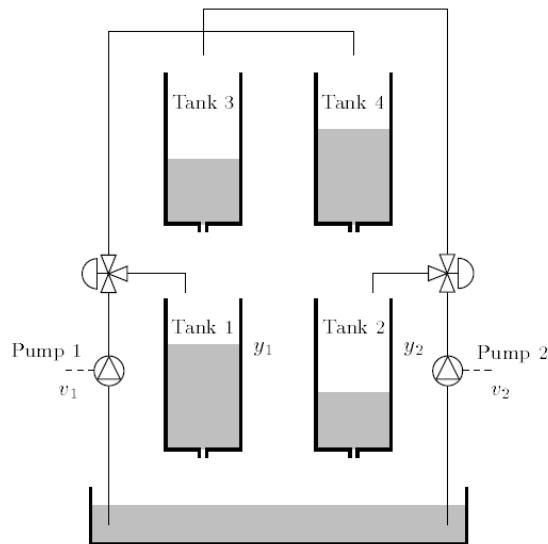


Figure 2.1: Diagrama esquemático do sistema de quatro tanques e planta didática.

As bombas impulsionam o fluído do reservatório para o sistemas: bomba 1 para os tanques 1 e 4,

bomba 2 para os tanques 2 e 3. As válvulas definem a proporção direcionada entre os tanques inferior e superior de cada rota.

A Imagem 2.2 a seguir apresenta a planta utilizada neste experimento, localizada no LARA (Laboratório de Automação e Robótica) - SG-11 (UnB) .



Figure 2.2: Planta de Quatro-Tanques no LARA.

Suas dimensões aferidas são apresentadas na Tabela 2.1, onde  $A_i$  e  $H_i$  representam a área da secção transversal da base do tanque e o nível máximo do tanque  $i$ ,  $i = 1, 2, 3, 4$ .

Table 2.1: Especificações Iniciais da Planta.

Especificações Iniciais da Planta	
$A_1, A_3$	47,6 cm <sup>2</sup>
$A_2, A_4$	47,6 cm <sup>2</sup>
$H_1, H_2, H_3, H_4$	24 cm
$g$	981 cm/s <sup>2</sup>

## 2.2 CLP ROCKWELL 1756-L62

Controladores Lógico Programáveis (CLP) são largamente utilizados para controle de processos e automação industrial atualmente. Trata-se de um equipamento eletrônico digital com hardware e software adaptados para as condições industriais. Utilizam uma memória programável para armazenar

instruções de controle e conexões com diversos módulos para interface com processos externos, entrada e saída de dados, comunicação digital, entre diversas outras funções.

### 2.2.1 Instalação

Neste trabalho realizou-se a montagem de toda a estação de controle. Assim, escolheu-se primeiramente um local adequado para a disposição do painel de controle: próximo à planta e ao microcomputador ao qual se conecta, porém afastado de fiações elétricas ou locais úmidos. Outro cuidado deve de ser observado durante a instalação da fonte junto ao chassi, observando a compatibilidade com as tensões de entrada e saída do controlador. Seguiu-se fixação do painel no local escolhida, instalação do microcomputador à ser utilizado e instalação da fiação elétrica. Observa-se na Figura 2.3 o resultado instalado.



Figure 2.3: Estação de trabalho.

A Figura 2.4 a seguir ilustra o interior do painel, já com o chassi do controlador instalado e as trilhas utilizadas distribuídas no espaço restante para conexão dos bornes a serem utilizados no projeto.



Figure 2.4: Interior do painel.

Os módulos de entrada e saída foram instalados conforme a Tabela 2.2 abaixo.

Table 2.2: Módulos 1756 instalados.

Especificação	Descrição	Posição no chassi
1756-A7/B	Chassi	.
1756-L62	Controlador	0
1756-ENBT/A	EtherNetIp	1
1756-IF8/A	Entradas Analógicas	2
1756-OF8/A	Saídas Analógicas	3
1756-IB16/A	Entradas DC	4
1756-OB8I/A	Saídas DC	5

Observa-se na Figura 2.5 a seguir a configuração instalada.



Figure 2.5: Módulos do painel.

### 2.2.2 Integração

Seguiu-se a configuração dos módulos de comunicação com o CLP. Dois modos são disponíveis com os módulos utilizados: serial, realizada diretamente com o controlador, e Ethernet, através do módulo EtherNetIP. Ambas foram implementadas e testadas.

Para comunicação serial, basta configurar a entrada serial no computador a ser utilizado e em seguida configurar o controlador no software RSLinx [4]. Para utilizar a comunicação EtherNetIp é necessário antes configurar o módulo EhterNetIp [5]. O software BOOTP/DHCP torna possível assinar um endereço IP para o módulo recém instalado. Para que a comunicação em uma rede EtherNetIp ocorra corretamente todos os dispositivos dela precisam possuir endereços IP seguindo o padrão definido pela máscara de sub-rede, neste caso, 255.255.255.0. Isso significa, basicamente, que os pontos comunicantes da rede devem possuir ids únicos apenas nos último octeto de seus endereços. A tabela a seguir apresenta os endereços utilizados, bem como a configuração padrão da rede.

Table 2.3: IPs dos dispositivos

Dispositivo	Endereço
PC (RSLinx)	192.168.2.1
1756-ENBT/A (CLP)	192.168.2.22
Geral	192.168.2.xxx

Um **importante** cuidado de segurança observado foi o aterramento de diversos elementos do

equipamento. É conhecida sua capacidade de operação em condições adversas, mesmo assim, como precaução houve o cuidado de aterrarr o chassi, a placa onde foi instalado e o painel exterior.

## 2.3 RSLINX E RSLOGIX

Os principais softwares utilizados para implementação do controlador são o RSLinx e o RSLogix. O primeiro é responsável por estabelecer a comunicação com o CLP Rockwell e sua ampla variedade de aplicativos e módulos. A Figura 2.6 apresenta sua interface, onde é possível visualizar o controlador e os módulos instalados. A partir deste menu são acessíveis diversas funções de configuração e exibição das informações dos dispositivos.

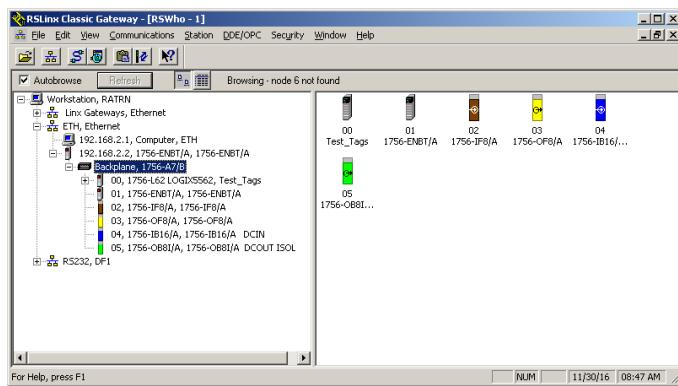


Figure 2.6: Interface do Software RSLinx.

Já o RSLogix 5000 é o ambiente de desenvolvimento proprietário da Rockwell (Allen-Bradley). Neste trabalho, os controladores foram desenvolvidos no RSLogix e enviados (Download) para o CLP em conjunto com o RSLinx. As seções a seguir apresentam as linguagens de desenvolvimento disponibilizadas pelo RSLogix, a saber linguagem Ladder, blocos de funções e texto estruturado.

### 2.3.1 Linguagem Ladder

A linguagem Ladder é a pioneira dos CLPs por se tratar de uma evolução natural de diagramas elétricos, utilizados antes da chegada dos controladores digitais. Seu ambiente de desenvolvimento utiliza o posicionamento de símbolos e blocos para implementação da lógica de controle. O ambiente inicial é formado por duas linhas verticais, que representam nível lógico alto (à esquerda) e baixo (à direita) de um sistema. Entre essa linhas são desenhados ramais horizontais que representam os estados do CLP.

Uma forma de compreender essa linguagem seria como uma série de conexões de contatos e bobinas. Se for possível traçar um caminho da esquerda para direita, conectando-se à uma bobina de saída ao final, então o valor dessa bobina será verdadeiro. Trabalhando-se com controladores digitais, são criadas variáveis no programa que representam diretamente os valores presentes nos módulos de saída e entrada. Essas variáveis recebem o nome de TAGs. Assim, as variáveis de entradas são assinaladas à tags utilizadas como chaves e as variáveis de saídas à tags associadas às bobinas de saída.

Percorrendo-se o caminho da esquerda para direita em um ramal, ao se chegar à uma chave, observa-se se o valor assinalado à ela é verdadeiro. Caso seja, continua-se o caminho até uma bobina de saída. Se está for alcançada, seu valor é setado para verdadeiro, consequentemente a saída associada a ela recebe a tensão associado à este nível lógico no controlador.

A Figura 2.7 à seguir ilustra um exemplo.



Figure 2.7: Exemplo de Diagrama Ladder.

### 2.3.2 Blocos Funcionais

Trata-se de outra linguagem de programação gráfica disponível aos CLPs Rockwell. É bastante semelhante à observada em vários outros softwares comuns ao meio acadêmico, como o MATLAB. Para sua utilização, assinala-se tags às entradas e saídas dos módulos já adicionados ao projeto. O desenvolvimento utiliza blocos de entradas e saídas associados à essas variáveis. Conexões entre os blocos, por meio de linhas representam passagem dos valores por esses fios. A lógica de controle é feita por meio de blocos de funções, estes possuem uma ou mais entradas e uma ou mais saídas. Os valores assinalados à suas saídas são determinados pelas funções às quais estão associados e que utilizam os valores de entrada como argumentos.

A Figura 2.8 à seguir ilustra um exemplo.

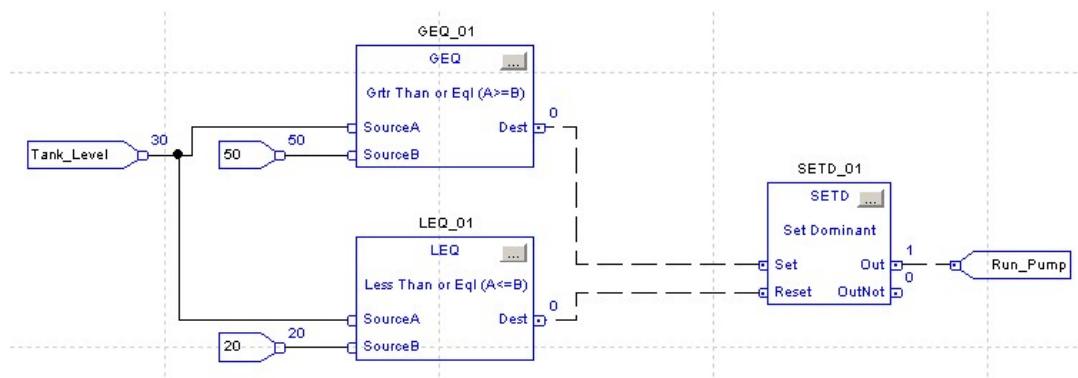


Figure 2.8: Exemplo de diagrama de blocos.

Observa-se o caso do bloco GEQ\_01. Suas entradas são o nível do tanque (SourceA = Tank\_Level) e um valor constante (SourceB = 50). Sua saída (dest) será assinalada com nível lógico verdadeiro

apenas quando  $SourceA \geq SourceB$ , ou seja,  $Dest = 1$  se  $Tank\_Level \geq 50$ , sendo  $Dest = 0$  caso contrário.

### 2.3.3 Texto Estruturado

A linguagem texto estruturado é muito semelhante às linguagens estruturais C e Pascal. Como elas, é baseada no uso simples de comandos que são executados sequencialmente em seu desenvolvimento. Da mesma forma que as anteriores, esta linguagem utiliza Tags como variáveis e é a partir delas que se faz a leitura das entradas e definem-se as saídas.

A Figura 2.9 à seguir ilustra um exemplo.

```
SBR(h1, h2);
  FOR row:=0 TO 1 BY 1 DO
    FOR col:=0 TO 5 BY 1 DO
      K[row,col] := alphal*k1[row,col];
    END_FOR;
  END_FOR;

  IF (h1 < h2) THEN
    Bombal := 0;
  ELSE
    Bombal := 50;
  END_IF;

RET();
```

Figure 2.9: Exemplo de Texto Estruturado.

### 3 SISTEMAS FUZZY

As respostas residem nas coisas que julgamos simples.

---

Steins; Gate

A lógica fuzzy, ou difusa, foi introduzida originalmente por Lofti A. Zadeh, em seu artigo "Fuzzy sets and systems" [1]. Sua teoria diverge da lógica booleana convencional no tratamento da pertinência das variáveis, podendo assumir qualquer valor entre todos os possíveis de um intervalo. Essa abordagem é mais eficaz na descrição de alguns sistemas reais, uma vez que é praticamente impossível eliminar todas as incertezas nos modelos que os representam. Este capítulo apresenta os fundamentos desse paradigma bem como sua aplicação na modelagem, proposta por Takagi e Sugeno [2].

#### 3.1 CONJUNTOS FUZZY

De acordo com a teoria de conjuntos clássica, um elemento  $x$  qualquer, pode pertencer ou não à um conjunto universo de discurso  $U$ , ou seja  $x \in U$  ou  $x \notin U$ . Portanto, para qualquer conjunto determinado, pode-se estar completamente dentro ou totalmente fora dele.

$$f_u(x) : U \rightarrow \{0, 1\} \quad f_u(x) = \begin{cases} 1 & \text{se e somente se } x \in U \\ 0 & \text{caso contrário} \end{cases} \quad (3.1)$$

Essa definição binária se encaixa bem em problemas restritos, cujo caráter dos sistemas reflete essa separação clara de estados, por exemplo a paridade ou não de uma da soma dos bits de uma mensagem binária. Conhecendo-se os valores, este resultado é ímpar ou par, indubitavelmente. No entanto, grande parte dos sistemas estudados nas teorias de controle trabalha com grandezas que não possuem limites tão claros assim, como exemplo a sensação térmica. Apesar de a temperatura ser matematicamente bem definida, existem descrições como "frio" e "quente" que não podem ser representadas com este conjunto binário, uma vez que são conceitos vagos e imprecisos.

A abordagem fuzzy aparece como uma alternativa muito capaz de tratar estes casos. Seus conjuntos são caracterizados por uma função contínua de pertinência fuzzy, que relaciona cada elemento do universo de discurso à sua conformidade no conjunto, podendo abranger todos os valores no intervalo de pertinência.

##### 3.1.1 Variáveis Linguísticas

As variáveis linguísticas são os termos que constituem os conjuntos nebulosos. Tratam-se de traduções das variáveis reais na forma de valores linguísticos, não numéricos. Assim, seguindo com exemplo anterior, a temperatura seria a variável linguística e "quente", "frio", "muito quente" e "muito

frio" alguns de seus possíveis valores linguísticos. Estes últimos são os conjuntos difusos e possuem, cada um deles, uma função de pertinência mapeando a adequação de uma determinada temperatura a sua conformidade neles.

### 3.2 FUNÇÕES DE PERTINÊNCIA

O conceito chave de toda a abordagem fuzzy são as funções de pertinência. Em exemplo, dados os conjuntos fuzzy  $U_1$ ,  $U_2$  e  $U_3$ , cada qual possui sua função de pertinência  $f_1(x)$ ,  $f_2(x)$  e  $f_3(x)$ , para todo elemento pertencente ao universo de discurso.

$$f_i(x) : i \rightarrow [0, 1] \quad (3.2)$$

Onde  $f_i(x) = 0$  implica que o elemento  $x$  é "completamente não"  $U_i$  e  $f_i(x) = 1$  indica que  $x$  é "completamente"  $U_i$ . Mas, diferentemente da lógica convencional, é possível que um elemento seja 50% pertinente à  $U_1$  ( $f_1(x) = 0.5$ ), 30% à  $U_2$  ( $f_2(x) = 0.3$ ) e 20% à  $U_3$  ( $f_3(x) = 0.2$ ).

Apesar de operar sobre grandezas linguísticas, é importante notar que normalmente os elementos são variáveis numéricas, portanto as funções de pertinência precisam ser bem definidas no intervalo do conjunto. Os formatos mais comuns para elas são apresentados na Figura 3.1 a seguir:

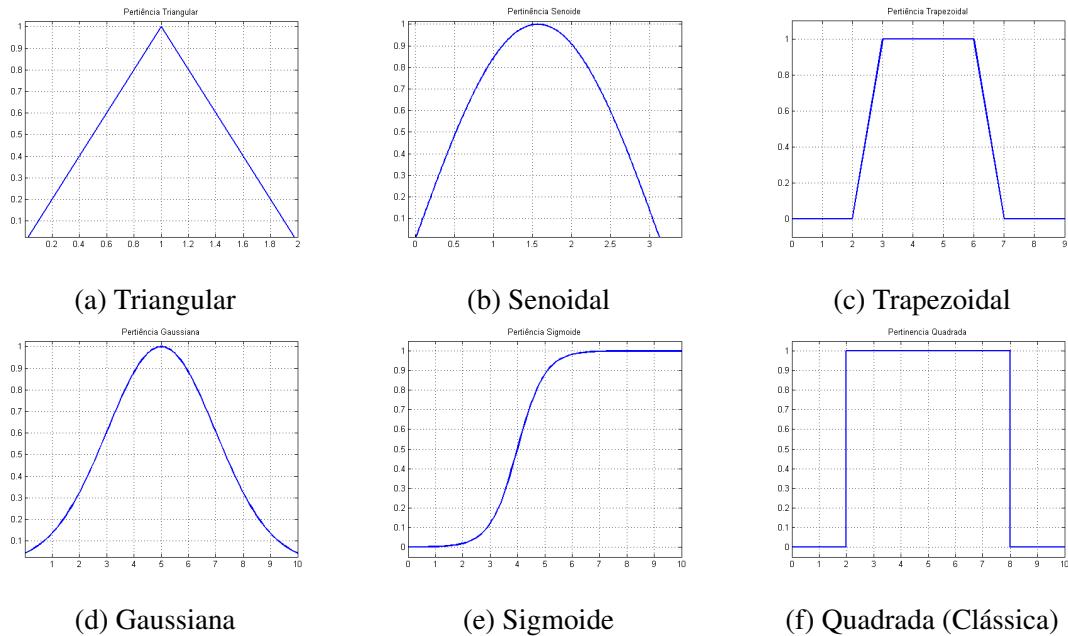


Figure 3.1: Funcões de Pertinênciа.

Apresenta-se a seguir os procedimentos comuns para obtenção da função de pertinência de um dado sistema, ilustrando-se com o exemplo:

- **Definir a variável linguística:** "Temperatura"
- **Definir os conjuntos fuzzy:** {"muito frio"}, {"frio"}, {"quente"}, {"muito quente"}

- **Definir os limites de cada conjunto:**  $[-\infty, 10C], [5C, 25C], [15C, 35C], [25C, \infty]$
- **Definir as funções de pertinência:** Neste caso opta-se por funções triangulares, com picos nos centros dos intervalos e nulas em qualquer caso fora deles.

Os resultados do exemplo são apresentados na Figura 3.2 a seguir:

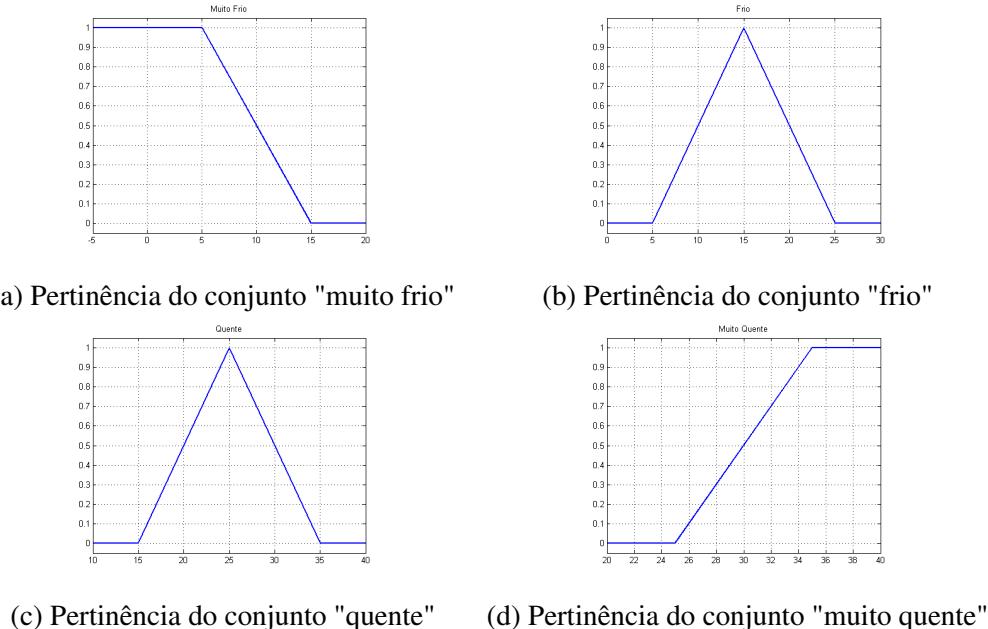


Figure 3.2: Funções de Pertinência.

A Tabela 3.1 a seguir apresenta os graus de pertinência de várias amostras a cada conjunto:

Table 3.1: Tabela de Exemplos

Temperatura ( $^{\circ}C$ )	Muito Frio	Frio	Quente	Muito Quente
0	1	0	0	0
5	1	0	0	0
10	0.5	0.5	0	0
15	0	1	0	0
20	0	0.5	0.5	0
25	0	0	1	0
30	0	0	0.5	0.5
35	0	0	0	1
40	0	0	0	1

É importante notar que a soma final dos valores de todas as pertinências de um elemento precisa ser 1, para que haja coerência entre o modelo e o real.

### 3.3 APLICAÇÃO

A aplicação da lógica fuzzy na teoria de controle se dá através da utilização de regras que definem o modelo final baseando-se no grau de pertinência do estado do sistema a cada um dos conjuntos difusos.

De maneira similar à tradicional, a lógica fuzzy baseia-se no paradigma de implicações, ou *modus ponens*. Esta linha de raciocínio é organizada em regras que implicam em conclusões a partir da autenticidade de premissas. O Exemplo 1 a seguir ilustra:

#### Exemplo 1

$$\left\{ \begin{array}{l} (1) \text{ Se está chovendo então é perigoso dirigir} \\ (2) \text{ Está chovendo} \\ (3) \text{ É perigoso dirigir} \end{array} \right.$$

A afirmação (1) é chamada de regra de implicação, ou regra Se-Então, e é ela quem rege o comportamento da conclusão (3) de acordo com a premissa (2). Ou seja, sempre que (2) é verdadeira, (3) também será.

##### 3.3.1 Regras Se-Então

Como visto no Exemplo 1 as regras Se-Então são parte das premissas que definem os resultados das afirmações. Uma vez que as funções de pertinência fuzzy assumem diferentes níveis de verdade, não binários, então a autenticidade de afirmações envolvendo-as também assumirá diferentes graus de verdade. Assim, ao contrário da lógica clássica onde há ou não ativação de uma determinada regra, em lógica difusa toda regra está ativada em determinado grau. Exemplifica-se a seguir:

$$\text{Regra: } \left\{ \begin{array}{l} \text{SE: } X \text{ pertence a } A \\ \text{ENTÃO: } Y \text{ pertence a } B \end{array} \right. \quad (3.3)$$

O grau de ativação da Regra na Equação 3.3 é definido a partir da pertinência do elemento  $X$  em  $A$ . A conclusão  $Y$  será definida de forma a cumprir pertinência semelhante ao conjunto  $B$ . Caso  $X$  seja 50% A, a saída  $Y$  deverá ser 50% B.

Prosseguindo o exemplo inicial, pode-se utilizar a temperatura de uma sala para controlar a potência ativa de um ar condicionado. Segundo os procedimentos descritos, define-se uma nova variável linguística: "Potência" e seus conjuntos fuzzy: {"muito fraca", "fraca", "forte", "muito forte"}. Chamando "T" a temperatura atual e "P" a potência, uma forma simples de projeto poderia ser:

- Regra 1:  $\begin{cases} \text{SE: } T \text{ pertence a "muito baixa"} \\ \text{ENTÃO: } P \text{ pertence a "muito fraca"} \end{cases}$
- Regra 2:  $\begin{cases} \text{SE: } T \text{ pertence a "baixa"} \\ \text{ENTÃO: } P \text{ pertence a "fraca"} \end{cases}$
- Regra 3:  $\begin{cases} \text{SE: } T \text{ pertence a "alta"} \\ \text{ENTÃO: } P \text{ pertence a "forte"} \end{cases}$
- Regra 4:  $\begin{cases} \text{SE: } T \text{ pertence a "muito alta"} \\ \text{ENTÃO: } P \text{ pertence a "muito forte"} \end{cases}$

### 3.4 MODELO FUZZY TAKAGI-SUGENO

Os trabalhos de Takagi, Sugeno [2] e Kang [6] introduziram e desenvolveram a aplicação da lógica difusa em sistemas dinâmicos. Neles é demonstrada a capacidade dos modelos fuzzy (aqui os trataremos por Takagi-Sugeno) de representarem, de forma tão aproximada quanto se queira, qualquer sistema dinâmico (respeito condições de domínio).

Suponha-se um sistema dinâmico qualquer a seguir:

$$\begin{aligned}\dot{x}(t) &= f(x(t)) + g(u(t)) \\ y(t) &= h(x(t))\end{aligned}$$

Linearizando o obteria-se o seguinte modelo em espaço de estados:

$$\begin{aligned}\dot{\Delta x}(t) &= A\Delta x(t) + B\Delta u(t) \\ y(t) &= C\Delta x(t)\end{aligned}$$

As regras fuzzy são:

$$\text{Regra i: } \begin{cases} \text{SE: } c_1(t) \text{ é } M_{1i} \text{ e } c_2(t) \text{ é } M_{2i} \text{ e ... e } c_n(t) \text{ é } M_{ni}, \\ \text{ENTÃO: } \begin{cases} \dot{\Delta x}(t) = A_i \Delta x(t) + B_i \Delta u(t), \\ y(t) = C_i \Delta x(t) \end{cases} \end{cases} \quad (3.4)$$

Onde  $c_j(t)$  é cada uma das variáveis nas quais são verificadas as pertinência  $f_{ij}$  ao conjunto  $M_{ji}$  correspondente, sendo  $i$  a regra em questão. Agrupando as premissas, tem-se:

$$c(t) = [c_1(t) \ c_2(t) \ c_3(t) \ ... \ c_n(t)] \quad (3.5)$$

O grau de ativação de cada regra é dado pelo produto das  $n$  pertinências que compõem seus antecedentes:

$$w_i(c(t)) = \prod_{j=1}^n f_{ij}(c_j(t)) \quad (3.6)$$

O sistema final é dado pela média ponderada dos consequentes das r regras por seus graus de ativação. Assim, o modelo Takagi-Sugeno final é dado por:

$$\dot{x}(t) = \frac{\sum_{i=1}^r w_i(c(t))(A_i x(t) + B_i u(t))}{\sum_{j=1}^r w_j(c(t))}$$

Simplificando o coeficiente de ponderação em cada regras, tem-se:

$$\alpha_i(h(t)) := \frac{w_i(h(t))}{\sum_{j=1}^r w_j(h(t))}$$

E finalmente:

$$\dot{x}(t) = \sum_{i=1}^r \alpha_i(c(t))(A_i x(t) + B_i u(t)) \quad (3.7)$$

## 4 CONTROLE FUZZY

Se você entende como o universo funciona,  
de certa forma pode controlá-lo.

Stephen Hawking

Neste capítulo, busca-se realizar a sintonia de controladores por realimentação de estados do modelo TS utilizado.

### 4.1 SISTEMA AUMENTADO

A Figura 4.1 apresenta o sistema em malha fechada.

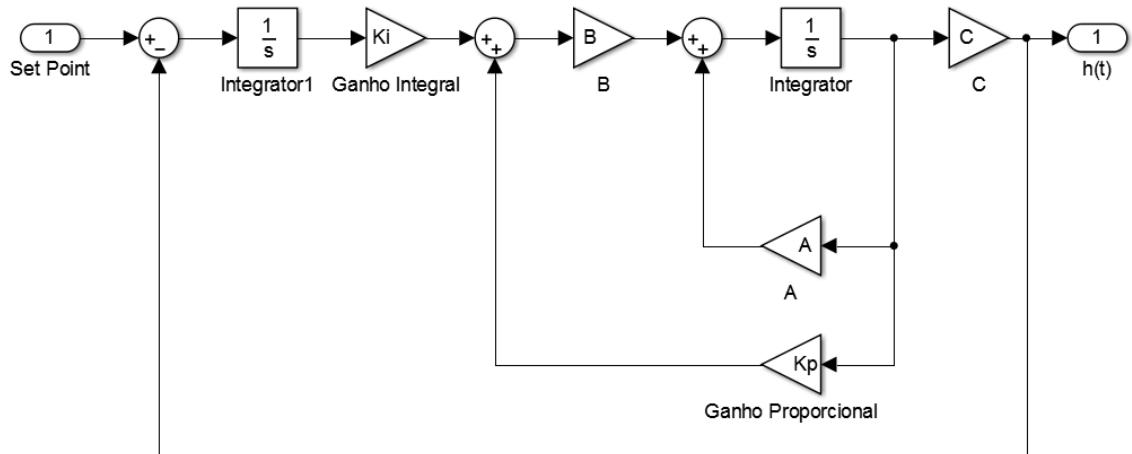


Figure 4.1: Sistema em Malha Fechada

O controlador realizado consiste em aplicar um ganho  $K = [K_p \ K_i]$  sobre o vetor espaço de estados. As condições buscadas são simplesmente erro nulo em regime permanente e estabilidade do sistema. Para a primeira, realiza-se a expansão do espaço de estado, acrescentando as integrais do erro em relação à referência para as duas variáveis controladas (níveis 1 e 2).

### 4.2 MÉTODO DE LYAPUNOV

A utilização de LMIs é uma abordagem prática e eficiente para garantir a estabilidade de sistemas fuzzy. Baseando-se no método de Lyapunov [8], desenvolvem-se as LMIs que aplicadas a cada um dos sistemas que compõem o TS final garantem sua estabilidade em malha fechada [7].

O método direto de Lyapunov é baseado na positividade de funções. Define-se então:

**Definição 1** Uma função escalar contínua  $w : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $w(0) = 0$  é semidefinida positiva se, e somente se

$$w(x) \geq 0, \forall x \in \mathbb{R}^n - 0 \quad (4.1)$$

Caso desigualdade seja estrita, então  $w$  será definida positiva. Uma função  $g$  é dita semidefinida (definida) negativa caso  $-g$  seja semidifinida (definida) positiva.

O método de Lyapunov consiste em encontrar uma função que seja sempre positiva e que decaia à medida que o estado do sistema avance. Sua formalização é dada pelo Teorema 1 a seguir:

**Teorema 1** Um sistema dinâmico autônomo é globalmente estável se existe uma função escalar  $V : \mathbb{R}^n \rightarrow \mathbb{R}$  tal que:

- $V$  é definida positiva
- $V$  possui derivada de primeira ordem
- $\dot{V}$  é definida negativa
- $V \rightarrow \infty$  a medida em que  $\|x\| \rightarrow \infty$

Assim, uma função  $V$  que satisfaça todo os requisitos do Teorema 1 para um dado sistema é chamada de função de Lyapunov.

### 4.3 ESTABILIDADE DE SISTEMAS

Para o sistema dinâmico contínuo e invariante no tempo definido pela Equação 4.2 a seguir:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (4.2)$$

Aplicando-se o teorema de Lyapunov [10], definindo uma matriz  $P$ , chega-se à:

**Teorema 2** O equilíbrio de um sistema dinâmico (4.2) é globalmente assintoticamente estável se existe uma matriz  $P$ , tal que:

$$\begin{aligned} P &= P' \\ P &> 0 \\ A'P + PA &< 0 \end{aligned}$$

#### 4.4 ESTABILIDADE FUZZY

Para um modelo TS, similar ao na Equação 3.7, tem-se:

$$\dot{x}(t) = \frac{\sum_{i=1}^4 w_i(c(t))(A_i x(t) + B_i u(t))}{\sum_{j=1}^4 w_j(c(t))}$$

Simplificando os termos de ativação de forma a obter um parâmetro A(c), definido como:

$$\begin{aligned}\alpha_i(c(t)) &:= \frac{w_i(c(t))}{\sum_{j=1}^r w_j(c(t))} \\ A(\alpha) &:= \sum_{i=1}^r \alpha_i(c(t)) A_i\end{aligned}$$

Aplicando-se a Teoria de Lyapunov, como proposto por Tanaka e Wang [10], uma vez que o modelo final é convexo, basta tratar a Equação 4.3 nos vértices do sistema, ou seja, em cada um dos sistemas que compõem as Regras 3.4. A análise computacional via LMIs possibilita a busca por uma matriz P que satisfaça essas condições, caso ela exista, prova-se a estabilidade do sistema. Assim, o sistema TS é globalmente assintoticamente estável [9] se existe solução para:

encontre  $P$

s.a  $P \succ 0$

$$A'_i P + P A_i \prec 0, \quad i = 1, 2, 3, \dots, r \quad (4.3)$$

Seguindo estas premissas, obtém-se o seguinte teorema:

**Teorema 3** *Dado um sistema fuzzy Takagi-Sugeno, convexo, composto por r regras e modelos correspondentes, em malha fechada com ganhos  $K_i = M_i X^{-1}$ , sua estabilidade é verificada [10] se existe solução para o problema*

encontre  $X = X'$ ,  $i = 1, 2, \dots, r$

$X \succ 0$

$$s.a \quad \begin{bmatrix} X & X A'_i - M'_i B'_i \\ A_i X - B_i M_i & X \end{bmatrix} \succ 0$$

(4.4)

## 5 MODELAGEM

Os modelos físicos do sistema de quatro-tanques utilizados neste trabalho são apresentados nas seções a seguir.

### 5.1 MODELO NÃO LINEAR

Baseado nos princípios de conservação de massa e na lei de Bernoulli para líquidos incompressíveis tem-se o seguinte sistema de equações não lineares que descrevem o processo.

$$\begin{cases} \dot{h}_1 = \frac{1}{A_1}(a_3\sqrt{2gh_3} + \gamma_1 k_1 v_1 - a_1\sqrt{2gh_1}) \\ \dot{h}_2 = \frac{1}{A_2}(a_4\sqrt{2gh_4} + \gamma_2 k_2 v_2 - a_2\sqrt{2gh_2}) \\ \dot{h}_3 = \frac{1}{A_3}((1 - \gamma_2)k_2 v_2 - a_3\sqrt{2gh_3}) \\ \dot{h}_4 = \frac{1}{A_4}((1 - \gamma_1)k_1 v_1 - a_4\sqrt{2gh_4}) \end{cases} \quad (5.1)$$

Na Equação 5.1, os termos  $h_i$ ,  $A_i$  e  $a_i$  representam o nível de água, a área da secção transversal da base do tanque  $i$  e a área de secção transversal do orifício de saída do tanque  $i$ ,  $i = 1, 2, 3, 4$ , respectivamente. A constante de fluxo da bomba  $j$  e a tensão aplicada sobre ela são dadas por  $k_j$  e  $v_j$ ,  $j = 1, 2$ . O parâmetro  $\gamma_1$  é a razão entre os fluxos para os tanques 1 e 4 enquanto  $\gamma_2$  é a razão entre os fluxos para os tanques 2 e 3 e  $g$  é a aceleração da gravidade.

É fácil notar nas equações deste sistema os termos não lineares (as raízes). O aspecto do acoplamento entre as variáveis também pode ser observado ao analisar as equações: o nível  $h_1$  varia conforme o fluxo da bomba 1, dependente de  $v_1$ , e conforme o nível  $h_3$ , que por sua vez depende do fluxo da bomba 2,  $v_2$ . Assim, as variáveis manipuladas,  $v_1$  e  $v_2$ , influem em ambos os níveis simultaneamente, apresentando um desafio considerável para a estabilização dos níveis desejados,  $h_1$  e  $h_2$ , que são as variáveis controladas deste trabalho.

### 5.2 LINEARIZAÇÃO

Linearizando o sistema em torno dos ponto de operação  $\bar{h} = (\bar{h}_1, \bar{h}_2, \bar{h}_3, \bar{h}_4)$  e  $\bar{v} = (\bar{v}_1, \bar{v}_2)$ , por expansão em série de Taylor, obtém-se a seguinte representação no espaço de estados:

$$\begin{bmatrix} \dot{\Delta h_1} \\ \dot{\Delta h_2} \\ \dot{\Delta h_3} \\ \dot{\Delta h_4} \end{bmatrix} = \begin{bmatrix} \frac{-a_1\sqrt{2g}}{2A_1\sqrt{h_1}} & 0 & \frac{a_3\sqrt{2g}}{2A_1\sqrt{h_3}} & 0 \\ 0 & \frac{-a_2\sqrt{2g}}{2A_2\sqrt{h_2}} & 0 & \frac{a_4\sqrt{2g}}{2A_2\sqrt{h_4}} \\ 0 & 0 & \frac{-a_3\sqrt{2g}}{2A_3\sqrt{h_3}} & 0 \\ 0 & 0 & 0 & \frac{-a_4\sqrt{2g}}{2A_4\sqrt{h_4}} \end{bmatrix} \begin{bmatrix} \Delta h_1 \\ \Delta h_2 \\ \Delta h_3 \\ \Delta h_4 \end{bmatrix} + \begin{bmatrix} \frac{\gamma_1 k_1}{A_1} & 0 \\ 0 & \frac{\gamma_2 k_2}{A_2} \\ 0 & \frac{(1-\gamma_2)k_2}{A_3} \\ \frac{(1-\gamma_1)k_1}{A_4} & 0 \end{bmatrix} \begin{bmatrix} \Delta v_1 \\ \Delta v_2 \end{bmatrix} \quad (5.2)$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = I \begin{bmatrix} \Delta h_1 \\ \Delta h_2 \\ \Delta h_3 \\ \Delta h_4 \end{bmatrix} \quad (5.3)$$

em que  $y_i$  são as saídas medidas do sistema,  $\Delta h_i = h_i - \bar{h}_i$ ,  $\Delta v_i = v_i - \bar{v}_i$ , e  $i = 1, 2, 3, 4$ .

E por fim, a matriz função de transferência do sistema obtida é:

$$G(s) = \begin{bmatrix} \frac{T_1 \gamma_1 k_1}{A_1(1+sT_1)} & \frac{T_1(1-\gamma_2)k_2}{A_1(1+sT_3)(1+sT_1)} \\ \frac{T_2(1-\gamma_1)k_1}{A_2(1+sT_4)(1+sT_2)} & \frac{T_2 \gamma_2 k_2}{A_2(1+sT_2)} \\ 0 & \frac{T_3(1-\gamma_2)k_2}{A_3(1+sT_3)} \\ \frac{T_4(1-\gamma_1)k_1}{A_4(1+sT_4)} & 0 \end{bmatrix} \quad (5.4)$$

em que  $G(s) = \frac{\Delta h(s)}{\Delta v(s)}$  e  $T_i = \frac{2A_i\sqrt{h_i}}{a_i\sqrt{2g}}$ ,  $i = 1, 2, 3, 4$ .

### 5.3 MODELAGEM FUZZY TAKAGI-SUGENO

A modelagem via Takagi-Sugeno segue os mesmos passos propostos na Seção 3.4: escolhem-se as variáveis linguística do sistema e seus conjuntos fuzzy, definem-se as funções de pertinência de cada um deles, as regras Se-Então, a fórmula de ativação e o resultado final do modelo.

#### 5.3.1 Variáveis Linguísticas

Como visto, as regras do modelo são ativadas de acordo com o estado atual do sistema. As variáveis aferidas da planta são os níveis, por este motivo serão as variáveis linguísticas definidas para o modelo fuzzy. Escolhem-se então "Nível do Tanque 1" e "Nível do Tanque 2" e os conjuntos nível baixo, nível alto para cada uma.

#### 5.3.2 Pertinência

Como visto, a linearização é baseada nos estados estacionários do sistema, ou seja, são escolhidos pontos em que  $\dot{h}(t) = 0$ . Assim, o modelo linear trata do valor de desvio das variáveis e é dado por:

$$\dot{\Delta h}(t) = A\Delta h(t) + B\Delta u(t) \quad (5.5)$$

Seguindo os passos descritos na Seção 3.2 definem-se os conjuntos escolhidos. Há quatro vértices para os estados do sistema:

$$\begin{cases} \text{Nível 1 Baixo} \\ \text{Nível 1 Alto} \end{cases} \quad \begin{cases} \text{Nível 2 Baixo} \\ \text{Nível 2 Alto} \\ \text{Nível 2 Baixo} \\ \text{Nível 2 Alto} \end{cases} \quad (5.6)$$

Obtemos um modelo linear em cada um destes vértices:

$$\Delta \dot{h}(t) = A_i \Delta h(t) + B_i \Delta u(t) \quad (5.7)$$

$$i = 1, 2, 3, 4 \quad (5.8)$$

Como descrito na Tabela 2.1, a altura dos tanques é de 25cm. Assim, escolhe-se os limitantes 5cm como "completamente" verdade para o nível baixo e 25cm como "completamente" verdade para o nível alto. As funções de pertinência obtidas são:

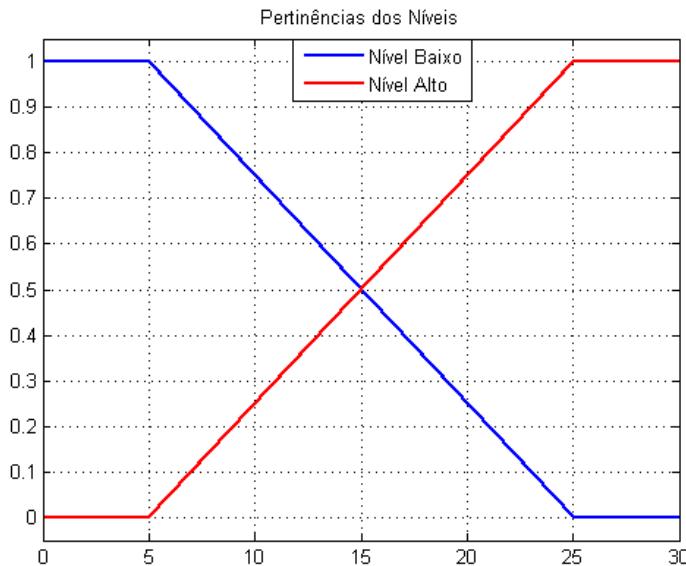


Figure 5.1: Funções de Pertinência.

Onde  $M_1(h_1(t))$  e  $M_2(h_1(t))$  é o grau de pertinência do nível  $h_1(t)$  aos conjuntos "baixo" e "alto", respectivamente. De maneira análoga, tem-se  $N_1(h_2(t))$  e  $N_2(h_2(t))$  para o nível  $h_2(t)$ .

### 5.3.3 Regras Se-Então

Haverá uma regra para cada um dos sistemas dados:

- Regra 1:

$$\begin{cases} \text{SE} & h_1 \text{ é baixo e } h_2 \text{ é baixo} \\ \text{ENTÃO} & \Delta\dot{h}(t) = A_1\Delta h(t) + B_1\Delta u(t) \end{cases}$$

- Regra 2:

$$\begin{cases} \text{SE} & h_1 \text{ é baixo e } h_2 \text{ é alto} \\ \text{ENTÃO} & \Delta\dot{h}(t) = A_2\Delta h(t) + B_2\Delta u(t) \end{cases}$$

- Regra 3:

$$\begin{cases} \text{SE} & h_1 \text{ é alto e } h_2 \text{ é baixo} \\ \text{ENTÃO} & \Delta\dot{h}(t) = A_3\Delta h(t) + B_3\Delta u(t) \end{cases}$$

- Regra 4:

$$\begin{cases} \text{SE} & h_1 \text{ é alto e } h_2 \text{ é alto} \\ \text{ENTÃO} & \Delta\dot{h}(t) = A_4\Delta h(t) + B_4\Delta u(t) \end{cases}$$

#### 5.3.4 Ativação

O nível de ativação de cada uma das **Regras**  $i$  é dado respectivamente por  $w_i$ :

$$\begin{aligned} w_1(t) &= M_1(h_1(t)) * N_1(h_2(t)) \\ w_2(t) &= M_1(h_1(t)) * N_2(h_2(t)) \\ w_3(t) &= M_2(h_1(t)) * N_1(h_2(t)) \\ w_4(t) &= M_2(h_1(t)) * N_2(h_2(t)) \end{aligned} \tag{5.9}$$

#### 5.3.5 Modelo Final

O peso de cada regra, na ponderação final, é dado por:

$$\alpha_i(h(t)) := \frac{w_i(h(t))}{\sum_{j=1}^r w_j(h(t))}$$

O modelo Takagi-Sugeno fornece, finalmente:

$$\Delta\dot{h}(t) = \sum_{i=1}^4 \alpha_i(h(t))(A_i\Delta h(t) + B_i\Delta u(t)) \tag{5.10}$$

### 5.3.6 Controlador Fuzzy

No Capítulo 4 demonstra-se o projeto de controladores capazes de estabilizar o sistema em todos os pontos de 5.10. O desenvolvimento do controlador final para modelo segue os mesmos passos já apresentados: realiza-se a sintonização do ganho  $K$  a ser utilizado para cada regra e o grau de ativação de cada ganho é obtido pelas mesmas Equações 5.9. O ganho final a ser utilizado é dado por:

$$K = \frac{\sum_{i=1}^4 w_i(h(t)) K_i}{\sum_{i=1}^4 w_i(h(t))}$$

Ou seja,

$$K = \sum_{i=1}^4 \alpha_i(h(t)) K_i \quad (5.11)$$

Deseja-se neste trabalho obter um controlador capaz de prover erro nulo para ambas os níveis inferiores em estado estacionário. Desta forma, foi desenvolvido um projeto aumentado dos erros integrais das variáveis controladas. O modelo final dos sistema é apresentado na Figura 5.2 a seguir:

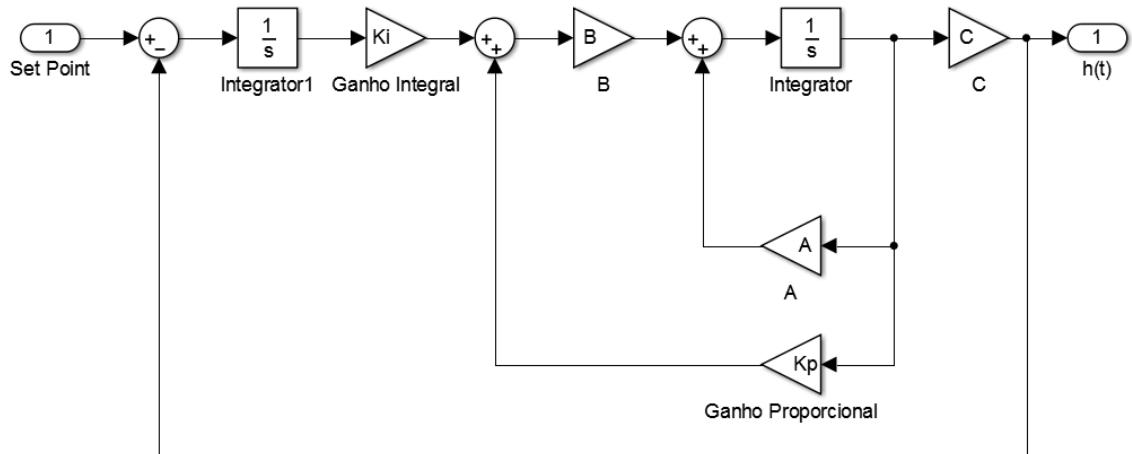


Figure 5.2: Espaço de estados da planta controlada

## 6 IMPLEMENTAÇÃO

Uma vez que se tenha modelado o sistema, é necessário realizar a identificação dos parâmetros da planta real afim de aplicar os mesmos passos para a sintonia do controlador. A implementação em bancada inicia-se com a calibração dos sensores e identificação dos ganhos das bombas e do sistema em 4 pontos, que irão compor os conjuntos das regras TS. Por fim, os ganhos são calculados desenvolvendo-se as matrizes de cada uma das regras e seguindo as equações 4.4.

### 6.1 IDENTIFICAÇÃO

Há diversas formas de realizar a identificação de um sistema. Baseando-se na forma e quantidade de informações disponíveis, existem três tipos de identificação:

- **Caixa-Branca:** ocorre quando é realizada aferindo-se diretamente todos os parâmetros envolvidos.
- **Caixa-Preta:** baseia-se completamente nos dados externos do sistema como seus resultados para algum teste para identificação desejada.
- **Caixa-Cinza:** é realizada a partir de informações internas aferidas diretamente do sistema utilizando resultados externos para determinação dos parâmetros ainda não identificados.

Neste trabalho utilizou-se inicialmente o software MATLAB em uma tentativa de realizar uma identificação caixa-preta, puramente baseada nos dados obtidos a partir de degraus nas entradas. Como será visto a seguir, está abordagem não forneceu um sistema condizente com o esperado, portanto optou-se por seguir a estratégia caixa-cinza para realizar a identificação. As descrições de ambas são apresentadas a seguir.

#### 6.1.1 Identificação Caixa-Preta

A identificação caixa-preta verifica o comportamento das saídas em relação às entradas e tenta inferir, por meio de algoritmos de regressão e outros, as equações que descrevem a dinâmica do sistema. Assim, as imagens 6.1 - 6.3 a seguir apresentam as respostas do sistema aos degraus nas entradas. Esses dados foram tratados e enviados para a Toolbox de Identificação do Matlab.

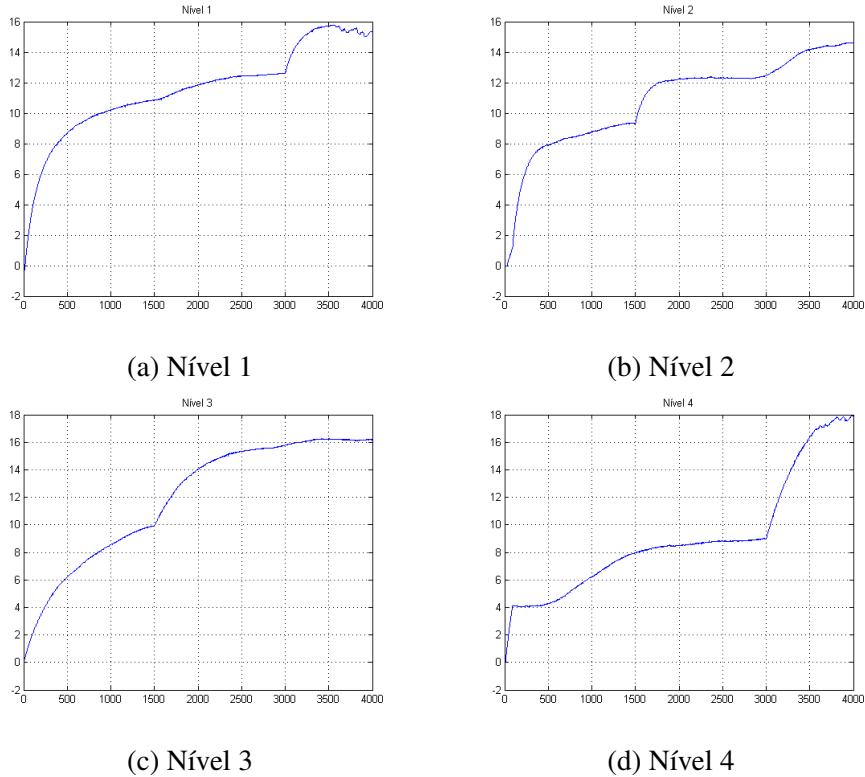


Figure 6.1: Identificação 1 - 4000 segundos

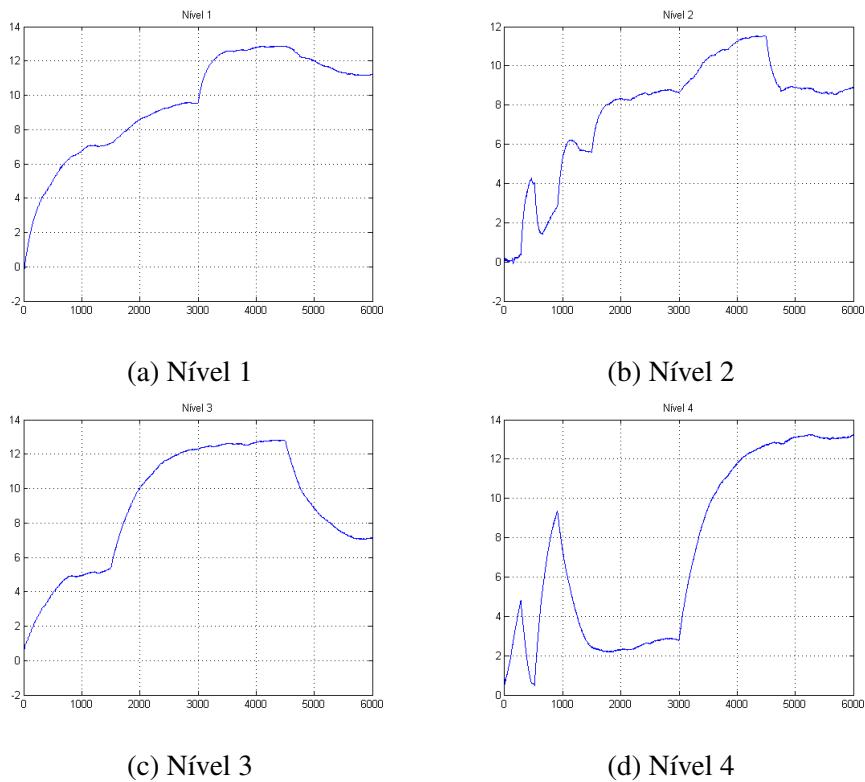


Figure 6.2: Identificação 2 - 6000 segundos

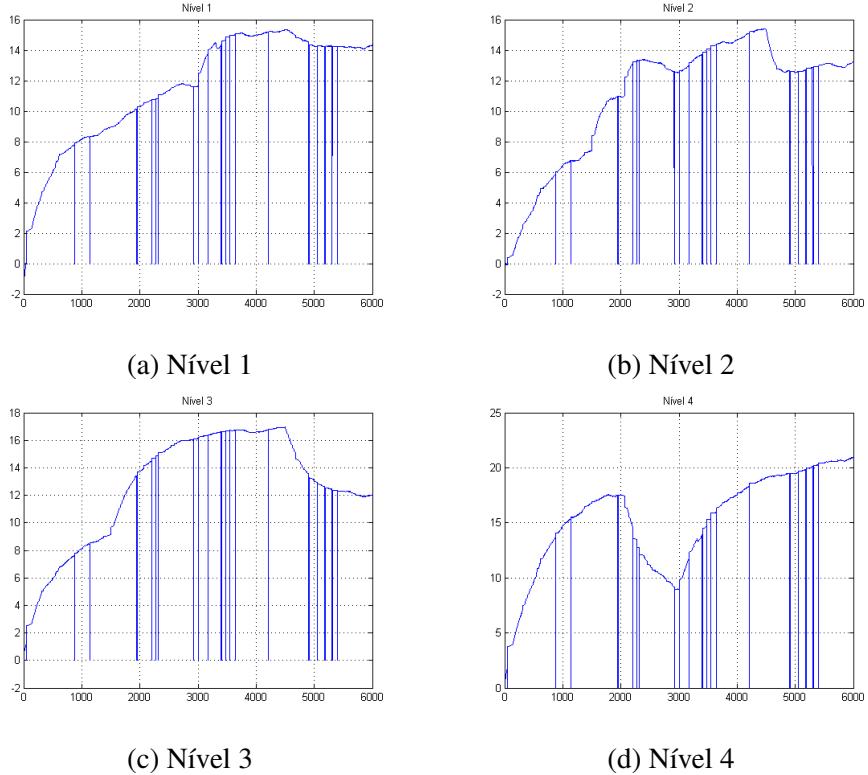


Figure 6.3: Identificação 3 - 6000 segundos

A ferramenta recebe os dados de entrada e saída dos testes do sistema para estimação que retorna as matrizes índices no espaço de estados. É possível determinar as ordens dos parâmetros para estimação e quais elementos de cada matriz são conhecidos. Neste trabalho, determinou-se:

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix} \quad B = \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \\ b_{3,1} & b_{3,2} \\ b_{4,1} & b_{4,4} \end{bmatrix} \quad (6.1)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (6.2)$$

Os parâmetros que se deseja encontrar são  $a_{i,j}$  e  $b_{i,j}$ . A matriz D é nula e C é a identidade, selecionando os quatro níveis, que compõe o estado, como as saídas. Os resultados obtidos são apresentados a seguir:

- Identificação I:

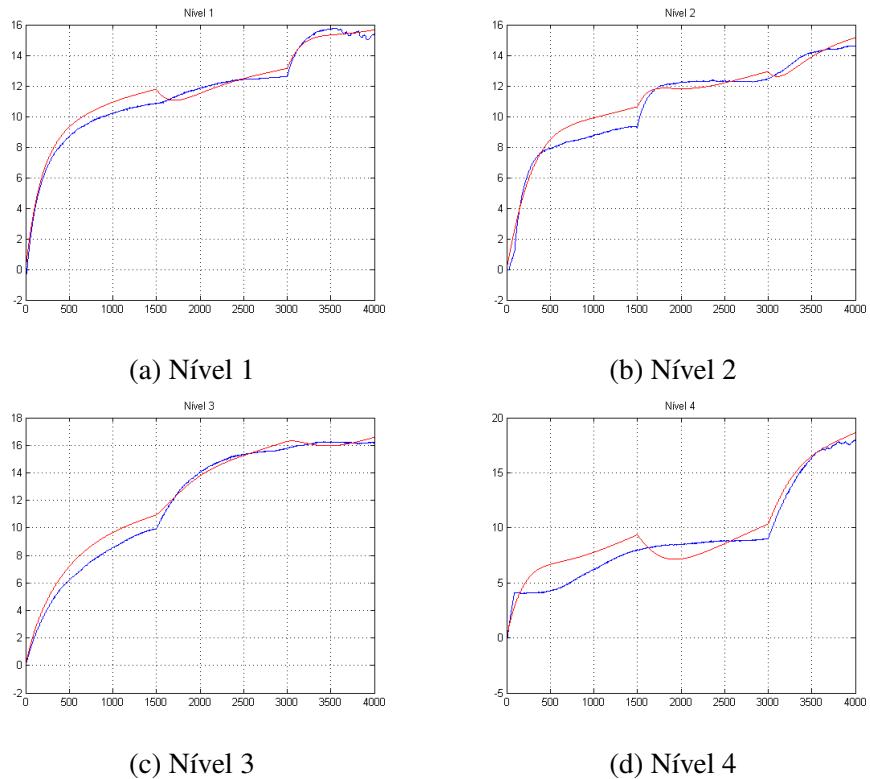


Figure 6.4: Identificação 1 - 4000 segundos

$$A = \begin{bmatrix} 0.9985 & -0.0002098 & 0.0007881 & 0.0001684 \\ 0.000169 & 0.999 & -7.159e-05 & 0.0004413 \\ -0.000736 & 0.0001313 & 1 & 0.0001939 \\ -0.0009293 & -0.0008117 & 0.00109 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0006189 & 0.0008599 \\ 0.000715 & -0.0005701 \\ 0.0001636 & -2.26e-05 \\ -0.0009055 & 0.00109 \end{bmatrix}$$

- Identificação II:

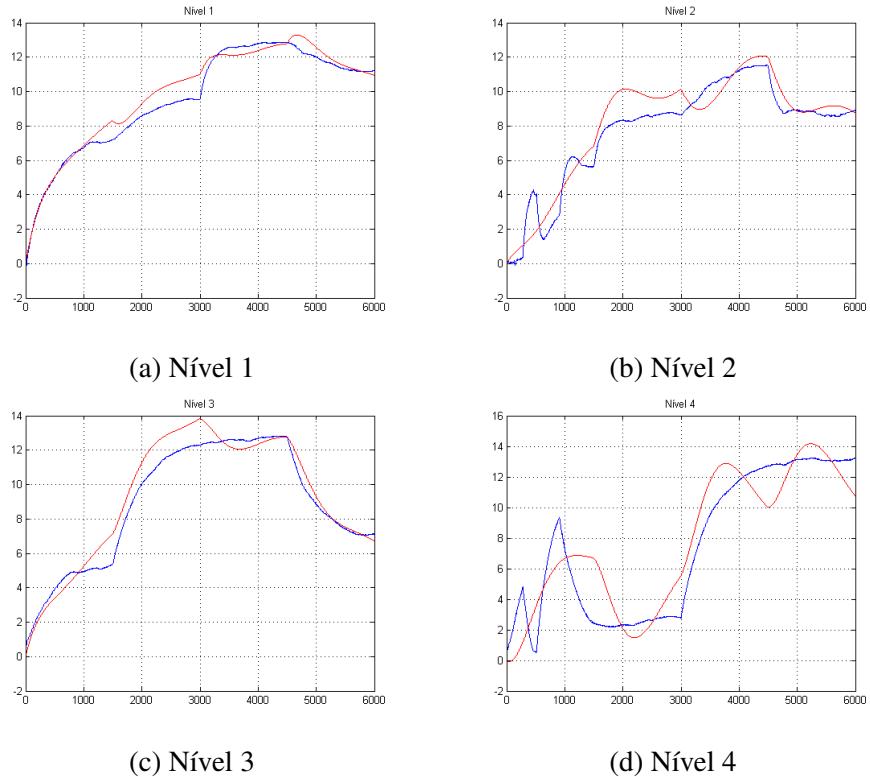


Figure 6.5: Identificação 2 - 6000 segundos

$$A = \begin{bmatrix} 0.999 & 0.0002 & 0.000456 & 3.182e-05 \\ -0.0001421 & 1 & -0.0001671 & 0.0002886 \\ -0.0005784 & 0.0003516 & 0.9999 & 0.0001039 \\ 0.0006769 & -0.001255 & 0.0006098 & 0.9999 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0004175 & 0.0005107 \\ 0.0006248 & -0.0005959 \\ 0.0002711 & -0.0002032 \\ -0.0002449 & 0.0002294 \end{bmatrix}$$

- Identificação III:

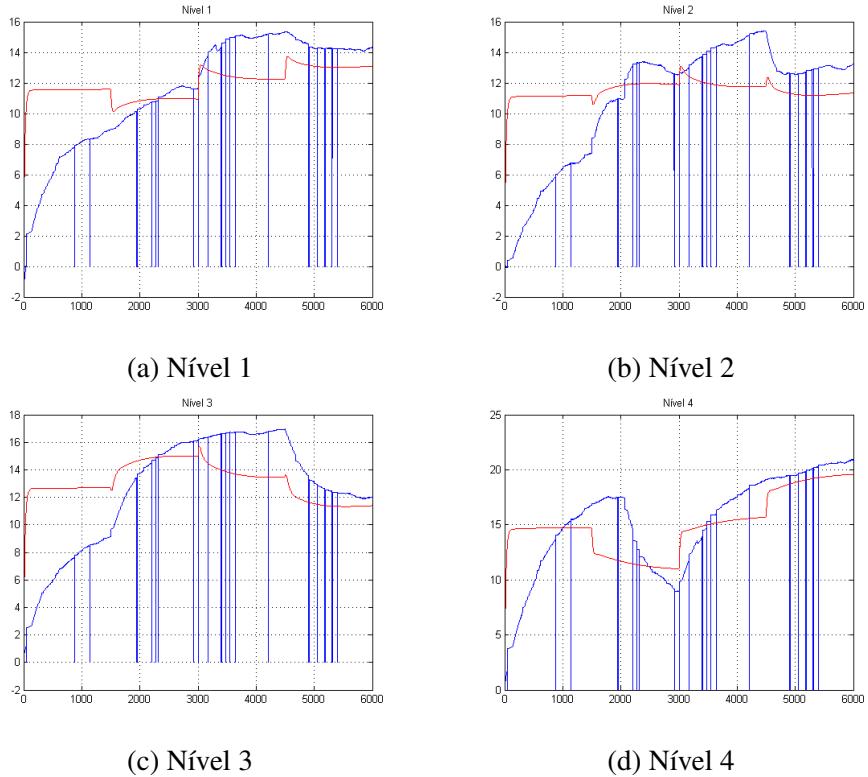


Figure 6.6: Identificação 3 - 6000 segundos

$$A = \begin{bmatrix} 0.9834 & 0.02014 & -0.007577 & -0.00362 \\ -0.0117 & 1.019 & -0.009873 & -0.003866 \\ -0.009733 & 0.02102 & 0.9863 & -0.004788 \\ -0.02114 & 0.02447 & -0.008362 & 0.9952 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.007073 & 0.009851 \\ -0.004253 & 0.006861 \\ -0.002624 & 0.005544 \\ -0.009936 & 0.01346 \end{bmatrix}$$

Embora os modelos se aproximem muito das curvas reais, nota-se que os coeficientes obtidos são absurdos, muito dispares dos esperados teoricamente. Por essa razão, ao se aplicar os métodos de obtenção de ganhos pro controlador, descritos nos capítulos anteriores, se obtém valores muito altos ou excessivamente baixos, o que torna o controle inviável.

### 6.1.2 Identificação Caixa-Cinza

Nesta abordagem obtêm-se os parâmetros aferindo diretamente do sistema ou por meio de respostas à entradas controladas e utiliza-se as equações 5.2 para definir o modelo linear identificado. A Tabela 2.1 apresenta os valores das dimensões mensuráveis do tanque e da aceleração da gravidade.

Os demais parâmetros serão identificados em 4 pontos, escolhidos por observação, que comporão as regras do modelo TS.

Escolheu-se realizar-se a identificação para o funcionamento das bombas a 42% e 45% de suas capacidades. Nota-se que abaixo de 30% não há resposta aparente e durante todo este trabalho limitou-se a atuar abaixo de 70% de sua potência máxima, uma vez que este valor foi observado como limiar seguro de utilização. Os valores de  $\gamma$  são obtidos enviando tensões constantes para cada bomba e verificando a proporção de fluído enviado para os tanques respectivos. Como o comportamento real da bomba é não linear, esses valores são identificados para cada uma das regras.

De forma semelhante, as constantes de fluxo  $k$  são identificadas aferindo-se o volume enviado aos tanques, para uma dada tensão constante num tempo fixo. A taxa volume/tempo é utilizada como fluxo médio,  $k_i$  é obtido dividindo o fluxo médio  $i$  pela tensão  $i$  utilizada.

Table 6.1: Tensões Escolhidas

Identificação	Tensão ( $v_i$ )	$\gamma_1$	$\gamma_2$	Ganho 1 ( $k_1$ )	Ganho 2 ( $k_2$ )
1	42%	0.8980	0.6810	7.4044 $cm^3/Vs$	7.1022 $cm^3/Vs$
2	45%	0.8276	0.6827	8.1801 $cm^3/Vs$	7.3339 $cm^3/Vs$

Realiza-se então a identificação do modelo, a partir dos estados estacionários, nos 4 pontos dados pela interpolação destes conjuntos. Os resultados obtidos são exibidos a seguir:

Table 6.2: Pontos de Operação da Planta Instalada

Sistema	Bomba 1 ( $\bar{v}_1$ )	Bomba 2 ( $\bar{v}_1$ )	Nível 1 ( $\bar{h}_1$ )	Nível 2 ( $\bar{h}_2$ )	Nível 3 ( $\bar{h}_3$ )	Nível 4 ( $\bar{h}_4$ )
1	42%	42%	11.1024 cm	9.4410 cm	8.7849 cm	17.4466 cm
2	42%	45%	16.3767 cm	16.1350 cm	11.4907 cm	20.5671 cm
3	45%	42%	15.0037 cm	13.9138 cm	20.7899 cm	19.5759 cm
4	45%	45%	18.0278 cm	18.8761 cm	18.3048 cm	19.6136 cm

São obtidas então 4 regras e para cada uma delas um controlador, sintonizado seguindo as equações 4.4. A tabela a seguir apresenta os ganhos obtidos:

Table 6.3: Ganhos aplicados à planta

Regra	Ganho						
	$K =$						
1		-10.1267	-0.1726	-0.0693	-0.0304	7.4632	0.0782
		0.4704	-8.9962	0.0234	-0.0081	-0.1735	11.3821
2	$K =$	-10.1655	-0.1445	-0.0628	-0.0301	7.4634	0.0680
		0.3944	-8.7470	0.0221	-0.0071	-0.1460	11.0130
3	$K =$	-9.7178	-0.2288	-0.0433	-0.0878	7.1828	0.0979
		0.5970	-8.9866	-0.0014	-0.0096	-0.2192	11.3719
4	$K =$	-9.7327	-0.2180	-0.0472	-0.0874	7.1831	0.0939
		0.5519	-8.7291	0.0082	-0.0090	-0.2028	11.0068

## 6.2 CLP

Os ganhos obtidos a partir da identificação do modelo foram implementados no CLP utilizando um diagrama de bloco de funções e uma rotina em texto estruturado. A Figura 6.7 a seguir apresenta o programa principal.

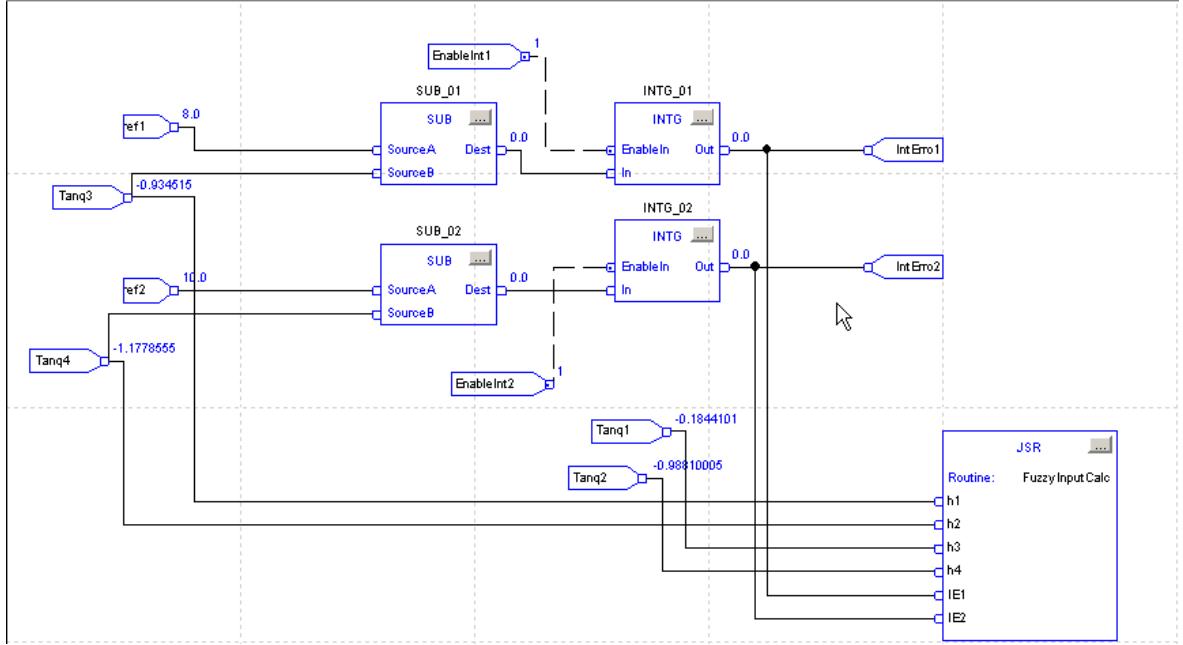


Figure 6.7: Programa Principal

O controlador recebe os estados atuais do sistema e os erros acumulados das variáveis controladas. A rotina textitFuzzyInputCalc implementa o controlador fuzzy a partir destas entradas e dos ganhos já apresentados assinalando às variáveis de saída o valor da tensão nas bombas desejado. A Figura 6.8 a seguir apresenta as variáveis definidas no trabalho.

Name	Value	Force Mask	Style	Data Type
Bomba1	0.0	Float	REAL	
Bomba2	0.0	Float	REAL	
Tang4	-1.1778555	Float	REAL	
Tang3	-0.934515	Float	REAL	
Tang2	-0.98810005	Float	REAL	
Tang1	-0.1844101	Float	REAL	
Saida2	0.0	Float	REAL	
Saida1	0.0	Float	REAL	
+ row	0	Decimal	INT	
ref2	10.0	Float	REAL	
ref1	8.0	Float	REAL	
+ PID2	{...}	{...}	PID	
+ PID	{...}	{...}	PID	
+ k4	{...}	{...}	Float	REAL[2,6]
+ k3	{...}	{...}	Float	REAL[2,6]
+ k2	{...}	{...}	Float	REAL[2,6]
+ k1	{...}	{...}	Float	REAL[2,6]
+ K	{...}	{...}	Float	REAL[2,6]
IntErro2	0.0	Float	REAL	
IntErro1	0.0	Float	REAL	
In_out2	0.0	Float	REAL	
In_out1	0.0	Float	REAL	
h4	0.0	Float	REAL	
h3	0.0	Float	REAL	
+ col	0	Decimal	INT	
alpha4	0.0	Float	REAL	
alpha3	0.0	Float	REAL	
alpha2	0.0	Float	REAL	
alpha1	0.0	Float	REAL	

Figure 6.8: Variáveis do programa

Os resultados obtidos são apresentados na seção 7.2 a seguir.

## 7 RESULTADOS

O destino é inexorável

---

Bernard Cornwell

O modelo fuzzy desenvolvido a partir das seções anteriores foi simulado utilizando o software MATLAB e implementado na bancada real via CLP Rockwell. As seções a seguir apresentam os resultados obtidos em cada caso.

### 7.1 SIMULAÇÕES

A planta de quatro-tanques, como apresentada no Capítulo 2, compõe um sistema capaz de ilustrar diversas dinâmicas para suas variáveis de processo. Assim, são escolhidas uma configuração em **fase mínima** e outra em **fase não-mínima** e a partir delas a modelagem e o controlador fuzzy são desenvolvidos.

#### 7.1.1 Fase Mínima

Nesta configuração a maior parte do fluído que saí das bombas é direcionado diretamente para os tanques controlados, ou seja  $\gamma_i > 0.5$ . A Tabela 7.1 a seguir apresenta suas especificações.

Table 7.1: Parâmetros da planta em fase mínima.

Especificações Iniciais da Planta	
A1, A3 ( $cm^2$ )	28
A2, A4 ( $cm^2$ )	32
a1, a3 ( $cm^2$ )	0.071
a2, a4 ( $cm^2$ )	0.057
g $cm/s$	981
k1	$3,33 \text{ cm}^3/Vs$
k2	$3.35 \text{ cm}^3/Vs$
$\gamma_1$	0.60
$\gamma_2$	0.70

O modelo não linear 5.1 para esta configuração é:

$$\begin{cases} \dot{h}_1 = \frac{1}{28}(0.071\sqrt{1962 * h_3} + 1,998 * v_1 - 0,071\sqrt{1962 * h_1}) \\ \dot{h}_2 = \frac{1}{32}(0,057\sqrt{1962 * h_4} + 2,345 * v_2 - 0,057\sqrt{1962 * h_2}) \\ \dot{h}_3 = \frac{1}{28}(1,005 * v_2 - 0,071\sqrt{1962 * h_3}) \\ \dot{h}_4 = \frac{1}{32}(1,332 * v_1 - 0,057\sqrt{1962 * h_4}) \end{cases} \quad (7.1)$$

Escolhendo os conjuntos fuzzy {"baixo", "alto"} e definindo {5, 15} como seus representantes os

níveis 1 e 2, por combinação simples obtém-se os seguintes pontos de linearização:

Table 7.2: Pontos de Operação

Sistema	$\bar{h}_1$ (cm)	$\bar{h}_2$ (cm)	$\bar{h}_3$ (cm)	$\bar{h}_4$ (cm)	$\bar{v}_1$ (v)	$\bar{v}_2$ (v)
1	5	5	0.0334	2.9076	3.2321	0.5716
2	5	15	0.9431	1.1033	1.9910	3.0390
3	15	5	0.2229	13.0191	6.8393	-1.4773
4	15	15	0.1001	8.7228	5.5982	0.9900

Haverá então quatro regras Se-Então para composição do modelo TS final. As imagens a seguir apresentam a comparação entre os modelos não-linear 5.1, linearizado 5.2 em um ponto único e Takagi-Sugeno 5.10:

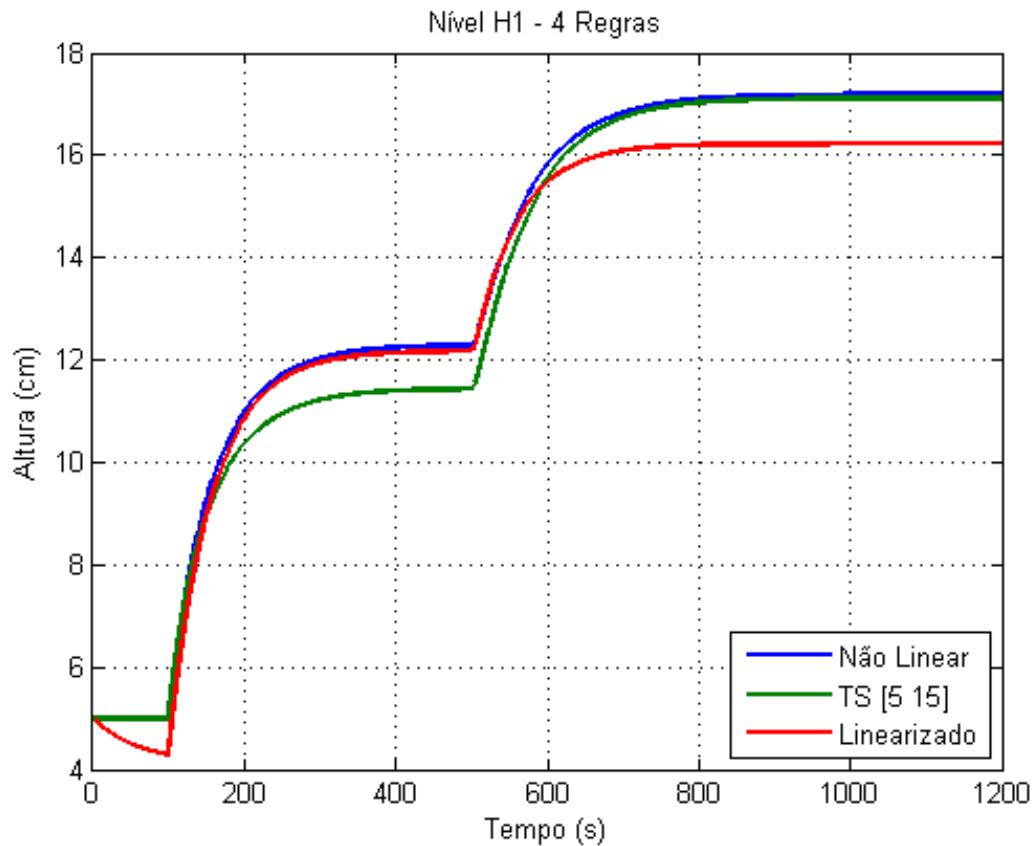


Figure 7.1: Nível do Tanque 1

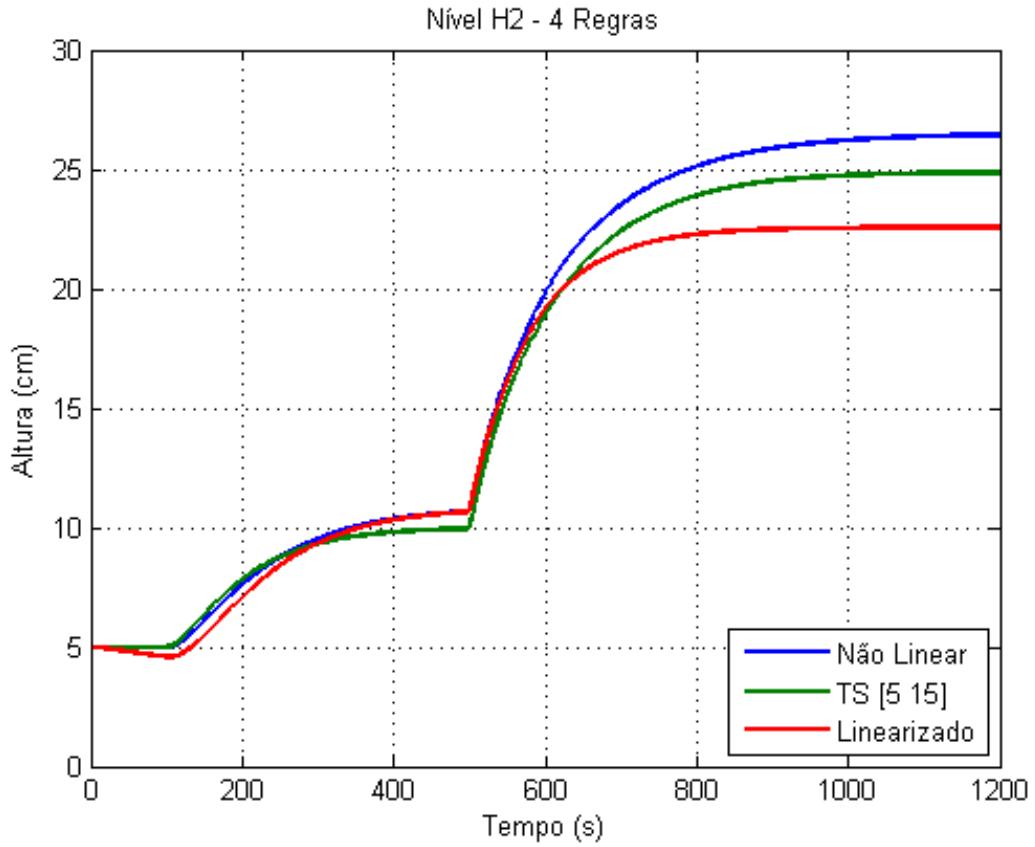
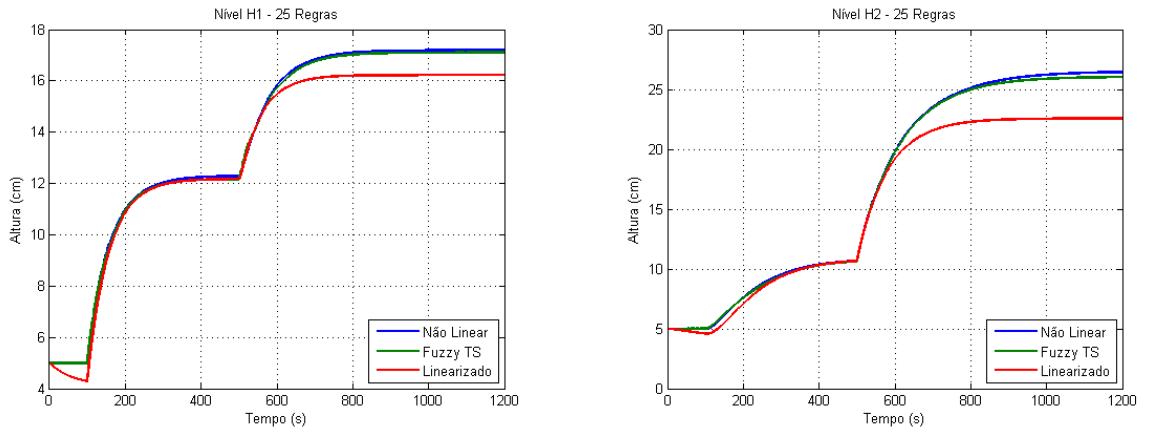


Figure 7.2: Nível do Tanque 2

É notável que o modelo fuzzy representa de modo mais eficiente o sistema. Como dito, o modelo TS pode se aproximar o quanto se desejar do não-linear no qual se baseia. A Imagem 7.3 apresenta um modelo com 5 conjuntos para os dois níveis e a Imagem 7.4 utilizando 15. É importante notar, no entanto, que a complexidade do modelo é exponencial, devido a combinação dos conjuntos das variáveis linguísticas presentes, assim, o primeiro é composto por 25 ( $5^2$ ) regras Se-Então e o segundo por 225 ( $15^2$ ).



(a) Nível 1

(b) Nível 2

Figure 7.3: Comparação com 25 regras

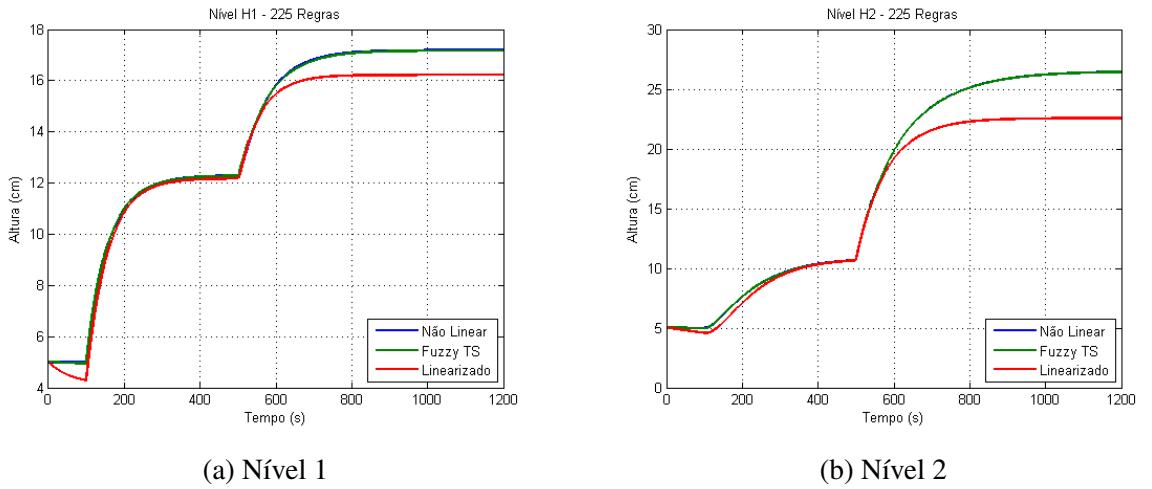


Figure 7.4: Comparação com 225 regras

A partir das Equações 4.4 são desenvolvidos os controladores para cada uma das regras. A tabela a seguir apresenta os ganhos obtidos:

Table 7.3: Ganhos Obtidos

Regra	Ganho					
1	$K = \begin{bmatrix} -13.1962 & 3.0637 & -3.0992 & 0.1430 & 15.0539 & -1.3580 \\ -5.4607 & -15.2912 & 3.4223 & 0.0239 & 1.6964 & 15.9282 \end{bmatrix}$					
2	$K = \begin{bmatrix} -12.8885 & 1.0745 & -1.5395 & -0.0323 & 14.9563 & -0.5608 \\ -1.7706 & -13.2431 & 1.0214 & -0.0123 & 0.4755 & 15.4671 \end{bmatrix}$					
3	$K = \begin{bmatrix} -13.1962 & 3.0637 & -3.0992 & 0.1430 & 15.0539 & -1.3580 \\ -5.4607 & -15.2912 & 3.4223 & 0.0239 & 1.6964 & 15.9282 \end{bmatrix}$					
4	$K = \begin{bmatrix} -12.8885 & 1.0745 & -1.5395 & -0.0323 & 14.9563 & -0.5608 \\ -1.7706 & -13.2431 & 1.0214 & -0.0123 & 0.4755 & 15.4671 \end{bmatrix}$					

Os ganhos são sintonizados para o sistema na forma:

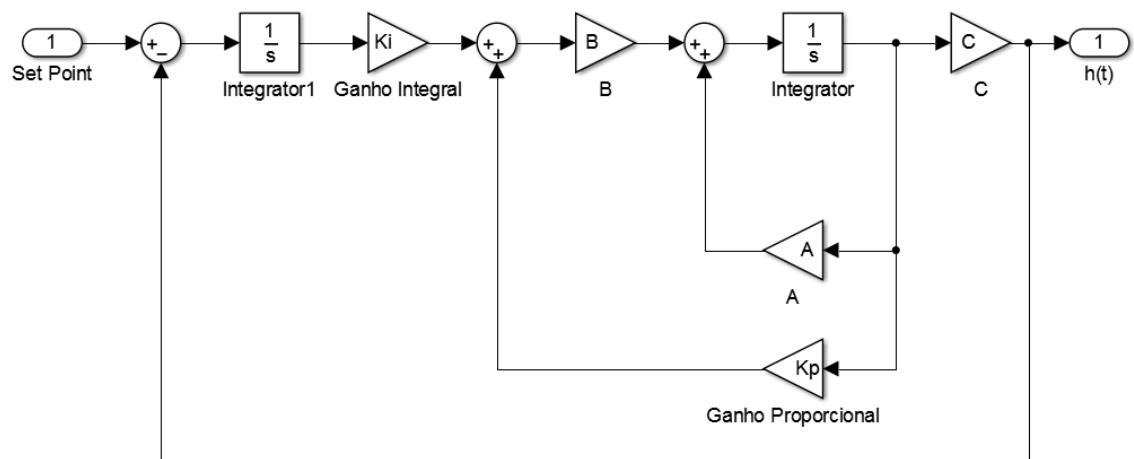


Figure 7.5: Espaço de estados da planta controlada

Os níveis controlados podem ser observados nas imagens a seguir:

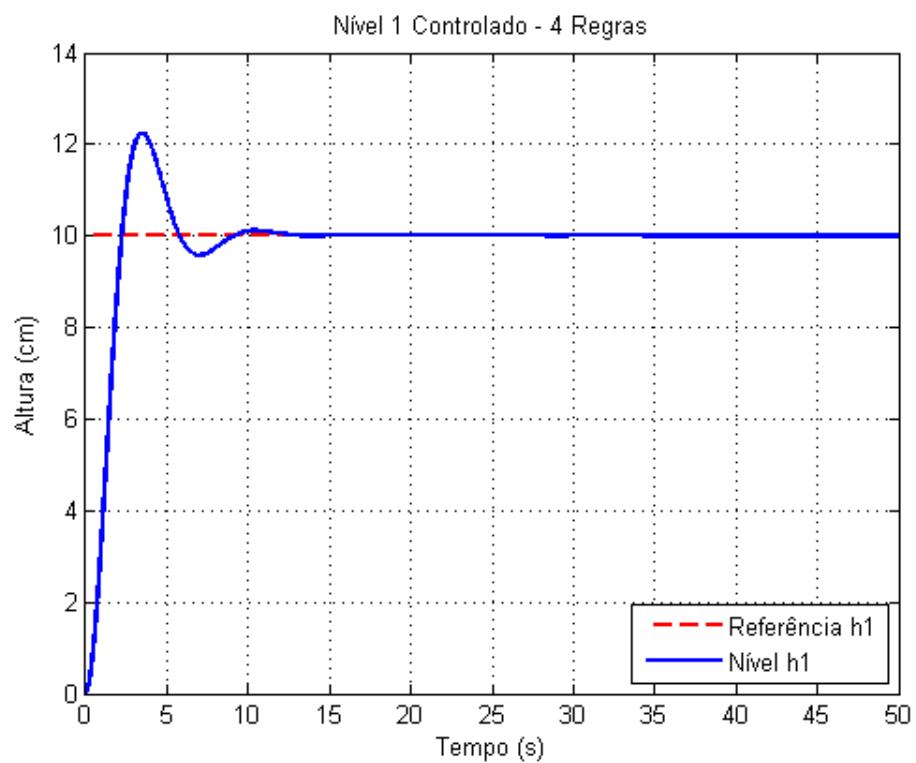


Figure 7.6: Nível H1 Controlado

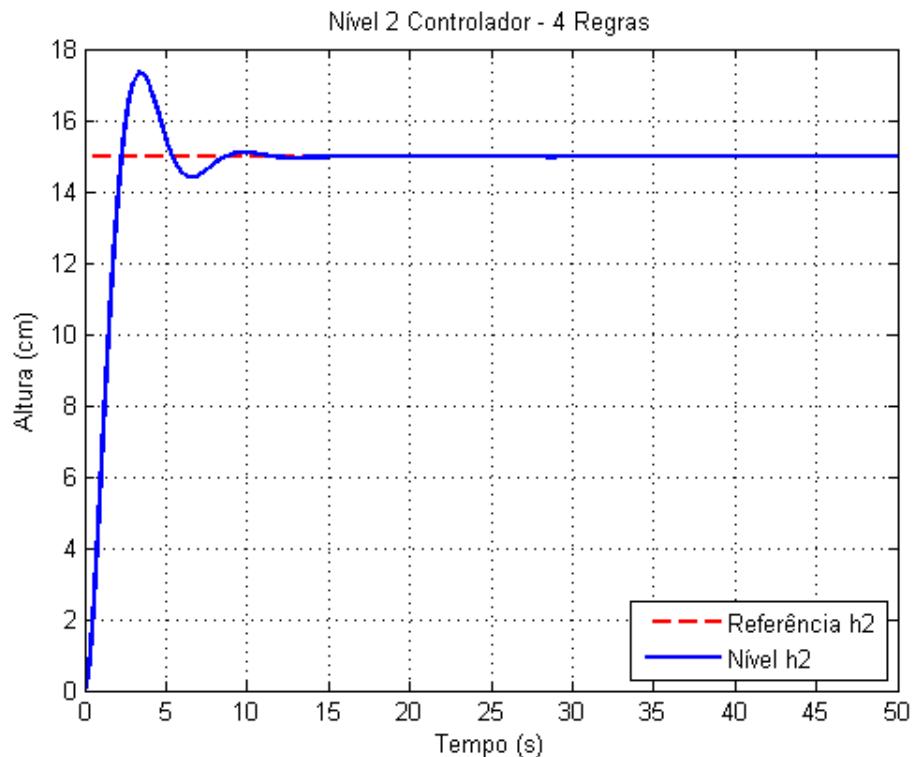


Figure 7.7: Nível H2 Controlado

No entanto, é importante notar que o controle desenvolvido até aqui não leva em consideração os limites(5 V) dos atuadores (bomba). Incluindo-a ao modelo, tem-se:

Para os sistemas das imagens 7.3 e 7.4 são sintonizados de forma similar 15 e 225 ganhos e aplicados ao sistema. As imagens a seguir ilustram os resultados obtidos:

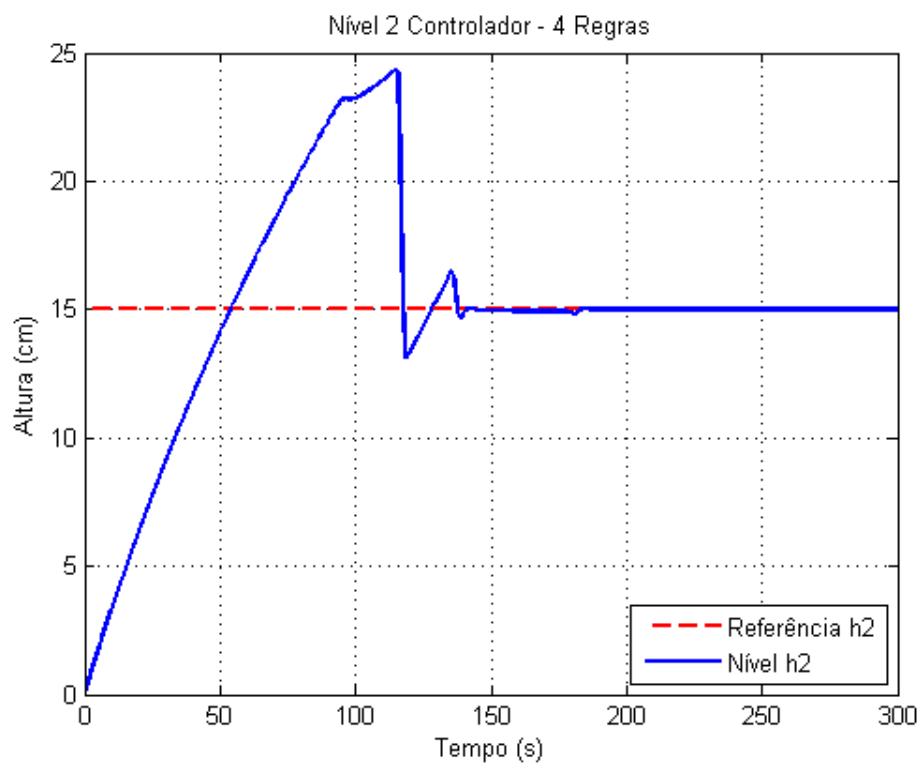


Figure 7.8: Nível H1 Controlado - Com saturação do Controlador

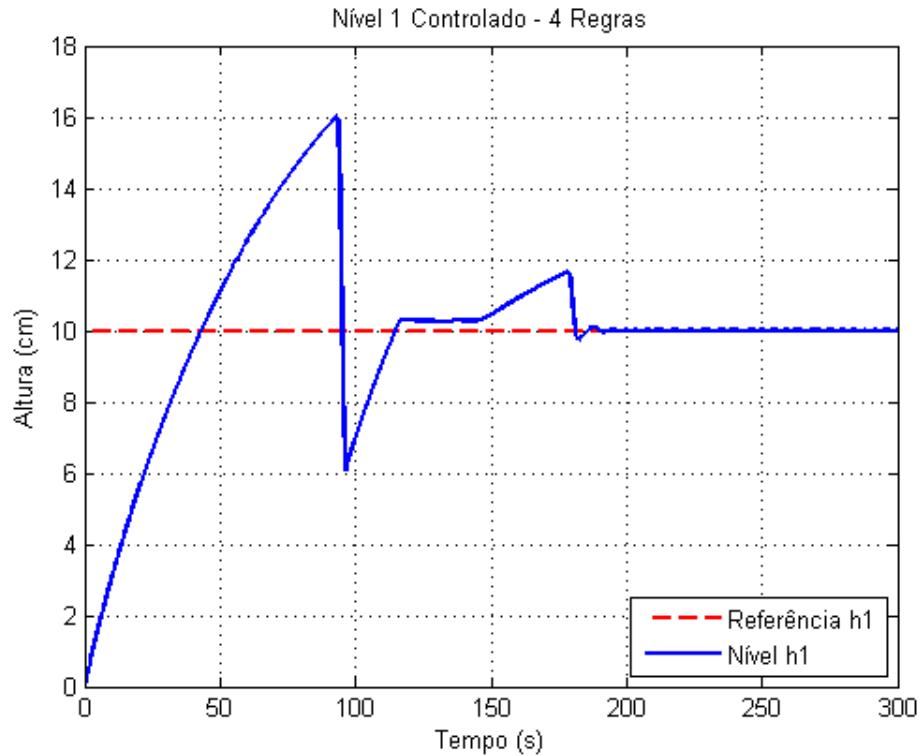


Figure 7.9: Nível H2 Controlado - Com saturação do Controlador

Para aliviar a ultrapassagem, dada pelo efeito *windup*, utiliza-se a saturação simples dos inte-

gradores nos momentos em que as variáveis manipuladas alcançam seus limites de atuação. Os gráficos a seguir ilustram:

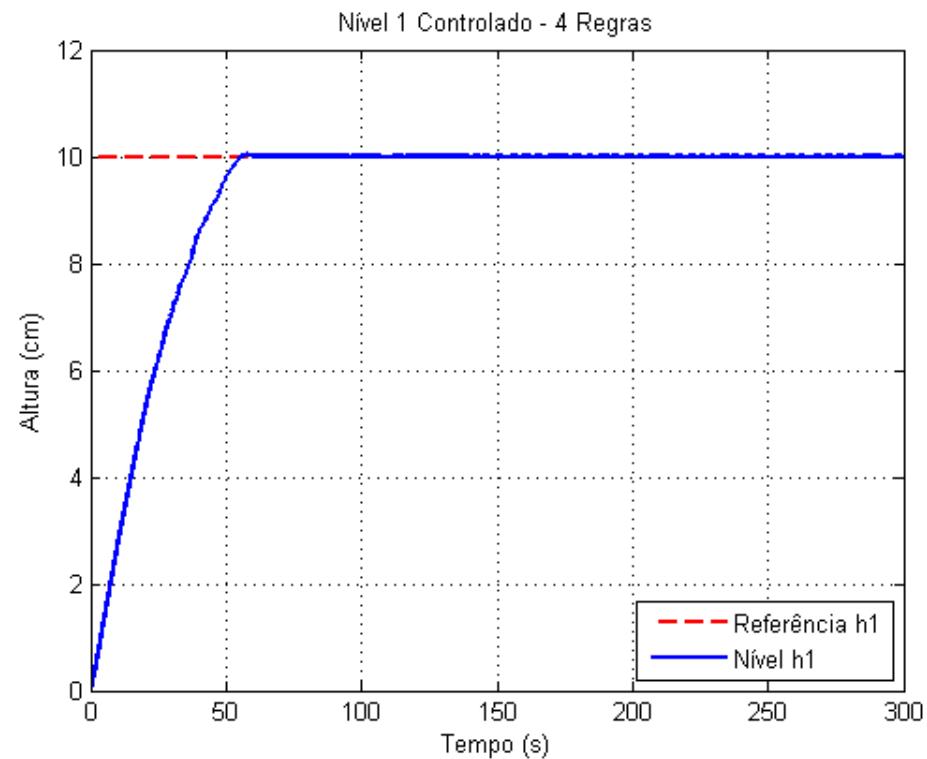


Figure 7.10: Nível H1 Controlado - Com *Anti-Windup*

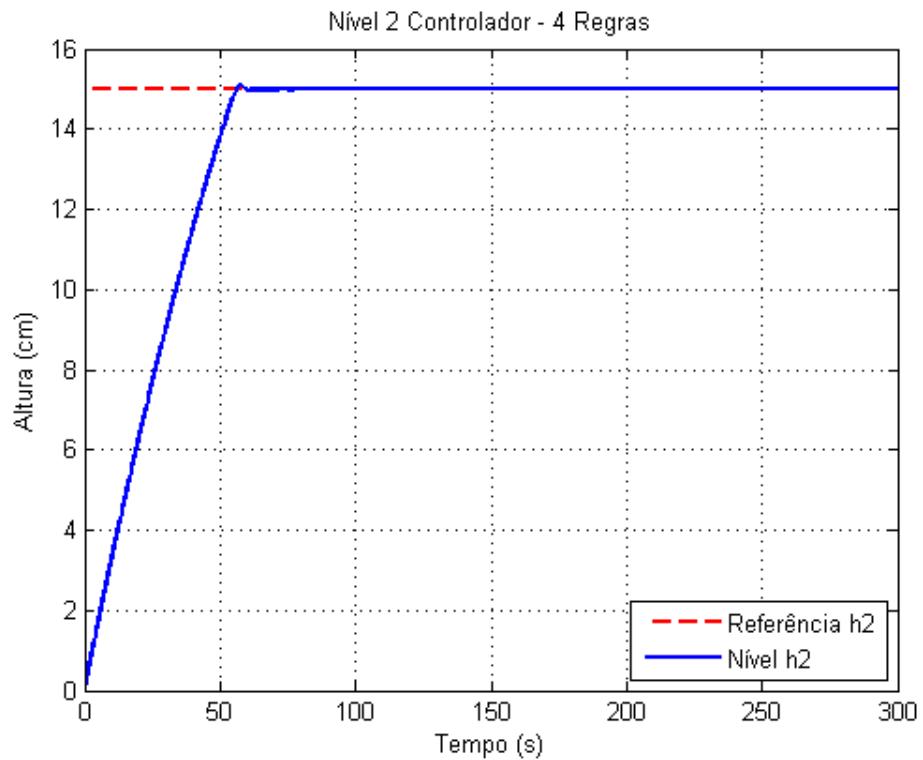


Figure 7.11: Nível H2 Controlado - Com Anti-Windup

### 7.1.2 Fase Não-Mínima

Ao contrário da configuração anterior, nesta a maior parte do fluído que saí das bombas é direcionado para os tanques superiores, ou seja  $\gamma_i < 0.5$ . A Tabela 7.4 a seguir apresenta as especificações da planta simulada.

Table 7.4: Parâmetros da planta em fase não-mínima

Especificações Iniciais da Planta	
A1, A3 ( $cm^2$ )	28
A2, A4 ( $cm^2$ )	32
a1, a3 ( $cm^2$ )	0.071
a2, a4 ( $cm^2$ )	0.057
g $cm/s$	981
k1	$3,15 \text{ cm}^3/Vs$
k2	$3.29 \text{ cm}^3/Vs$
$\gamma_1$	0.43
$\gamma_2$	0.34

O modelo não linear 5.1 para esta configuração é dado por:

$$\begin{cases} \dot{h}_1 = \frac{1}{28}(0.071\sqrt{1962 * h_3} + 1,3545 * v_1 - 0,071\sqrt{1962 * h_1}) \\ \dot{h}_2 = \frac{1}{32}(0.057\sqrt{1962 * h_4} + 1,1186 * v_2 - 0,057\sqrt{1962 * h_2}) \\ \dot{h}_3 = \frac{1}{28}(2,1714 * v_2 - 0,071\sqrt{1962 * h_3}) \\ \dot{h}_4 = \frac{1}{32}(1,7955 * v_1 - 0,057\sqrt{1962 * h_4}) \end{cases} \quad (7.2)$$

De forma similar, escolhendo os conjuntos fuzzy {"baixo", "alto"} e definindo {5, 15} como seus representantes os níveis 1 e 2, tem-se os seguintes pontos de linearização:

Table 7.5: Pontos de operação

Sistema	$\bar{h}_1$ (cm)	$\bar{h}_2$ (cm)	$\bar{h}_3$ (cm)	$\bar{h}_4$ (cm)	$\bar{v}_1$ (v)	$\bar{v}_2$ (v)
1	5	5	2.0804	1.7175	1.8428	2.0890
2	5	15	0.0321	15.9038	5.6078	-0.2595
3	15	5	16.9727	0.1661	-0.5730	5.9668
4	15	15	6.2413	5.1525	3.1919	3.6183

As imagens a seguir apresenta a comparação entre os modelos não-linear 5.1, linearizado 5.2 em um ponto único e Takagi-Sugeno 5.10:

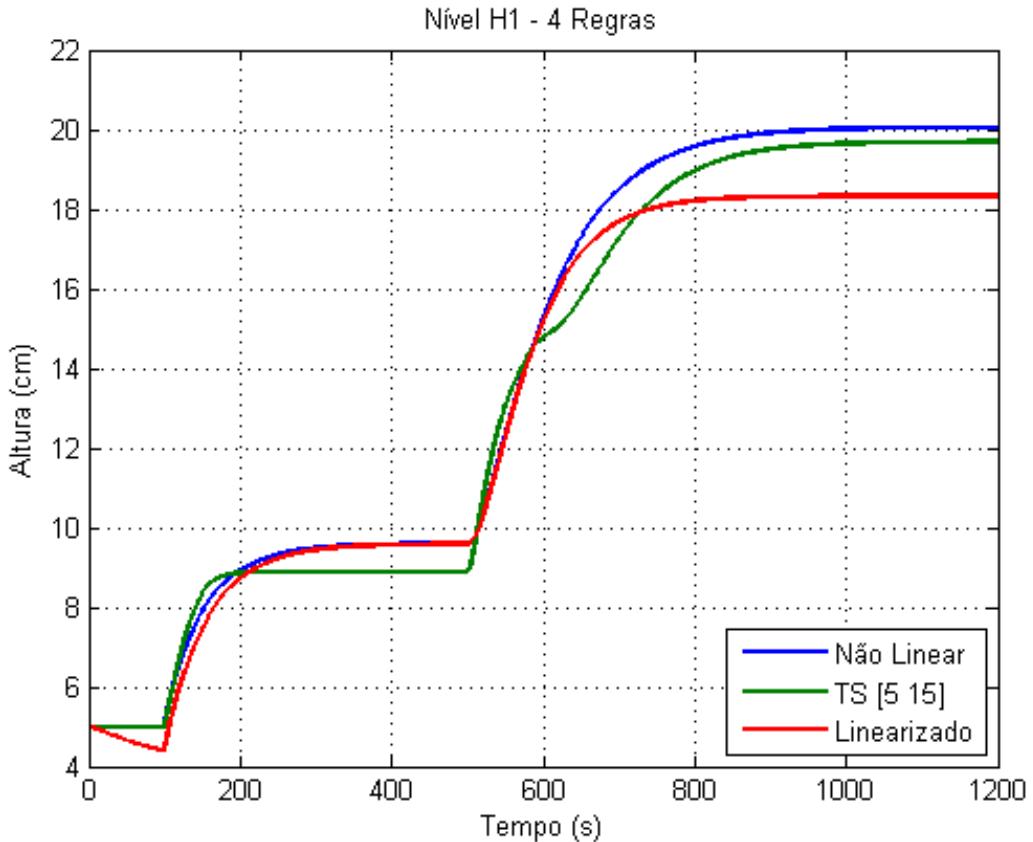


Figure 7.12: Nível do Tanque 1

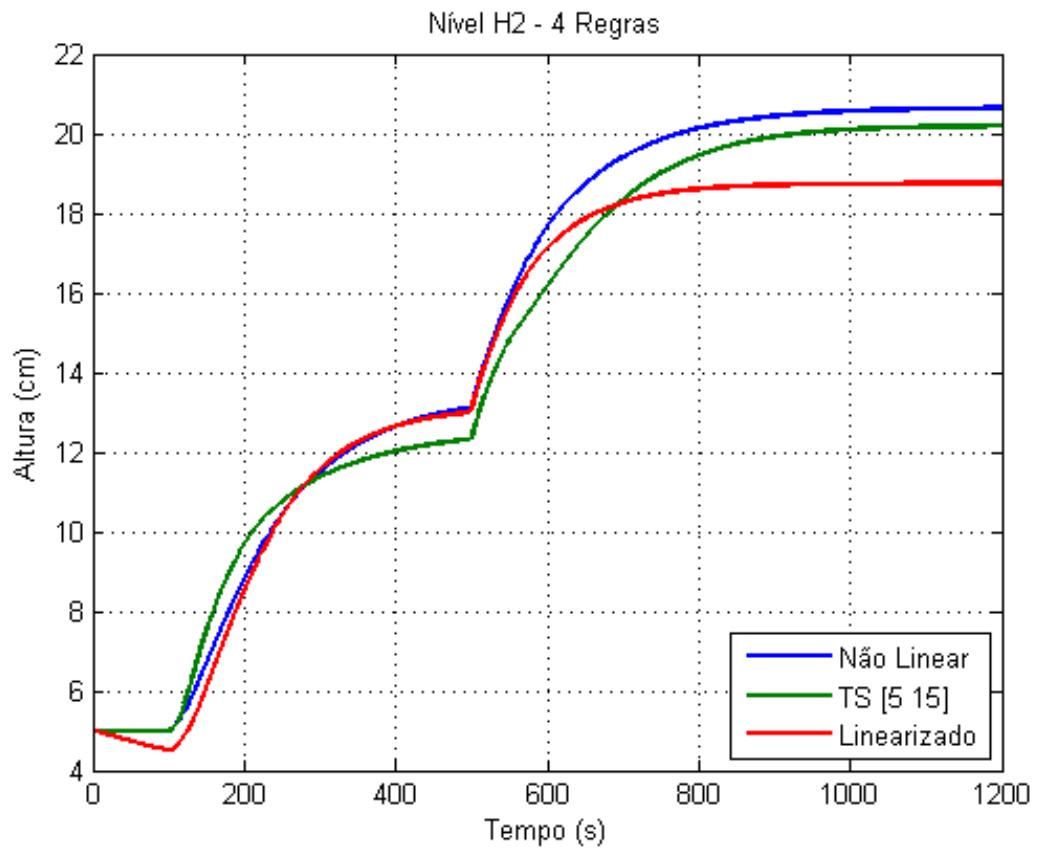


Figure 7.13: Nível do Tanque 2

Percebe-se novamente que o sistema Takagi-Sugeno apresenta resultados mais próximos do real na maior parte do tempo. Aumentando a quantidade de regras obtém-se resultados cada vez melhores. As imagens a seguir apresentam os modelos com 5 e com 15 pontos linearização

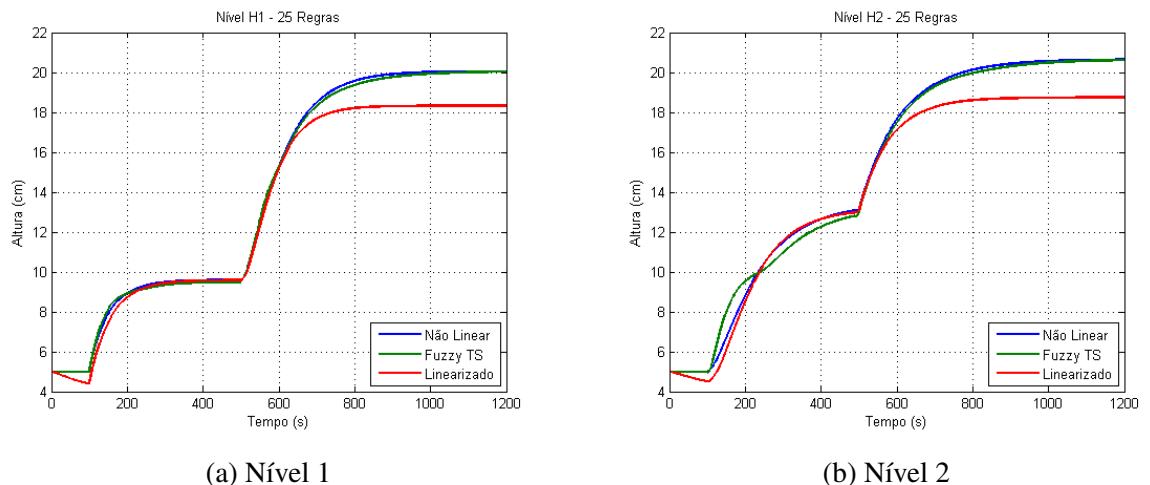


Figure 7.14: Comparação com 25 regras TS

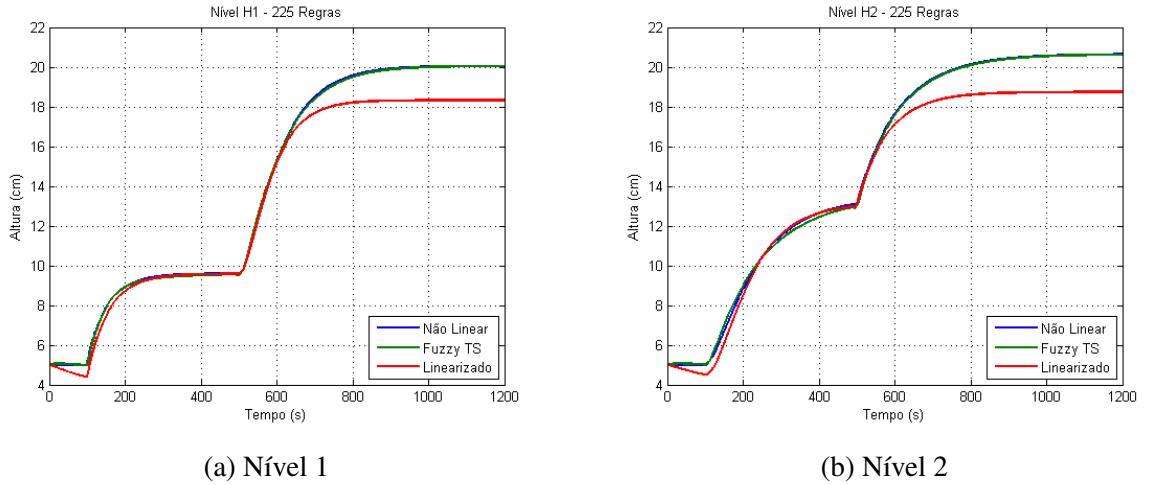


Figure 7.15: Comparação com 225 regras TS

A partir das Equações 4.4 são desenvolvidos os controladores para cada uma das regras. A tabela a seguir apresenta os ganhos obtidos:

Table 7.6: Ganhos Obtidos

Regra	Ganho
1	$K = \begin{bmatrix} 490.5249 & -805.4358 & 366.4860 & -439.5865 & 0.3923 & 19.9419 \\ -385.9505 & 484.5975 & -230.7682 & 333.2462 & 10.4116 & -4.9420 \end{bmatrix}$
2	$K = \begin{bmatrix} 0.9652 & -1.5850 & 0.7224 & -0.8567 & -0.0047 & 0.0357 \\ -0.2384 & 0.3018 & -0.1458 & 0.2055 & 0.0069 & -0.0025 \end{bmatrix}$
3	$K = \begin{bmatrix} 490.5249 & -805.4358 & 366.4860 & -439.5865 & 0.3923 & 19.9419 \\ -385.9505 & 484.5975 & -230.7682 & 333.2462 & 10.4116 & -4.9420 \end{bmatrix}$
4	$K = \begin{bmatrix} 0.9652 & -1.5850 & 0.7224 & -0.8567 & -0.0047 & 0.0357 \\ -0.2384 & 0.3018 & -0.1458 & 0.2055 & 0.0069 & -0.0025 \end{bmatrix}$

Os níveis controlados podem ser observados nas imagens a seguir:

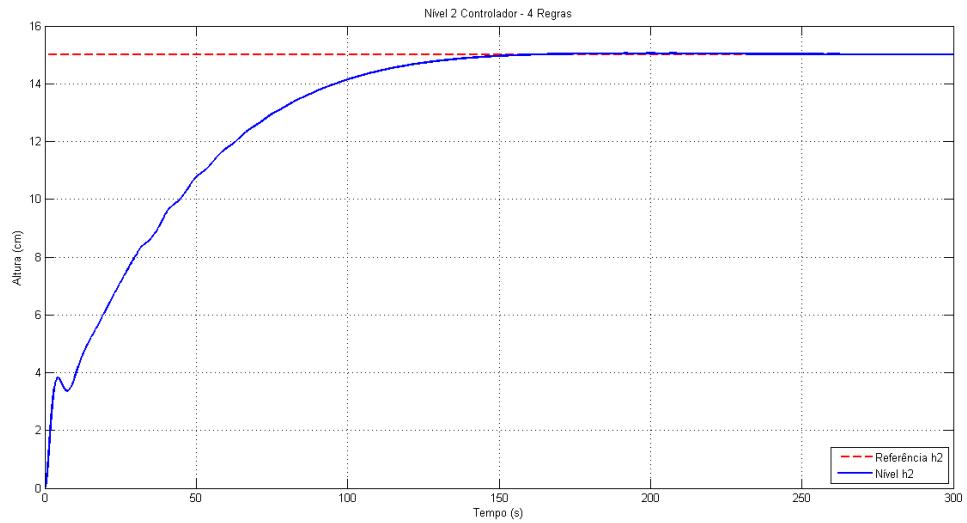


Figure 7.16: Nível H1 Controlado

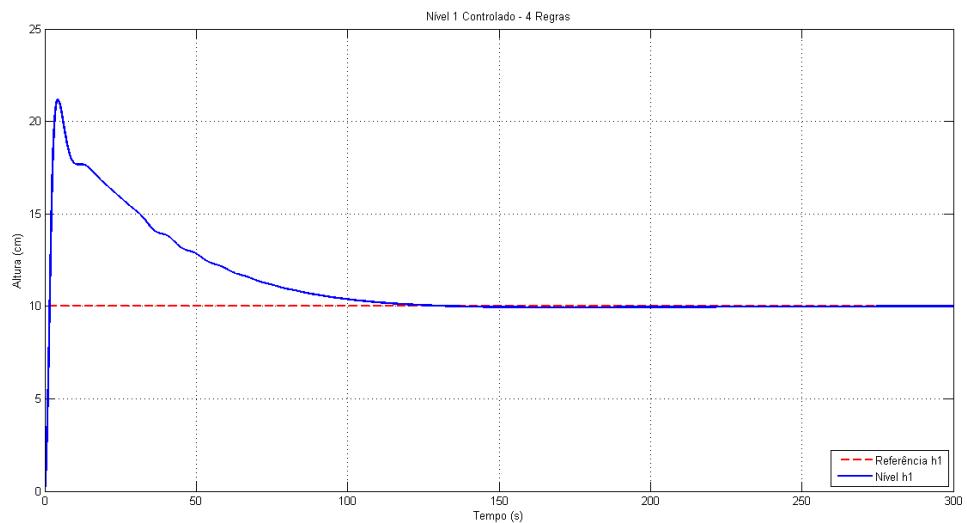


Figure 7.17: Nível H2 Controlado

No entanto, é importante notar que o controle desenvolvido até aqui não leva em consideração os limites(5 V) dos atuadores (bomba). Incluindo-a ao modelo, tem-se:

Para os sistemas das imagens 7.3 e 7.4 são sintonizados de forma similar 15 e 225 ganhos e aplicados ao sistema. As imagens a seguir ilustram os resultados obtidos:

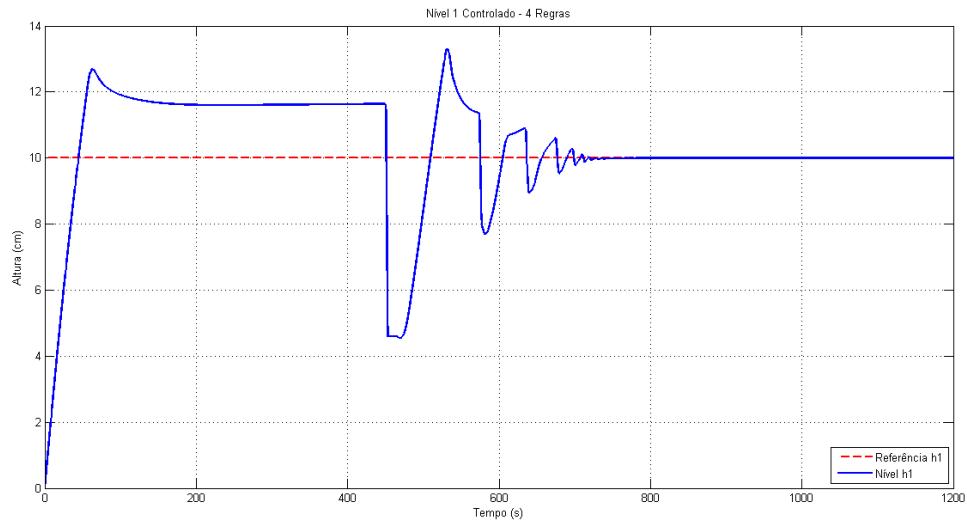


Figure 7.18: Nível H1 Controlado - Com saturação do Controlador

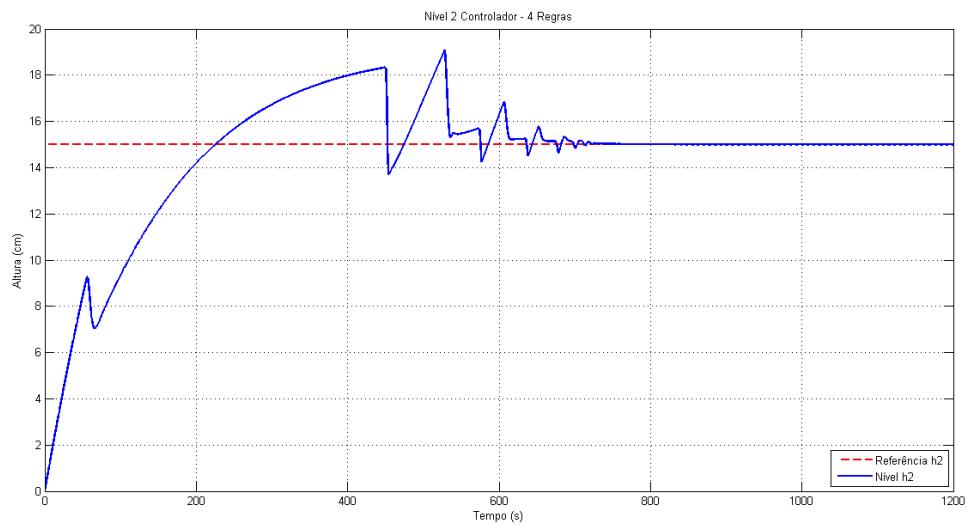


Figure 7.19: Nível H2 Controlado - Com saturação do Controlador

Para aliviar a ultrapassagem, dada pelo efeito *windup*, utiliza-se a saturação simples dos integradores nos momentos em que as variáveis manipuladas alcançam seus limites de atuação. Os gráficos a seguir ilustram:

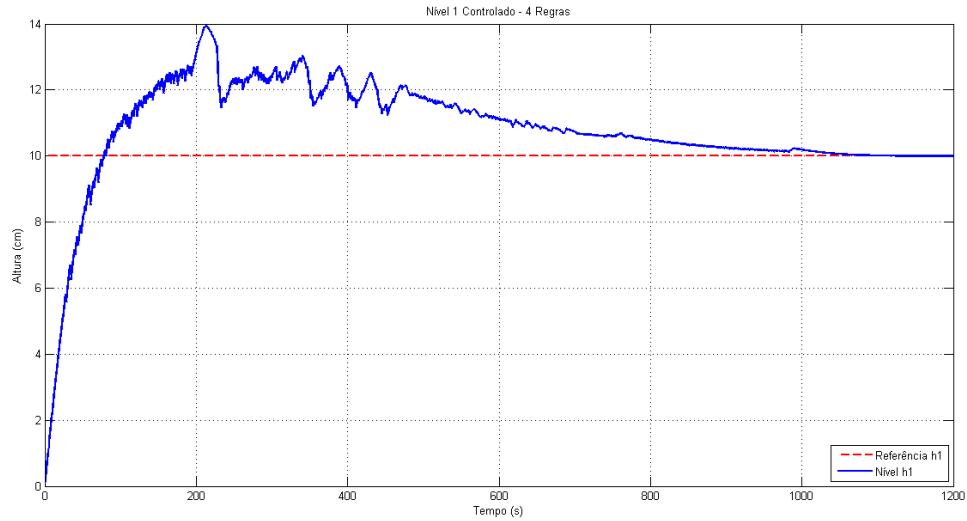


Figure 7.20: Nível H1 Controlado - Com *Anti-Windup*

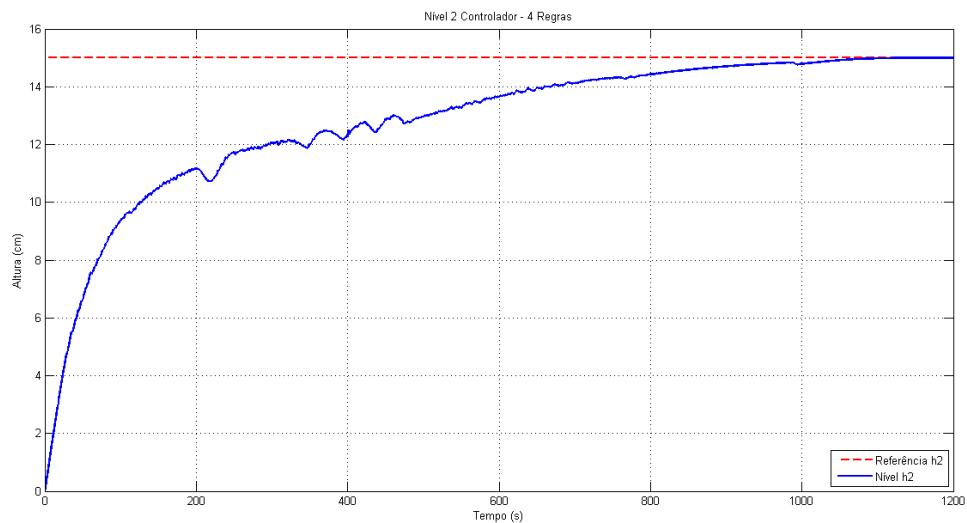


Figure 7.21: Nível H2 Controlado - Com *Anti-Windup*

## 7.2 IMPLEMENTAÇÃO NA BANCADA

Seguindo os procedimentos descritos no Capítulo 6, os ganhos obtidos na Tabela 6.3 são implementados e um degrau de referência para as alturas  $h1 = 15$  e  $h2 = 15$ , observa-se:

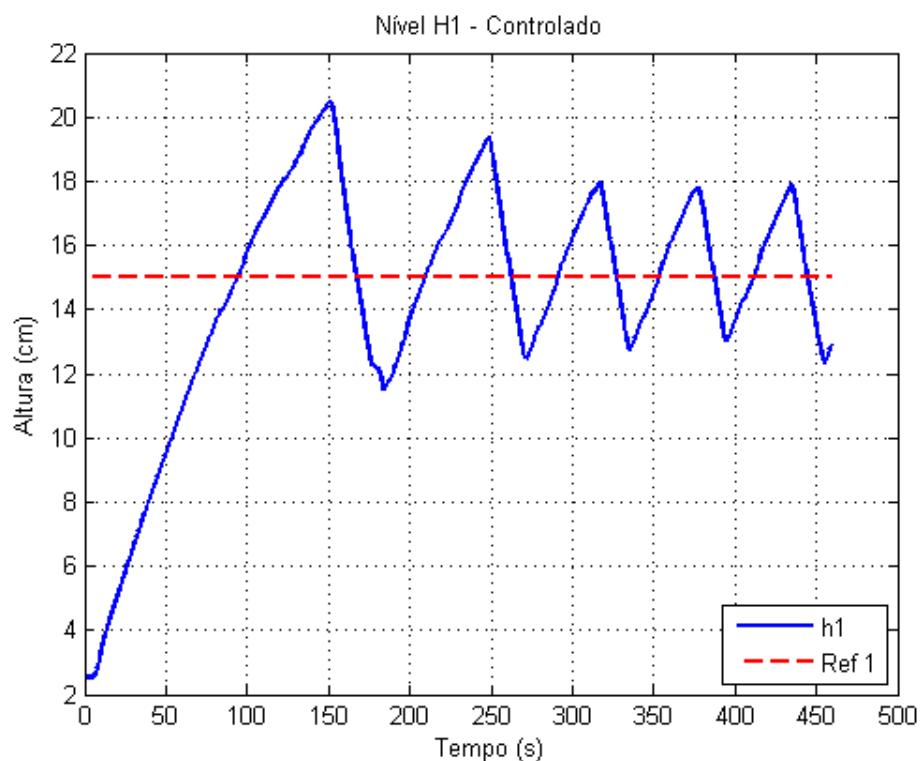


Figure 7.22: Nível H1 Controlado - Real

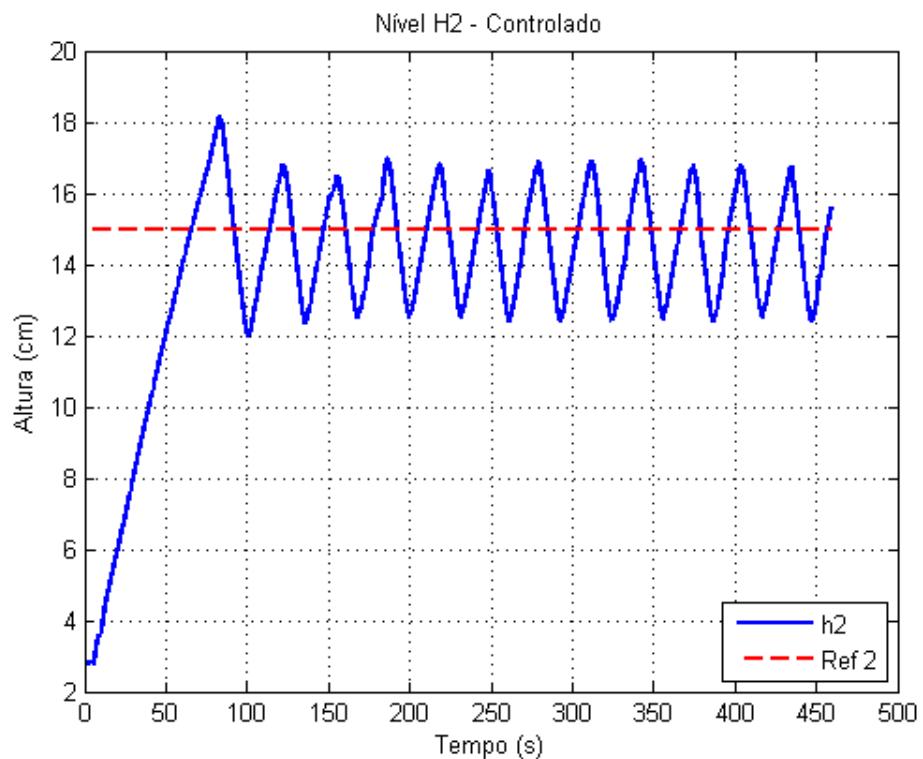


Figure 7.23: Nível H2 Controlado - Real

Nota-se que o comportamento geral do sistema tende a se estabilizar em torno da referência

desejada. A forma oscilatória da resposta, não prevista no modelo, ocorre principalmente porque apesar do sinal de controle enviado, as bombas possuem uma zona de atuação que começa a funcionar apenas para sinais superiores à 30%. Isso faz com que o erro continue aumentando até que se obtenha entradas superiores à este valor, o que gera a resposta observada.

## 8 CONCLUSÃO

Há momentos, e você chega a esses momentos, em que de repente o tempo para e acontece a eternidade.

---

Fiodor Dostoievski

Neste trabalho foi realizado o desenvolvimento de controladores fuzzy para o sistema de quatro-tanques. Iniciou-se com as modelagens não-linear e linear do sistema. Em seguida, partindo da teoria fuzzy e dos modelos Takagi-Sugeno, foram estudadas suas aplicações e o desenvolvimento sistemático de controladores via LMIs.

O CLP Rockwell foi instalado e integrado à bancada, juntamente com os módulos de entrada e saída de dados utilizados. Seguiu-se a integração aos softwares de desenvolvimento e as configurações de comunicação entre eles. Os algoritmos para controle da banca visam puramente a estabilidade em regime permanente.

Por fim, foram simulados os modelos TS, para validação em relação aos resultados esperados. Em seguida, simulou-se os controladores desenvolvidos a partir deles aplicados ao modelo não linear. Por fim, a implementação em bancada e seus resultados foram observados.

### 8.1 OBJETIVOS

O objetivo geral de desenvolver o controlador fuzzy foi obtido e testado, os resultados simulados dos modelos demonstraram comportamento eficiente e bastante confiável. A aplicação real dos métodos propostos por Takagi-Sugeno à planta de quatro tanques foi realizada e a interpolação dos múltiplos ganhos utilizando as linguagens disponíveis no CLP Rockwell instalado foi desenvolvida. Os resultados da implementação não seguiram os simulados em decorrência das faixas de operações da bomba e de suas limitações reais, que não foram acrescentadas ao modelo para manter a simplicidade do projeto.

### 8.2 TRABALHOS FUTUROS

Para trabalhos futuros indica-se a inclusão de critérios de desempenho específicos aos controladores como ultrapassagem percentual e tempo de assentamento. Além disso, a utilização de restrições de entrada na sintonia realizando um controle anti-windup mais eficaz. Como visto, o modelo TS pode aproximar-se tanto quanto se queira do real, assim, novas estratégias de identificação do sistema, mais claras em vários pontos, tornariam o projeto do controlador mais preciso. Além disso, deveria-se incluir a zona morta da bomba no modelo para delegar ao controlador uma estratégia que elimine as oscilações observadas nos resultados obtidos, gerando assim um controle mais eficaz.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] L. A. Zadeh, “Fuzzy sets ans system,” 1965.
- [2] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 116 – 132, 1985.
- [3] K. H. Johansson, A. Horch, O. Wijk, and A. Hansson, “Teaching multivariable control using the quadruple-tank process,” 1999.
- [4] *RSLinx Classic Getting Results Guide*, Allan-Bradley and Rocwell Automation, 2014.
- [5] *User Manual - EtherNet/IP Network Configuration*, Allan-Bradley and Rocwell Automation, 2014.
- [6] M. Sugeno and G. T. Kang, “Structure identification of fuzzy model,” 1986.
- [7] L. A. Mozelli, “Controle fuzzy para sistemas takagi-sugeno: Condições aprimoradas e aplicações,” 2008.
- [8] A. M. Lyapunov, “The general problem of stability of motion,” 1992.
- [9] K. Tanaka and H. O. Wang, “Fuzzy control systems design and analysis: A linear matrix inequality approcah,” 2010.
- [10] H. O. Wang and K. Tanaka, “An lmi-based stable fuzzy control of nonlinear systems and its application to control of chaos,” *Proceedigns of 5th IEEE International Conference on Fuzzy Systems*, 1996.

## A CÓDIGOS UTILIZADOS

### A.1 MATLAB

#### A.1.1 Planta Simulada

```
function [A1,A2,A3,A4,a1,a2,a3,a4,g,k1,k2,gamma1,gamma2] = planta()
% Tank Specs
A1 = 28;
A2 = 32;
A3 = 28;
A4 = 32;

a1 = 0.071;
a2 = 0.057;
a3 = 0.071;
a4 = 0.057;

kc = 1;
g = 981;

% Minimum Phase
% Pump Specs
k1 = 3.33;
k2 = 3.35;

% Gamas
gamma1 = 0.60;
gamma2 = 0.70;

% % Non Minimum Phase
% Pump Specs
% k1 = 3.15;
% k2 = 3.29;
%
% % Gamas
% gamma1 = 0.43;
% gamma2 = 0.34;

end
```

#### A.1.2 Modelo TS com 4 Regras

```
function output = mainTS2()
% Fuzzy Sets
vec_h1 = [5 15];
vec_h2 = [5 15];
```

```

% System to Each Point
[Ai, Bi, vec_v1, vec_v2, vec_h3, vec_h4] = fuzzy_pontos_v2(vec_h1,vec_h2);
vec_v1
vec_v2
vec_h3
vec_h4
h01 = 5;
h02 = 5;
h03 = 0.0334;
h04 = 2.9076;
v01 = 3.2321;
v02 = 0.5716;

% Solving
samp_time = 0.01;
final_time = 1200;
tot_samps = final_time/samp_time;
t = linspace(0, final_time, tot_samps);
[t,Hv] = ode45(@quadtank,t,[h01 h02 h03 h04]);

% Vizualization
y = [ Hv(:,1) Hv(:,2) ];

% figure
% plot(t,y(:,1),'--b', 'LineWidth',2);
% title('Modelo TS')
% ylabel('Altura (cm)');
% xlabel('Tempo (s)');
% legend('h1 TS [5 15]');
% grid on
%
% figure
% plot(t,y(:,2),'--r', 'LineWidth',2);
% title('Modelo TS')
% ylabel('Altura (cm)');
% xlabel('Tempo (s)');
% legend('h2 TS [5 15]');
% grid on

mTS2_h = Hv;
save('mainTS2.mat');

% ///////////////
% ---- Fuzzy Takagi-Sugeno Model ---- %
% ///////////////
function dh = quadtank(t,h)
% Inference System
alphas = fuzzy_pertinencia(h(1), h(2), vec_h1, vec_h2);

A = zeros(4,4);
B = zeros(4,2);
resp = zeros(size(h));

```

```

% Inputs
u = [v1(t); v2(t)];

for i= 1:size(alphas,1)
    for j= 1:size(alphas,2)
        % A = A + alphas(i,j) * Ai(:,:,i,j);
        % B = B + alphas(i,j) * Bi(:,:,i,j);
        resp = resp + alphas(i,j) * (Ai(:,:,i,j)*[h(1)-vec_h1(i); h(2)-vec_h2(j); h(3)-vec_h3(j)]);
    end
end

% Result System
% dh = A*h + B*u;
dh = resp;
end

% /////////////////
% ---- Inputs ---- %
% //////////////////////

function v11 = v1(t)
    if (t < 100)
        v11 = v01;
    else
        v11 = v01+2;
    end
end

function v22 = v2(t)
    if (t < 500)
        v22 = v02;
    else
        v22 = v02+2;
    end
end

end

```

### A.1.3 Função de Cálculo dos "r" Sistemas

```

function [As, Bs, V1s, V2s, H3s, H4s] = fuzzy_pontos_v2(vec_h1, vec_h2)

[A1,A2,A3,A4,a1,a2,a3,a4,g,k1,k2,gamma1,gamma2] = planta();

outA = zeros(4,4,length(vec_h1),length(vec_h2));
outB = zeros(4,2,length(vec_h1),length(vec_h2));
outV1 = zeros(length(vec_h1),length(vec_h2));
outV2 = zeros(length(vec_h1),length(vec_h2));
outH3 = zeros(length(vec_h1),length(vec_h2));
outH4 = zeros(length(vec_h1),length(vec_h2));

for i1 = 1:length(vec_h1)

```

```

hp1 = vec_h1(i1);

for i2 = 1:length(vec_h2)
    hp2 = vec_h2(i2);

    v2 = ((a1*sqrt(2*g*hp1)*((gamma1-1)/gamma1))+ a2*sqrt(2*g*hp2))* (gamma1/(k2*(gamma1+g
    v1 = (a1*sqrt(2*g*hp1) + (gamma2-1)*(k2*v2))/(gamma1*k1);

    hp3 = (1/(2*g))*((((1-gamma2)*k2*v2)/a3)^2);
    hp4 = (1/(2*g))*((((1-gamma1)*k1*v1)/a4)^2);

    T1 = (A1/a1)*sqrt(2*hp1/g);
    T2 = (A2/a2)*sqrt(2*hp2/g);
    T3 = (A3/a3)*sqrt(2*hp3/g);
    T4 = (A4/a4)*sqrt(2*hp4/g);

    A = [-(1/T1) 0 (A3/(A1*T3)) 0;
          0 -(1/T2) 0 (A4/(A2*T4));
          0 0 -(1/T3) 0;
          0 0 0 -(1/T4)];

    B = [gamma1*k1/A1 0;
          0 gamma2*k2/A2;
          0 (1 - gamma2)*k2/A3;
          (1 - gamma1)*k1/A4 0];

    outA(:,:,i1,i2) = A;
    outB(:,:,i1,i2) = B;
    outV1(i1,i2) = v1;
    outV2(i1,i2) = v2;
    outH3(i1,i2) = hp3;
    outH4(i1,i2) = hp4;
end
end

As = outA;
Bs = outB;
V1s = outV1;
V2s = outV2;
H3s = outH3;
H4s = outH4;

```

#### A.1.4 Função de Cálculo da Pertinências

```

function alphas = fuzzy_pertinencia(h1t, h2t, vec_h1, vec_h2)

% /////////////////////////////////
% Triang. Membership to H1
% ///////////////////////////////
u1 = zeros(1,length(vec_h1));

if(h1t <= vec_h1(1))

```

```

u1(1) = 1;

elseif (h1t >= vec_h1(length(vec_h1)))
u1(length(vec_h1)) = 1;

else
for i = 1:(length(vec_h1)-1)
if((h1t >= vec_h1(i)) && (h1t < vec_h1(i+1)))
left = (i>2)*(i - 1) + 1;
u1(i) = trimf(h1t, [vec_h1(left) vec_h1(i) vec_h1(i+1)]);
u1(i+1) = 1 - u1(i);
end
end
end

% ///////////////////////////////////////////////////
% Triang. Membership to H2
% ///////////////////////////////////////////////////
u2 = zeros(1,length(vec_h2));

if(h2t <= vec_h2(1))
u2(1) = 1;

elseif (h2t >= vec_h2(length(vec_h2)))
u2(length(vec_h2)) = 1;

else
for i = 1:(length(vec_h2)-1)
if((h2t >= vec_h2(i)) && (h2t < vec_h2(i+1)))
left = (i>2)*(i - 1) + 1;
u2(i) = trimf(h2t, [vec_h2(left) vec_h2(i) vec_h2(i+1)]);
u2(i+1) = 1 - u2(i);
end
end
end
% Alphas
alphas = (u1') * u2;

```

### A.1.5 Simulink de Comunicação OPC

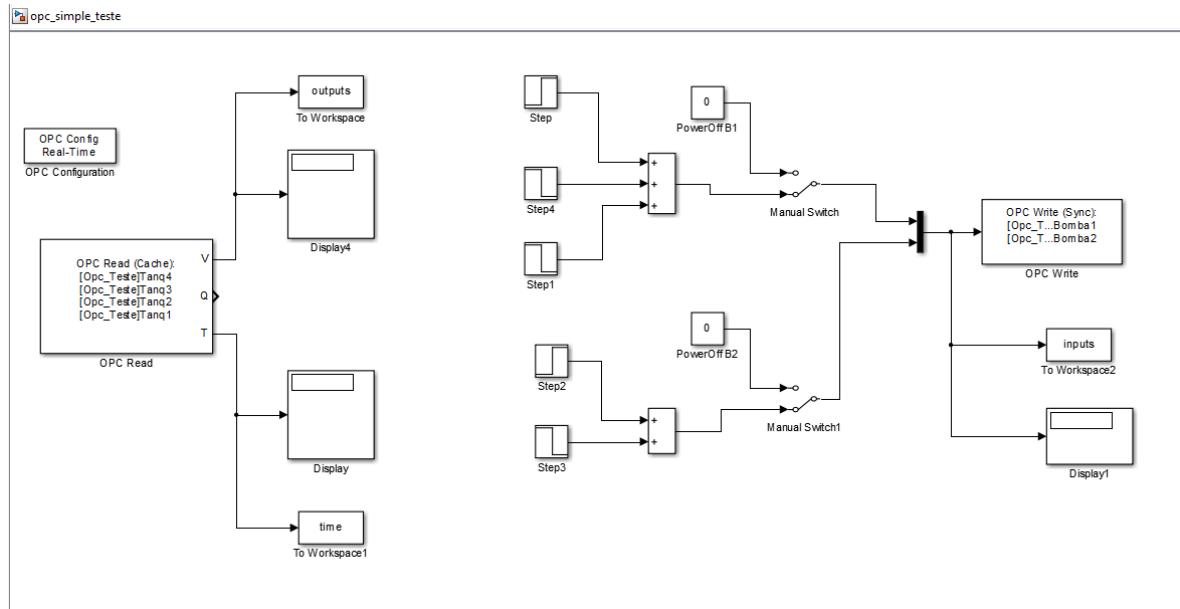


Figure A.1: Comunicação OPC Simulink-RSLinx

## A.2 TEXTO ESTRUTURADO

### A.2.1 SUBROTINA: FuzzyInputCalc

```

SBR(h1, h2, h3, h4, erro1, erro2);
    h11 := 12;
    h12 := 16;
    h21 := 12;
    h22 := 16;

    IF h1 < h11 THEN
        u11 := 1;
        u12 := 0;
    ELSIF h1 > h12 THEN
        u11 := 0;
        u12 := 1;
    ELSE
        u12 := (h1 - h11)/(h12 - h11);
        u11 := 1 - u12;
    END_IF;

    IF h2 < h21 THEN
        u21 := 1;
        u22 := 0;
    ELSIF h2 > h22 THEN
        u21 := 0;
        u22 := 1;
    ELSE

```

```

    u22 := (h2 - h21) / (h22 - h21);
    u21 := 1 - u22;
END_IF;

alpha1 := u11*u21;
alpha2 := u11*u22;
alpha3 := u12*u21;
alpha4 := u12*u22;

k1[0,0] := -10.1267;
k1[0,1] := -0.1726;
k1[0,2] := -0.0693;
k1[0,3] := -0.0304;
k1[0,4] := 7.4632;
k1[0,5] := 0.0782;
k1[1,0] := 0.4704;
k1[1,1] := -8.9962;
k1[1,2] := 0.0234;
k1[1,3] := -0.0081;
k1[1,4] := -0.1735;
k1[1,5] := 11.3821;

k2[0,0] := -9.7178;
k2[0,1] := -0.2288;
k2[0,2] := -0.0433;
k2[0,3] := -0.0878;
k2[0,4] := 7.1828;
k2[0,5] := 0.0979;
k2[1,0] := 0.5970;
k2[1,1] := -8.9866;
k2[1,2] := -0.0014;
k2[1,3] := -0.0096;
k2[1,4] := -0.2192;
k2[1,5] := 11.3719;

k3[0,0] := -10.1655;
k3[0,1] := -0.1445;
k3[0,2] := -0.0628;
k3[0,3] := -0.0301;
k3[0,4] := 7.4634;
k3[0,5] := 0.0680;
k3[1,0] := 0.3944;
k3[1,1] := -8.7470;
k3[1,2] := 0.0221;
k3[1,3] := -0.0071;
k3[1,4] := -0.1460;
k3[1,5] := 11.0130;

k4[0,0] := -9.7327;
k4[0,1] := -0.2180;
k4[0,2] := -0.0472;
k4[0,3] := -0.0874;
k4[0,4] := 7.1831;

```

```

k4[0,5] := 0.0939;
k4[1,0] := 0.5519;
k4[1,1] := -8.7291;
k4[1,2] := 0.0082;
k4[1,3] := -0.0090;
k4[1,4] := -0.2028;
k4[1,5] := 11.0068;

FOR row:=0 TO 1 BY 1 DO
    FOR col:=0 TO 5 BY 1 DO
        K[row,col] := alpha1*k1[row,col] + alpha2*k2[row,col];
        K[row,col] := K[row,col] + alpha3*k3[row,col] + alpha4*k4[row,col];
    END_FOR;
END_FOR;

u1 := -K[0,0]*h1 - K[0,1]*h2 - K[0,2]*h3 - K[0,3]*h4 + K[0,4]*erro1 + K[0,5]*erro2;
u2 := -K[1,0]*h1 - K[1,1]*h2 - K[1,2]*h3 - K[1,3]*h4 + K[1,4]*erro1 + K[1,5]*erro2;

IF (u1 < 0) THEN
    Bomba1 := 0;
    IF (h1 > ref1) THEN
        EnableInt1 := 0;
    ELSE
        EnableInt1 := 1;
    END_IF;
ELSIF (u1 > 3.5) THEN
    Bomba1 := 70;
    IF (h1 < ref1) THEN
        EnableInt1 := 0;
    ELSE
        EnableInt1 := 1;
    END_IF;
ELSE
    Bomba1 := u1*20;
    EnableInt1 := 1;
END_IF;

IF (u2 < 0) THEN
    Bomba2 := 0;
    IF (h2 > ref2) THEN
        EnableInt2 := 0;
    ELSE
        EnableInt2 := 1;
    END_IF;
ELSIF (u2 > 3.5) THEN
    Bomba2 := 70;
    IF (h1 < ref1) THEN
        EnableInt2 := 0;
    ELSE
        EnableInt2 := 1;
    END_IF;
ELSE
    Bomba2 := u2*20;

```

```
    EnableInt1 := 1;  
    END_IF;  
RET();
```