

UNIVERSIDADE DE SÃO PAULO – USP
ESCOLA DE ENGENHARIA DE SÃO CARLOS – ESSC
DEPARTAMENTO DE ENGENHARIA ELÉTRICA – SEL

PROJETO DE FORMATURA II
TRABALHO DE CONCLUSÃO DE CURSO

ESTUDO COMPARATIVO ENTRE CONTROLE ADAPTATIVO E CONTROLE FUZZY

Cyro de Araújo Gianelli

São Carlos

2013

CYRO DE ARAÚJO GIANELLI

**ESTUDO COMPARATIVO ENTRE
CONTROLE ADAPTATIVO E
CONTROLE FUZZY**

Trabalho de Conclusão de Curso apresentado à Escola de
Engenharia de São Carlos, da Universidade de São Paulo

Curso de Engenharia Elétrica com ênfase em Eletrônica

ORIENTADOR: JERSON BARBOSA DE VARGAS

São Carlos

2013

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Gianelli, Cyro de Araújo
G433e Estudo comparativo entre controle adaptativo e
controle Fuzzy / Cyro de Araújo Gianelli; orientador
Jerson Barbosa de Vargas. São Carlos, 2013.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Eletrônica) -- Escola de Engenharia de São
Carlos da Universidade de São Paulo, 2013.

1. Controle adaptativo. 2. Controle Fuzzy. 3.
controlador de motor de corrente contínua. I. Título.

FOLHA DE APROVAÇÃO

Nome: Cyro de Araujo Gianelli

Título: "Estudo comparativo entre controle adaptativo e controle Fuzzy"

Trabalho de Conclusão de Curso defendido e aprovado
em 21 / 11 / 2013,

com NOTA 9,5 (NOVE, CINCO), pela Comissão Julgadora:

*Prof. Assistente Jerson Barbosa de Vargas - (Orientador -
SEL/EESC/USP)*

Prof. Assistente Edson Gesualdo - (SEL/EESC/USP)

Prof. Dr. Valdir Grassi Júnior - (SEL/EESC/USP)

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Homero Schiabel

AGRADECIMENTOS

Agradeço à Universidade de São Paulo, à Escola de Engenharia de São Carlos e ao Departamento de Engenharia Elétrica, que proporcionaram a minha formação como Engenheiro Eletricista.

Agradeço a todos os professores, que contribuíram para o meu aprendizado durante a graduação, e contribuíram para o meu enriquecimento cultural, técnico e científico.

Agradeço ao meu orientador, Professor Jerson, que contribui de forma ímpar, sendo um docente atencioso e prestativo em todas as dúvidas e orientações.

Agradeço a meus amigos, que tornaram esta jornada mais agradável e inesquecível.

Agradeço, por fim, aos meus pais, irmãs e minha namorada, que me deram a força necessária para enfrentar as dificuldades e concluir com êxito esta caminhada.

RESUMO

GIANELLI, CYRO DE A.; **ESTUDO COMPARATIVO ENTRE CONTROLE ADAPTATIVO E CONTROLE FUZZY**. Trabalho de Conclusão de Curso – Escola de Engenharia de São Carlos, Universidade de São Paulo, 2013.

Motores de corrente contínua tem grande aplicação em uma ampla gama de processos industriais, residenciais, robótica, entre outros. O controle de velocidade e posição do motor é de extrema importância nessas aplicações. Várias são as técnicas disponíveis para o controle de tal processo, como controle clássico, controle adaptativo, controle fuzzy, controle robusto, entre outros. Controle clássico é a escolha mais utilizada por ser robusta, eficiente e facilmente desenvolvida, porém tem limitações. Variações na planta controlada podem não ser acompanhadas pelo sistema de controle. Desta forma, surge a necessidade de um controle mais preciso. Este trabalho trata da comparação entre dois métodos de controle inteligente, o sistema de controle adaptativo por modelo de referência e o controle fuzzy. Ambos são mais robustos e menos limitados que o controle clássico. Este estudo tem como objetivo descrever detalhadamente o processo de desenvolvimento de cada um dos controladores, assim como compará-los perante sua eficiência e robustez.

Palavras-chave: Controle adaptativo, controle Fuzzy, controlador de motor de corrente contínua.

ABSTRACT

GIANELLI, CYRO DE A.; **COMPARATIVE STUDY BETWEEN ADAPTIVE CONTROL AND FUZZY CONTROL**. Undergraduate Final Project – Escola de Engenharia de São Carlos, Universidade de São Paulo, 2013.

Direct current motors has large applications in a wide range of industrial and residential processes, robotics, among others. The speed control and motor position is extremely important in these applications. Several techniques are available to control this process, as classical control, adaptive control, fuzzy control, robust control, among others. Classic control is the most used choice because is robust, efficient and easily developed, but has limitations. Variations in the controlled plant may not be followed by the control system. Thus, arises the need for a more precise control. This work includes the comparison between two methods of intelligent and adaptive control, adaptive control system by model reference and fuzzy control, where both are more robust and less limited than the classical control. This comparative study of adaptive control and fuzzy control aims to describe in details the development process of each drivers, as well as compare them among their efficiency and robustness.

Key-words: Adaptive control, Fuzzy control, Direct current motor controller.

Sumário

Índice de Figuras	3
Índice de Tabelas.....	5
1 Introdução	6
1.1 Proposta do trabalho.....	7
1.2 Organização.....	7
2 Motor de corrente contínua	8
2.1 Modelagem matemática do motor de corrente contínua	9
2.2 Simulação do motor de corrente contínua.....	12
3 Controle PID	15
3.1 Métodos de sintonia do PID.....	16
3.1.1 Método do ajuste manual.....	17
3.1.2 Método Ziegler-Nichols	17
3.1.3 Método de Ajuste via <i>software</i> (lugar de raízes).....	19
4 Controle Adaptativo	21
4.1 Escalonamento de ganho.....	22
4.2 Controlador auto-sintonizável	23
4.3 Sistema adaptativo por modelo de referência.....	23
4.3.1 Método do gradiente.....	24
4.3.2 Ganhos de adaptação.....	26
5 Fuzzy.....	27
5.1 Fundamentos da teoria Fuzzy.....	27
5.1.1 Funções pertinência.....	28
5.1.2 Operações Fuzzy	30
5.1.3 Defuzzyficação.....	31
5.2 Controle Fuzzy	32
5.3 Fuzzy no Matlab.....	35
6 Controladores	38
6.1 Controle PID	38
6.2 Sistema adaptativo por modelo de referência.....	39
6.2.1 Mecanismo de adaptação.....	39
6.2.2 Ganhos de adaptação	45
6.3 Controle Fuzzy	46
7 Simulações e resultados	50

7.1	Degraus consecutivos	51
7.1.1	MRAS	51
7.1.2	Fuzzy	53
7.1.3	Comparativo	54
7.2	Trem de pulsos	55
7.2.1	MRAS	55
7.2.2	Fuzzy	59
7.2.3	Comparativo	61
7.3	Aplicação de Carga	63
7.3.1	MRAS	64
7.3.2	Fuzzy	65
7.3.3	Comparativo	66
7.4	Variações Paramétricas	67
7.4.1	Variação positiva de 10% na indutância e 10% na resistência.....	67
7.4.2	Variação positiva de 50% na inércia do motor (J) e 50% no coeficiente de atrito estático (B)	70
8	Conclusão	74
	Referências	76
	Apêndice	77
	Desenvolvimento via Matlab do Motor, PID e Modelo de Referência	77
	Desenvolvimento via Matlab do método para encontrar os ganhos de adaptação, diagrama de simulink para o mesmo	78
	Desenvolvimento via Matlab do método para encontrar Kp, Ki e Kd do sistema Fuzzy, diagrama de simulink para o mesmo	80

Índice de Figuras

FIGURA 2.1 - ESQUEMA DE UM MOTOR DE CORRENTE CONTÍNUA	9
FIGURA 2.2 - DIAGRAMA ELETROMECÂNICO DO MOTOR DE CORRENTE CONTÍNUA	9
FIGURA 2.3 - DIAGRAMA DE BLOCOS DE UM MOTOR DE CORRENTE CONTÍNUA	11
FIGURA 2.4 - RESPOSTA DO MOTOR DE CORRENTE CONTÍNUA AO DEGRAU UNITÁRIO	13
FIGURA 2.5 - RESPOSTA AO IMPULSO DO MOTOR DE CORRENTE CONTÍNUA	13
FIGURA 2.6 - RESPOSTA DO MOTOR DE CORRENTE CONTÍNUA À APLICAÇÃO DE CARGA	14
FIGURA 2.7 - CARGA APLICADA	14
FIGURA 3.1 - DIAGRAMA DE BLOCOS DE UM SISTEMA COM CONTROLE PID	15
FIGURA 3.2 - CURVA DE RESPOSTA AO DEGRAU E PARÂMETROS	18
FIGURA 3.3 - OSCILAÇÃO CRÍTICA	19
FIGURA 3.4 - ADIÇÃO DE PÓLOS E ZEROS DO CONTROLADOR NA FERRAMENTA RLTOOL	19
FIGURA 3.5 - ALTERAÇÃO DOS LOCAIS DOS PÓLOS E ZEROS DO SISTEMA	20
FIGURA 3.6 - RESPOSTA DO SISTEMA DE ACORDO COM A VARIAÇÃO DO LOCAL DE SEUS PÓLOS E ZEROS	20
FIGURA 4.1 - DIAGRAMA DE BLOCOS DE UM SISTEMA COM CONTROLE ADAPTATIVO	21
FIGURA 4.2 - DIAGRAMA DE BLOCOS DE UM SISTEMA COM ESCALONAMENTO DE GANHO	22
FIGURA 4.3 - DIAGRAMA DE BLOCOS DE UM SISTEMA COM CONTROLADOR AUTO-SINTONIZÁVEL	23
FIGURA 4.4 - DIAGRAMA DE BLOCOS DE UM SISTEMA ADAPTATIVO POR MODELO DE REFERÊNCIA (MRAS)	24
FIGURA 5.1 - FUNÇÃO DE PERTINÊNCIA TRIANGULAR	28
FIGURA 5.2 - FUNÇÃO PERTINÊNCIA TRAPEZOIDAL	29
FIGURA 5.3 - FUNÇÃO PERTINÊNCIA GAUSSIANA	29
FIGURA 5.4 - PRINCIPAIS NORMAS	31
FIGURA 5.5 - PRINCIPAIS CO-NORMAS	31
FIGURA 5.6 - ESTRUTURA DE UM CONTROLADOR FUZZY	34
FIGURA 5.7 - FIS EDITOR	35
FIGURA 5.8 - MEMBERSHIP FUNCTION EDITOR	36
FIGURA 5.9 - RULE EDITOR	36
FIGURA 5.10 - RULE VIEWER	37
FIGURA 5.11 - SURFACE VIEWER	37
FIGURA 6.1 - SISTEMA COM CONTROLE PID	38
FIGURA 6.2 - DIAGRAMA DE BLOCOS DO SISTEMA ADAPTATIVO POR MODELO DE REFERÊNCIA UTILIZADO ..	42
FIGURA 6.3 - MECANISMO DE AJUSTE PROPORCIONAL	42
FIGURA 6.4 - MECANISMO DE AJUSTE INTEGRAL	42
FIGURA 6.5 - MECANISMO DE AJUSTE DERIVATIVO	43
FIGURA 6.6 - SISTEMA ADAPTATIVO POR MODELO DE REFERÊNCIA	44
FIGURA 6.7 - VARIÁVEL ERRO NO <i>TOOLBOX</i> FUZZY	46
FIGURA 6.8 - VARIÁVEL VARIAÇÃO DO ERRO NO <i>TOOLBOX</i> FUZZY	47
FIGURA 6.9 - VARIÁVEL AÇÃO DE CONTROLE NO <i>TOOLBOX</i> FUZZY	48
FIGURA 6.10 - SISTEMA COM CONTROLE FUZZY	49
FIGURA 7.1 - COMPARATIVO ENTRE RESPOSTA DO MODELO DE REFERÊNCIA E DO SISTEMA ADAPTATIVO ..	51
FIGURA 7.2 - ERRO DA RESPOSTA DO SISTEMA ADAPTATIVO EM RELAÇÃO AO MODELO DE REFERÊNCIA	51
FIGURA 7.3 - AÇÃO DE CONTROLE DO SISTEMA ADAPTATIVO	52
FIGURA 7.4 - RESPOSTA DO SISTEMA FUZZY	53
FIGURA 7.5 - AÇÃO DE CONTROLE DO SISTEMA FUZZY	53
FIGURA 7.6 - COMPARATIVO ENTRE A RESPOSTA DO SISTEMA ADAPTATIVO E DO SISTEMA FUZZY	54

FIGURA 7.7 - COMPARATIVO ENTRE A AÇÃO DE CONTROLE DO SISTEMA ADAPTATIVO E DO SISTEMA FUZZY	54
FIGURA 7.8 - COMPARATIVO ENTRE A RESPOSTA DO MODELO DE REFERÊNCIA E DO SISTEMA ADAPTATIVO	55
FIGURA 7.9 - DETALHE DO COMPARATIVO EM SEU INÍCIO	55
FIGURA 7.10 - DETALHE DO COMPARATIVO AO LONGO DA SIMULAÇÃO	56
FIGURA 7.11 - ERRO DA RESPOSTA DO SISTEMA ADAPTATIVO EM RELAÇÃO AO MODELO DE REFERÊNCIA	56
FIGURA 7.12 - AÇÃO DE CONTROLE DO SISTEMA ADAPTATIVO	57
FIGURA 7.13 - COMPARATIVO EVIDENCIANDO A CONVERGÊNCIA DO PARÂMETRO PROPORCIONAL	57
FIGURA 7.14 - COMPARATIVO EVIDENCIANDO A CONVERGÊNCIA DO PARÂMETRO INTEGRATIVO	58
FIGURA 7.15 - COMPARATIVO EVIDENCIANDO A CONVERGÊNCIA DO PARÂMETRO DERIVATIVO	58
FIGURA 7.16 - RESPOSTA DO SISTEMA FUZZY	59
FIGURA 7.17 - DETALHE DA RESPOSTA DO SISTEMA FUZZY	59
FIGURA 7.18 - AÇÃO DE CONTROLE DO SISTEMA FUZZY	60
FIGURA 7.19 - DETALHE DA AÇÃO DE CONTROLE DO SISTEMA FUZZY	60
FIGURA 7.20 - COMPARATIVO ENTRE A RESPOSTA DO SISTEMA ADAPTATIVO E DO SISTEMA FUZZY	61
FIGURA 7.21 - DETALHE DO COMPARATIVO DA RESPOSTA	61
FIGURA 7.22 - COMPARATIVO ENTRE A AÇÃO DE CONTROLE DO SISTEMA ADAPTATIVO E DO SISTEMA FUZZY	62
FIGURA 7.23 - DETALHE DA AÇÃO DE CONTROLE DOS SISTEMAS	62
FIGURA 7.24 - CARGA APLICADA	63
FIGURA 7.25 - RESPOSTA DO SISTEMA ADAPTATIVO	64
FIGURA 7.26 - AÇÃO DE CONTROLE DO SISTEMA ADAPTATIVO	64
FIGURA 7.27 - RESPOSTA DO SISTEMA FUZZY	65
FIGURA 7.28 - AÇÃO DE CONTROLE DO SISTEMA FUZZY	65
FIGURA 7.29 - COMPARATIVO ENTRE A RESPOSTA DO SISTEMA ADAPTATIVO E DO SISTEMA FUZZY	66
FIGURA 7.30 - COMPARATIVO ENTRE A AÇÃO DE CONTROLE DO SISTEMA ADAPTATIVO E DO SISTEMA FUZZY	66
FIGURA 7.31 - COMPARATIVO DO INÍCIO ENTRE AS RESPOSTA DO MODELO DE REFERÊNCIA E DO PROCESSO ALTERADO	67
FIGURA 7.32 - COMPARATIVO AO LONGO DO TEMPO ENTRE AS RESPOSTAS DO MODELO DE REFERÊNCIA E DO PROCESSO ALTERADO	67
FIGURA 7.33 - INÍCIO DA AÇÃO DE CONTROLE DO PROCESSO ALTERADO	68
FIGURA 7.34 - AÇÃO DE CONTROLE DO PROCESSO ALTERADO	68
FIGURA 7.35 - COMPARATIVO ENTRE AS RESPOSTAS DO PROCESSO SEM ALTERAÇÃO E DO PROCESSO ALTERADO	69
FIGURA 7.36 - COMPARATIVO ENTRE AS AÇÕES DE CONTROLE DO PROCESSO SEM ALTERAÇÃO E DO PROCESSO ALTERADO	69
FIGURA 7.37 - COMPARATIVO DO INÍCIO ENTRE AS RESPOSTA DO MODELO DE REFERÊNCIA E DO PROCESSO ALTERADO	70
FIGURA 7.38 - COMPARATIVO AO LONGO DO TEMPO ENTRE AS RESPOSTA DO MODELO DE REFERÊNCIA E DO PROCESSO ALTERADO	70
FIGURA 7.39 - INÍCIO DA AÇÃO DE CONTROLE DO PROCESSO ALTERADO	71
FIGURA 7.40 - AÇÃO DE CONTROLE DO PROCESSO ALTERADO	71
FIGURA 7.41 - COMPARATIVO ENTRE AS RESPOSTAS DO PROCESSO SEM ALTERAÇÃO E DO PROCESSO ALTERADO	72
FIGURA 7.42 - COMPARATIVO ENTRE AS AÇÕES DE CONTROLE DO PROCESSO SEM ALTERAÇÃO E DO PROCESSO ALTERADO	72
FIGURA I.0.1 - DIAGRAMA UTILIZADO PARA CÁLCULO DOS GANHOS DE ADAPTAÇÃO	79
FIGURA I.0.2 - DIAGRAMA UTILIZADO PARA CÁLCULO DE K_p , K_i E K_d	81

Índice de Tabelas

TABELA 2.1 - PARÂMETROS DE CONSTRUÇÃO DO MOTOR DE CORRENTE CONTÍNUA	12
TABELA 2.2 - CARACTERÍSTICAS DO MOTOR	13
TABELA 3.1 - COMPARATIVO ENTRE MÉTODOS DE SINTONIA PID	16
TABELA 3.2 - EFEITO DA VARIAÇÃO DO AUMENTO DOS PARÂMETROS	17
TABELA 3.3 - PARÂMETROS OBTIDOS COM O MÉTODO RESPOSTA AO DEGRAU - ZIEGLER-NICHOLS	18
TABELA 3.4 - PARÂMETROS OBTIDOS COM O MÉTODO DA OSCILAÇÃO CRÍTICA – ZIEGLER-NICHOLS	18
TABELA 5.1 - PRINCIPAIS NORMAS, CO-NORMAS E SEUS NOMES.....	30
TABELA 6.1 - VALORES ENCONTRADOS PARA OS GANHOS DE ADAPTAÇÃO.....	45
TABELA 6.2 - FUNÇÕES PERTINÊNCIA DA VARIÁVEL ERRO	46
TABELA 6.3 - FUNÇÕES PERTINÊNCIA DA VARIÁVEL VARIAÇÃO DO ERRO.....	47
TABELA 6.4 - FUNÇÕES PERTINÊNCIA DA VARIÁVEL AÇÃO DE CONTROLE.....	47
TABELA 6.5 - REGRAS DO CONTROLE FUZZY	48
TABELA 6.6 - VALORES PARA K_p , K_i E K_d ENCONTRADOS SEGUNDO MÉTODOS DE MINIMIZAÇÃO DO ERRO	49

1 Introdução

A aplicação de melhorias contínuas no processo de produção, desenvolvimento e manutenção de sistemas tem levado as empresas à busca de novas tecnologias e aprimoramento das já existentes para aumentar sua eficiência e manter sua competitividade no mercado. A área de controle tem grande papel na eficiência de sistemas e, portanto é muito estudada e desenvolvida.

A maioria dos processos de controle industrial hoje no mundo opera com controladores proporcional-integral-derivativo (PID). Isto se deve à sua simplicidade e robustez para a maioria das aplicações. Porém, quando dispomos de processos que possuem não linearidades, atrasos de transporte ou parâmetros variantes no tempo, estes controladores respondem de forma insatisfatória.

Desta forma, devemos recorrer a outros métodos de controle, como por exemplo, o controle adaptativo e o controle Fuzzy. Entretanto, esta é uma via de mão dupla, quanto mais preciso o controlador, mais complexo é seu desenvolvimento.

Como o próprio nome diz, controlador adaptativo é aquele capaz de modificar sua resposta em função da dinâmica dos parâmetros do processo e distúrbios. Existem vários métodos onde cada um apresenta vantagens e desvantagens em relação ao outro.

Um fato que merece destaque é o aumento da utilização de recursos computacionais para o controle de sistemas. Esses recursos permitem os mais diversos tipos de controle. Neste contexto, o controle Fuzzy aparece como uma boa alternativa, se mostrando muito eficiente, robusto e aplicável a uma ampla gama de processos. O uso de controladores Fuzzy proporciona a possibilidade de supervisão inteligente usando apenas informações qualitativas sobre a operação do sistema, não havendo necessidade de modelagem matemática.

Para o trabalho considerado, foi utilizado um motor de corrente contínua devido à sua ampla utilização em várias aplicações e sua tradição de uso como máquinas de velocidade e posição controlável.

1.1 Proposta do trabalho

A proposta deste trabalho é o controle da velocidade angular de um motor de corrente contínua, através do uso de controladores PID, adaptativo por modelo de referência e Fuzzy, assim como o estudo comparativo de tais técnicas de controle.

O uso do controlador PID se deu pelo fato de ser uma das técnicas mais utilizadas e estudadas. O sistema adaptativo por modelo de referência foi feito para contornar o problema das não linearidades do sistema, como também as variações dos parâmetros da planta. Já o controle Fuzzy foi desenvolvido para verificar a eficiência e robustez de um controle relativamente novo e projetado via métodos computacionais.

Todo o trabalho foi desenvolvido no *software* Matlab - Simulink, realizando tanto os cálculos como as simulações dos sistemas.

1.2 Organização

No capítulo 2 temos a introdução do motor de corrente contínua, explanando sua configuração, vantagens e desvantagens, e seu funcionamento e dinâmica, assim como seu modelamento matemático e algumas simulações.

No capítulo 3 tratamos do controle PID, discutindo sua base teórica, métodos para desenvolvimento, cálculo de parâmetros e desenvolvimento via *software*.

No capítulo 4 tratamos do controle adaptativo, definindo sua base teórica, discutindo seus métodos de projeto, cálculo das teorias e cálculo dos ganhos de adaptação.

No capítulo 5 temos a introdução do controle Fuzzy, inicialmente tratando dos fundamentos da teoria Fuzzy e suas operações. Posteriormente entramos no âmbito do controlador, discutindo suas etapas. Ao final demonstramos a sua utilização via *software* Matlab.

No capítulo 6 é feito o desenvolvimento dos controladores utilizados, demonstrando passo-a-passo seus projetos.

No capítulo 7 temos as simulações e resultados obtidos, juntamente com a discussão de tais dados, assim como comparativo dos métodos de controle.

No capítulo 8, por fim, conclui-se o trabalho, retomando os principais pontos estudados e principais conclusões obtidas.

2 Motor de corrente contínua

Motores de corrente contínua apresentam alta confiabilidade, flexibilidade e baixo custo, por isso tem sido amplamente utilizados em várias aplicações industriais, aplicações residenciais, robótica, carros elétricos, e situações onde a portabilidade é uma necessidade, uma vez que baterias e células solares podem prover a energia necessária para seu funcionamento.

Um motor de corrente contínua utiliza eletricidade e o campo magnético para produzir o torque necessário para rotacionar a parte móvel do motor. A tensão aplicada modifica a velocidade, enquanto a corrente nos enrolamentos da armadura modificam o torque do motor. Se aplicada uma carga ao eixo, a velocidade tenderá a diminuir ao passo que a corrente aumentará para tentar contrapor esta variação.

O desenvolvimento de controladores de alto desempenho de motores de corrente contínua é, portanto, muito importante para sua eficiência e uso. Geralmente, um controlador de alto desempenho de motor deve ter uma boa dinâmica de controle de comando de velocidade e boa resposta de regulação de carga.

Os motores de corrente contínua tem uma longa tradição de uso como máquinas de velocidade e posição controlável e uma ampla variedade de opções que envolvem esta característica. Nestas aplicações, o motor necessita ser precisamente controlado para apresentar o desempenho requerido.

Algumas são as dificuldades do controle de motores de corrente contínua, dentre elas:

- Não linearidade do motor;
- Propagação de oscilações provenientes de ruído;
- Parâmetros desconhecidos;
- Variações na dinâmica dos parâmetros;

O maior problema na aplicação de algoritmos clássicos de controladores é sem dúvida o efeito da não linearidade do motor de corrente contínua. A característica não linear, como a saturação e a fricção do motor, pode degradar o desempenho dos controladores convencionais.

Na figura (2.1), adaptada de: <http://www.electrical-knowhow.com/2012/05/electrical-motors-basic-components.html>, temos em detalhes os componentes e sua disposição em um motor de corrente contínua, seguindo sua construção.

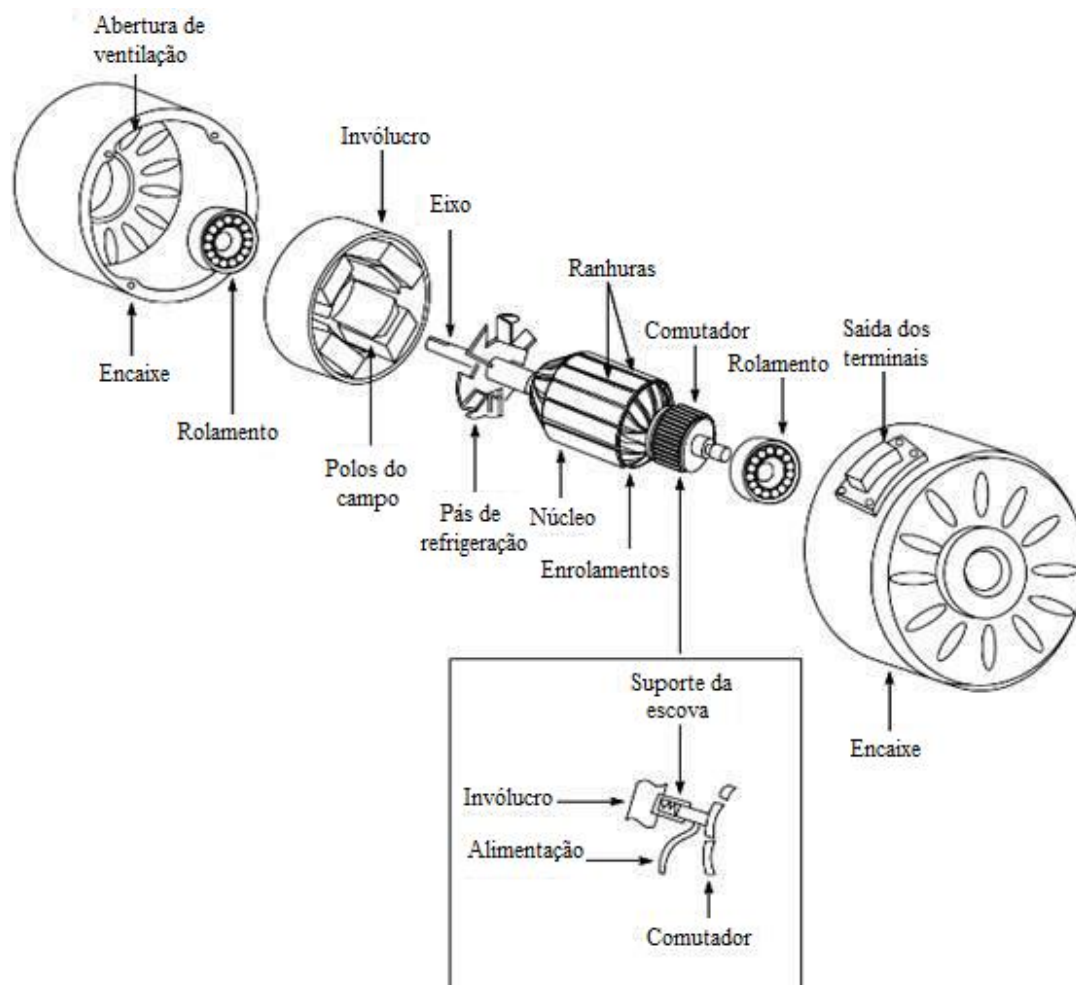


Figura 2.1 - Esquema de um motor de corrente contínua

2.1 Modelagem matemática do motor de corrente contínua

O modelo matemático utilizado nesta abordagem segue o proposto por Oliveira, Aguiar e Vargas em [1]. O diagrama eletromecânico do motor é mostrado na figura (2.2).

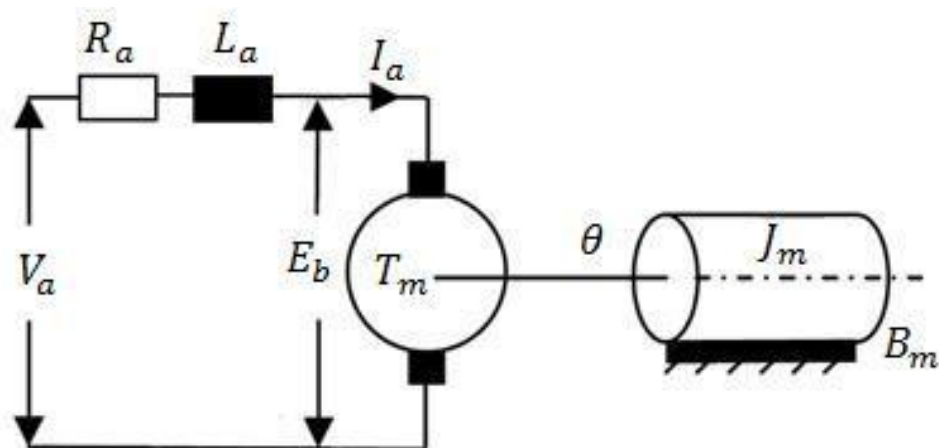


Figura 2.2 - Diagrama eletromecânico do motor de corrente contínua

Onde:

R_a = resistência de armadura (Ω);

L_a = indutância de armadura (H);

V_a = tensão da armadura (V);

I_a = corrente da armadura (A);

E_b = força contra eletromotriz (V);

T_m = torque do motor (Nm);

w = velocidade angular (rad/s);

Θ = posição angular (rad);

J_m = inercia do rotor (kgm^2);

B_m = coeficiente de atrito viscoso (Nms/rad);

K_e = constante contra eletromotriz (Vs/rad);

K_t = constante de torque (Nm/A)

Como equações que regem a dinâmica do sistema, temos:

$$V_a(t) = R_a * I_a(t) + L_a \frac{d}{dt} I_a(t) + K_e * w(t) \quad (2.1)$$

$$T_m(t) = K_t * I_a(t) \quad (2.2)$$

$$T_m(t) = J \frac{d}{dt} w(t) + b * w(t) \quad (2.3)$$

$$E_b(t) = K_e * w(t) \quad (2.4)$$

Realizando a transformada de Laplace:

$$V_a(s) = R_a * i_a(s) + L_a * s * I_a(s) + K_e * w(s) \quad (2.5)$$

$$T_m(s) = K_t * I_a(s) \quad (2.6)$$

$$T_m(s) = J * w(s) * s + b * w(s) \quad (2.7)$$

Assim, de (2.7) em (2.6):

$$Ia(s) = \frac{J*w(s)*s+b*w(s)}{Kt} \quad (2.8)$$

Substituindo (2.8) em (2.5):

$$Va(s) = (La * s + Ra) * \frac{J*w(s)*s+b*w(s)}{Kt} + Ke * w(s) \quad (2.9)$$

Portanto:

$$\frac{w(s)}{Va(s)} = \frac{Kt}{(La*s+Ra)*(J*s+b)+Ke*Kt} \quad (2.10)$$

E como:

$$\theta(s) = \frac{1}{s} w(s) \quad (2.11)$$

Segue que:

$$\frac{\theta(s)}{Va(s)} = \frac{1}{s} \left(\frac{Kt}{(La*s+Ra)*(J*s+b)+Ke*Kt} \right) \quad (2.12)$$

Esquemáticamente, o motor de corrente contínua é mostrado na figura (2.3).

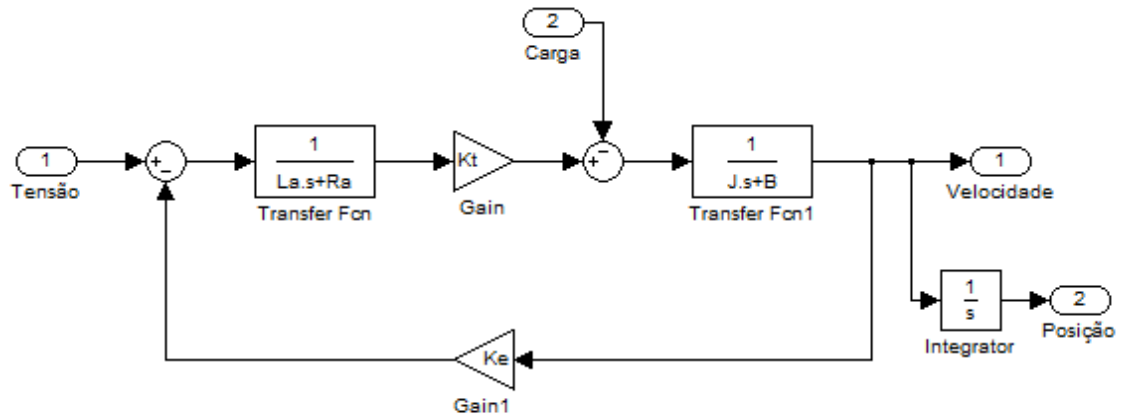


Figura 2.3 - Diagrama de blocos de um motor de corrente contínua

2.2 Simulação do motor de corrente contínua

O motor de corrente contínua utilizado apresenta os parâmetros de construção da tabela (2.1). Os valores dos parâmetros utilizados são os valores nominais do motor presente em [1].

O motor utilizado apresenta estabilidade natural, assim como a grande maioria dos motores existentes.

Tabela 2.1 - Parâmetros de construção do motor de corrente contínua

Resistência de armadura	$R_a = 1,63 \, \Omega$
Indutância de armadura	$L_a = 0,003 \, \text{H}$
Inércia do motor	$J = 3,67 * 10^{-5} \, \text{kgm}^2$
Constante atrito viscoso	$B = 1,1224 * 10^{-6} \, \text{Nms/rad}$
Constante contra eletromotriz	$K_e = 0,0678 \, \text{Vs/rad}$
Constante de torque	$K_t = 0,0678 \, \text{Nm/A}$

Temos resultante a seguinte função transferência correspondendo ao motor:

$$G_{motor} = \frac{0,0678}{1,101*10^{-7} s^2 + 5,982*10^{-5} s + 0,004599} \quad (2.13)$$

De acordo com as equações acima desenvolvidas e com os parâmetros do motor de corrente contínua, utilizou-se o *software* Matlab para determinar a resposta dinâmica de saída do motor. De acordo com a equação (2.10) temos que a entrada do sistema é em V, dada por V_a e a saída em rad/s dada por w . O código desenvolvido em Matlab está no apêndice deste trabalho.

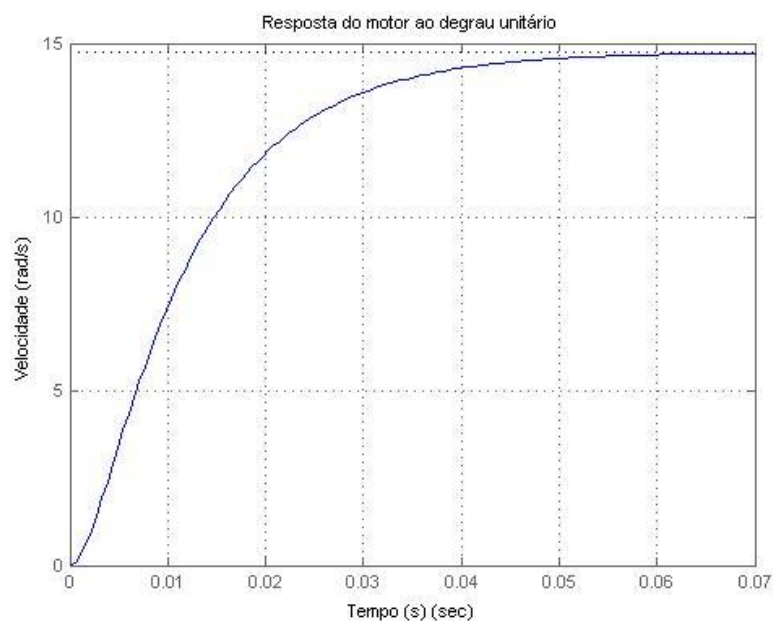


Figura 2.4 - Resposta do motor de corrente contínua ao degrau unitário

As características do motor estão descritas na tabela (2.2), após observação das figuras (2.4) e (2.5).

Tabela 2.2 - Características do motor

Tempo de acomodação	0,0447 s
Tempo de subida	0,0246 s
Regime de estado	14,7 rad/s

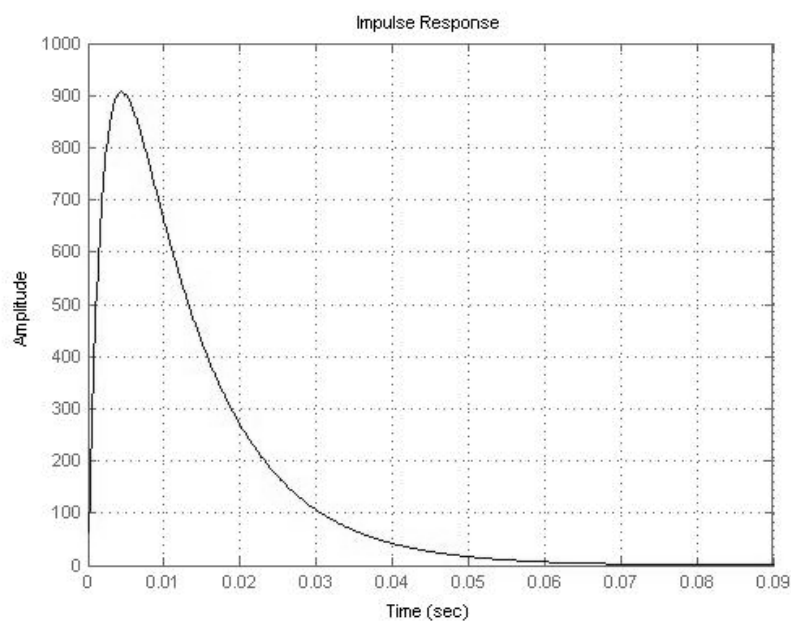


Figura 2.5 - Resposta ao impulso do motor de corrente contínua

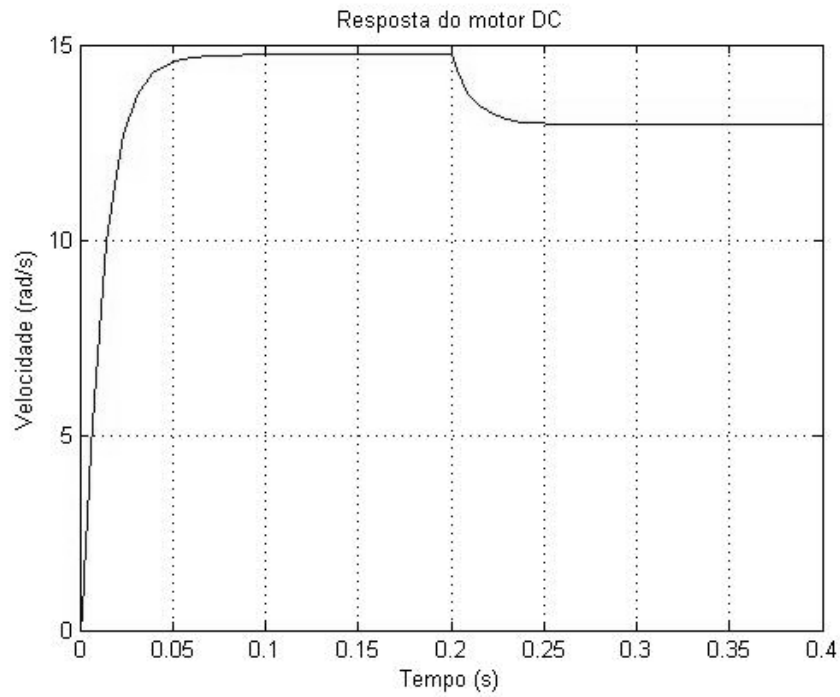


Figura 2.6 - Resposta do motor de corrente contínua à aplicação de carga

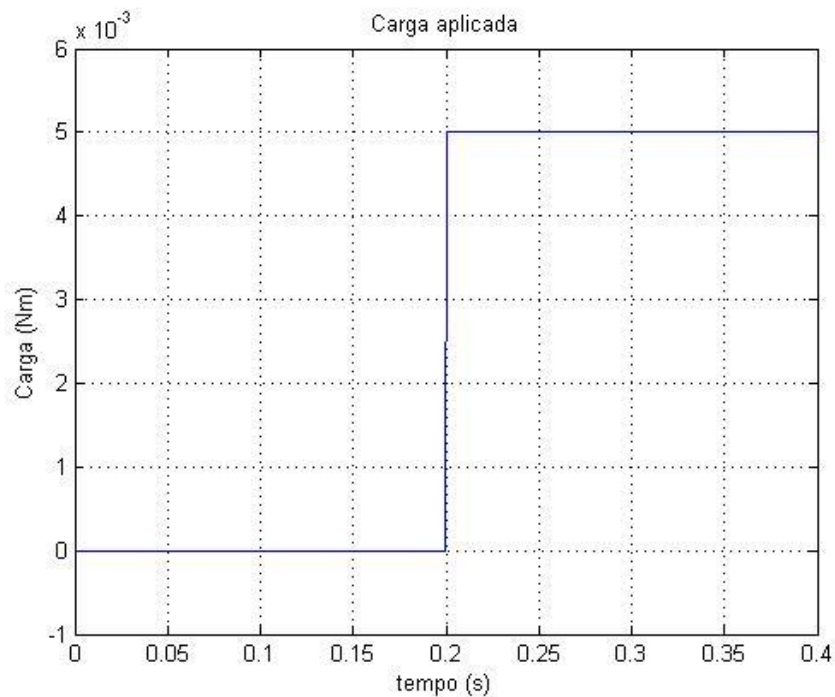


Figura 2.7 - Carga aplicada

Notamos pelas figuras (2.6) e (2.7) que aplicando carga ao eixo do motor, temos uma redução de sua velocidade, o que é indesejável em aplicações que exijam resposta constante e precisa.

3 Controle PID

De acordo com Dorf e Bishop: “A popularidade dos controladores PID pode ser atribuída parcialmente ao seu desempenho robusto sobre uma grande faixa de condições operacionais e em parte à sua simplicidade funcional, que permite aos engenheiros operá-los de maneira simples e correta. Para implementar um controlador destes, três parâmetros devem ser determinados para um dado processo: ganho proporcional, ganho integral e ganho derivativo” [2].

O controle proporcional-integrativo-derivativo (PID) é de longe o algoritmo de controle mais comumente utilizado. Seu princípio de funcionamento é calcular o erro entre a resposta do processo e um valor de referência, onde o controlador tenta minimizar este erro variando as suas ações de controle. A figura (3.1) demonstra um sistema com controle PID.

Sua grande robustez e larga utilização em uma grande gama de aplicações são devidas à interação de três processos distintos e simples:

- Ação proporcional: promove ação de controle razoável ao erro;
- Ação integrativa: garante que a resposta do sistema seja igual ao valor de referência, levando o erro a zero;
- Ação derivativa: promove a estabilidade do sistema, diminuindo o tempo necessário para atingir o regime de estado;

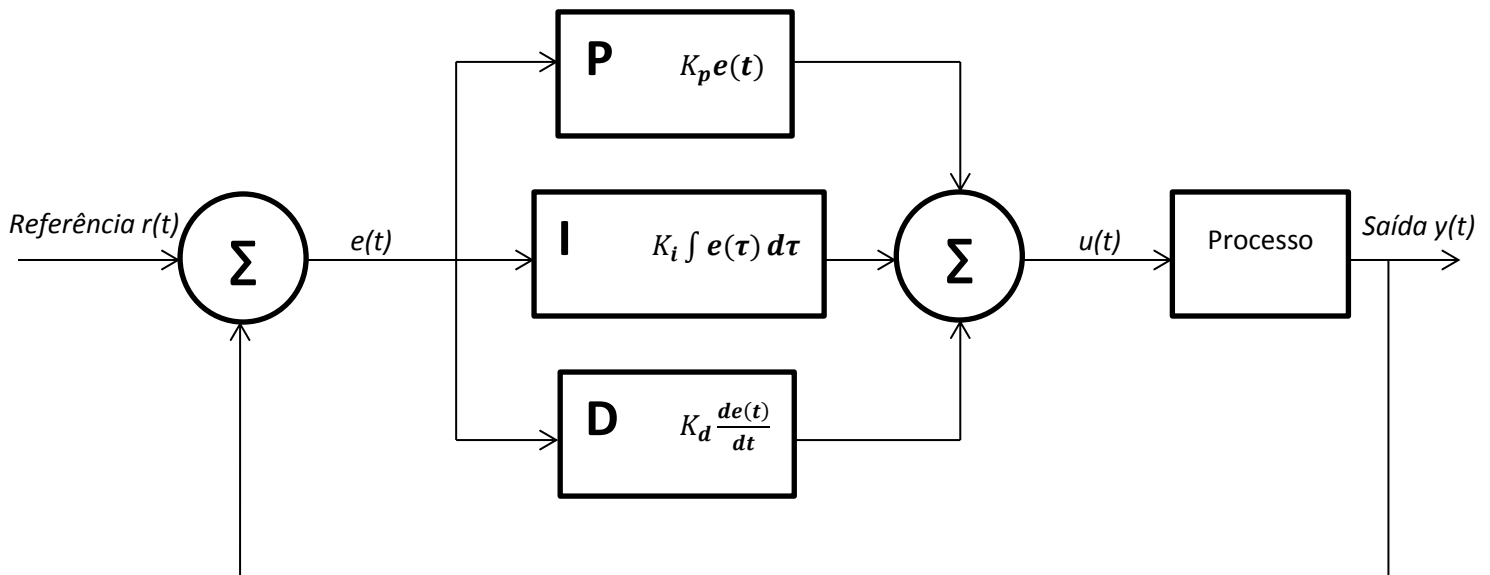


Figura 3.1 - Diagrama de blocos de um sistema com controle PID

Portanto temos que a ação de controle é dada por:

$$u(t) = K_p e(t) + K_i \int e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (3.1)$$

Diante deste esquema geral de um controlador PID, podemos ter controladores derivados desta família, como por exemplo, o controlador P, onde K_i e K_d são iguais a zero; controlador PI, onde K_d é igual a zero; e controlador PD, onde K_i é igual a zero.

O uso da ação derivativa requer cuidado e deve ser evitada em situações em que o erro varie bruscamente, em forma de degrau, uma vez que neste instante a sua derivada será infinita, e em situações em que a medição esteja sujeita a ruídos, já que o comportamento oscilante faz com que a derivada do erro varie continuamente. Estes casos causam um efeito negativo sobre o desempenho do controlador.

Para que se obtenha o melhor desempenho para uma determinada aplicação deve-se ajustar os parâmetros do controlador alterando seu comportamento. De acordo com Ogata: “Um campo interessante da teoria de controle, com muita aplicação prática, é a sintonia do algoritmo PID. Hoje, dispõe-se de um conjunto de regras empíricas e matemáticas que permitem sistematizar a busca de melhores desempenhos através da sintonia de parâmetros.” [3].

3.1 Métodos de sintonia do PID

Vários métodos de ajuste de controladores PID são conhecidos e utilizados na prática de sistemas de controle. Com o intuito de obter um método prático de ajustes, deve-se obter informações a partir de ensaios simples sobre o processo, ao mesmo tempo que estas informações devem ser suficientes para possibilitar um ajuste adequado do controlador.

Por se tratarem de métodos genéricos, muitas vezes os modelos de controladores não são suficientemente completos, podendo ser melhorados com o uso de um ajuste manual fino dos parâmetros.

A tabela (3.1) mostra alguns métodos e cita algumas vantagens e desvantagens.

Tabela 3.1 - Comparativo entre métodos de sintonia PID

Método	Vantagens	Desvantagens
Ajuste manual	Não exige teoria matemática. Método online	Requer certa experiência.
Ziegler-Nichols	Método comprovado. Método online	Aplicação de distúrbio. Exige tentativa e erro. Sintonia agressiva
Ajuste via software	Sintonia consistente. Pode ser online ou off-line. Permite simulação.	Exige um certo custo e treinamento computacional.
Cohen-Coon	Ótimo método de processo	Exige embasamento matemático. Método off-line. Eficaz para processos de primeira ordem.

3.1.1 Método do ajuste manual

O método do ajuste manual é o método mais simples para a sintonia de um controlador PID. É um método que não exige embasamento teórico matemático, sendo necessário apenas a variação dos parâmetros por tentativa e erro a fim de se obter uma resposta satisfatória.

Inicialmente deve-se aumentar o ganho proporcional até que a saída atinja o tempo desejado, mantendo o ganho integrativo e derivativo nulos. Deve-se então aumentar o ganho derivativo para estabilizar o sobre sinal. Por fim deve-se aumentar o ganho integrativo para que a resposta atinja o regime de estado desejado. Deve-se repetir estes passos até que o ganho proporcional seja o maior possível. A tabela (3.2) auxilia na variação dos parâmetros manualmente.

Tabela 3.2 - Efeito da variação do aumento dos parâmetros

Aumento do parâmetro	Tempo de subida	Sobre tensão (overshoot)	Tempo de acomodação	Erro de regime de estado	Estabilidade
Kp	Diminui	Aumenta	Pequena variação	Diminui	Diminui
Ki	Diminui	Aumenta	Aumenta	Diminui muito	Diminui
Kd	Diminui pouco	Diminui pouco	Diminui pouco	Não altera	Aumenta

3.1.2 Método Ziegler-Nichols

Os métodos Ziegler-Nichols são considerados clássicos e aplicados até hoje em sua forma original ou com algumas modificações. Temos dois métodos básicos de ajuste, que diferem a respeito da natureza da informação sobre a dinâmica do processo.

3.1.2.1 Método da resposta ao degrau

Este método se limita a sistemas com resposta estável a entrada degrau. Ele faz análise da resposta do processo a um degrau de referência, tomando como base duas constantes, T e L, obtidas no eixo de tempo através da reta tangente ao ponto de inflexão da curva de saída, e a constante de estabilização do sistema, K. A figura (3.2) nos mostra as constantes T, L e K.

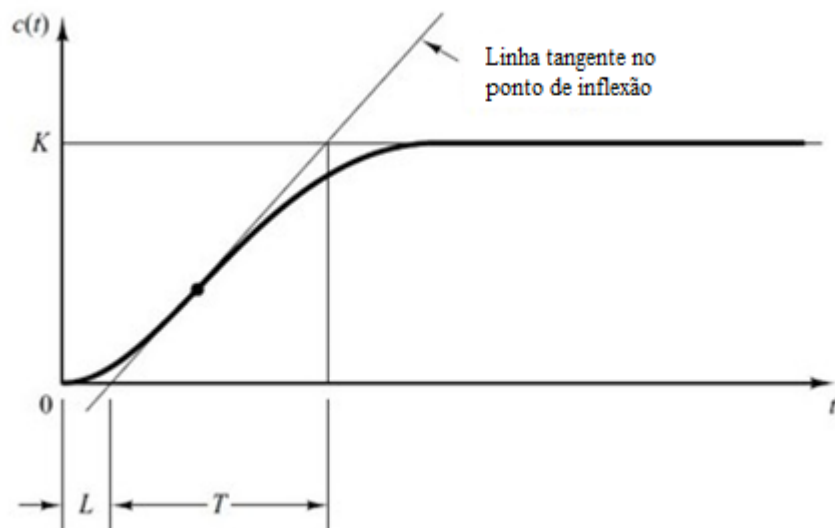


Figura 3.2 - Curva de resposta ao degrau e parâmetros

Com a determinação destas três constantes, através da tabela (3.3) conseguimos calcular facilmente os parâmetros do controlador.

Tabela 3.3 - Parâmetros obtidos com o método resposta ao degrau - Ziegler-Nichols

Tipo de controlador	K_p	K_i	K_d
P	$\frac{T}{KL}$	-	-
PI	$0,9 \frac{T}{KL}$	$3L$	-
PID	$1,2 \frac{T}{KL}$	$2L$	$0,5L$

3.1.2.2 Método da oscilação crítica

Este método aplica-se a sistemas instáveis ou oscilantes. Em malha fechada, deve-se aumentar o ganho K_p gradativamente até que o sistema entre em oscilação permanente, nesta condição, K_p corresponde ao ganho crítico, K_{cr} . Determina-se também o período crítico, P_{cr} . Ambos demonstrados na figura (3.3). Com estes parâmetros, calcula-se através da tabela (3.4) os valores dos parâmetros do controlador.

Tabela 3.4 - Parâmetros obtidos com o método da oscilação crítica – Ziegler-Nichols

Tipo de controlador	K_p	T_i	T_d
P	$0,5 K_{cr}$	-	-
PI	$0,45 K_{cr}$	$\frac{P_{cr}}{1,2}$	-
PID	$0,6 K_{cr}$	$0,5 P_{cr}$	$\frac{P_{cr}}{8}$

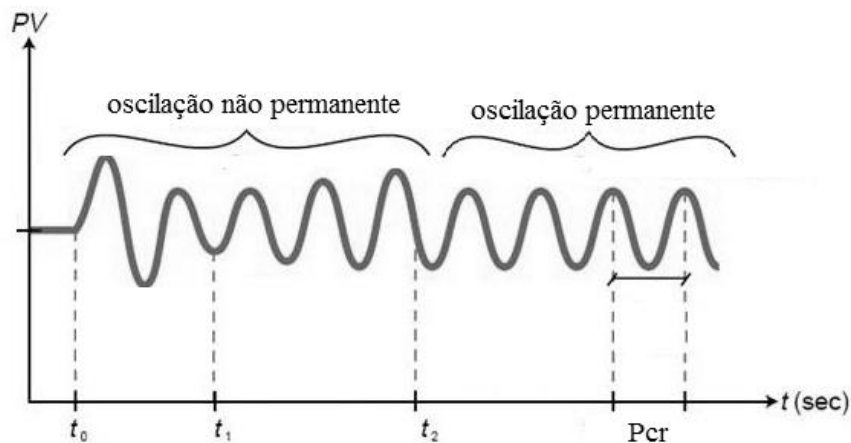


Figura 3.3 - Oscilação crítica

3.1.3 Método de Ajuste via *software* (lugar de raízes)

O *software* MATLAB dispõe de alguns métodos para o projeto de controladores. Um deles é o “root locus”, realizado através do comando `rltool`. Ele utiliza o princípio do método do lugar geométrico das raízes (LGR) que tem como objetivo representar graficamente o deslocamento dos pólos de um sistema em malha fechada devido à alteração de seus parâmetros.

É uma técnica extremamente prática e intuitiva para a sintonia de controladores. Após selecionar o processo a ser controlado, escolhe-se a estrutura de controle. Feito isto, faz-se a edição do controlador, adicionando zeros e pólos, processo demonstrados na figura (3.4).

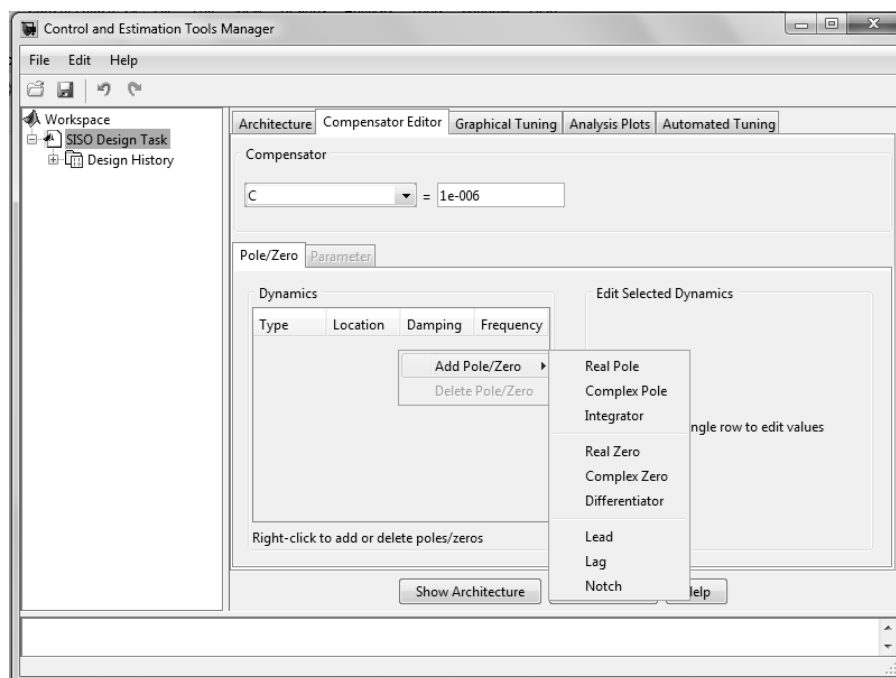


Figura 3.4 - Adição de pólos e zeros do controlador na ferramenta Rttool

Ao adicionar os pólos e zeros integrantes do controlador, alteram-se manualmente os locais de todos os pólos e zeros constituintes do sistema, como na figura (3.5), modificando portanto a sua resposta, que pode ser conferida dentro da própria ferramenta Rltool, como na figura (3.6). Desta forma, consegue-se alcançar as especificações estabelecidas para o funcionamento do controlador.

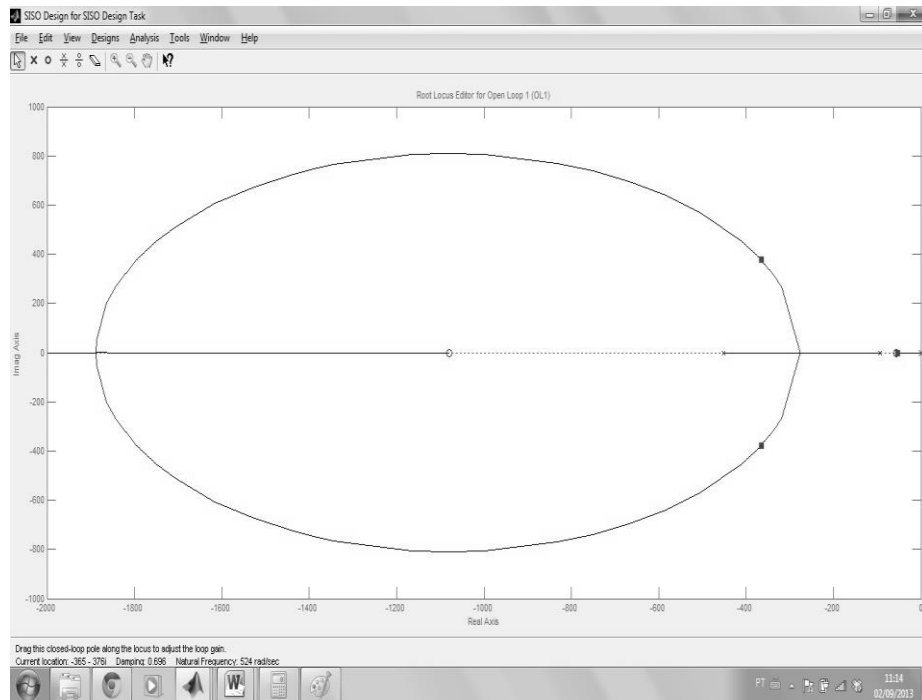


Figura 3.5 - Alteração dos locais dos pólos e zeros do sistema

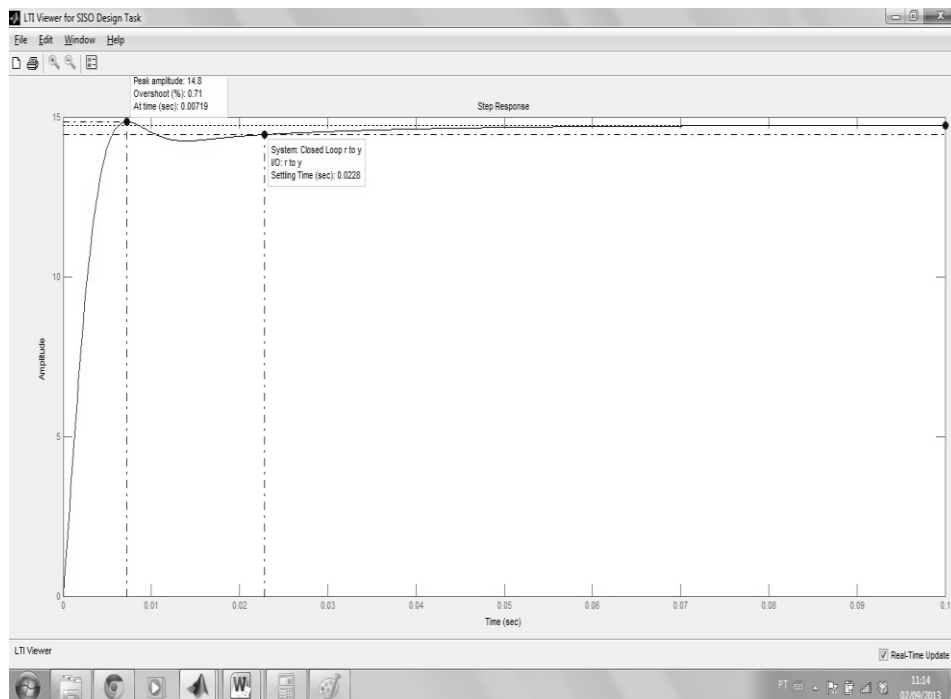


Figura 3.6 - Resposta do sistema de acordo com a variação do local de seus pólos e zeros

4 Controle Adaptativo

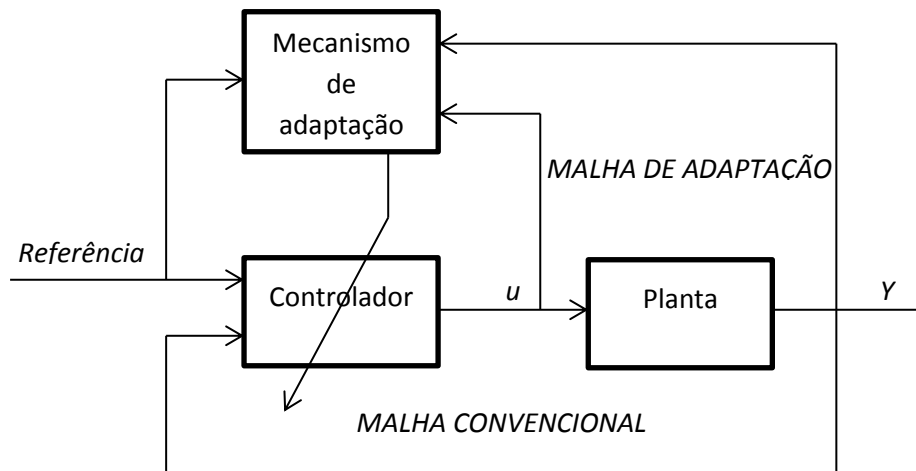


Figura 4.1 - Diagrama de blocos de um sistema com controle adaptativo

A teoria de controle convencional lida predominantemente com aproximações de sistemas lineares que possuem parâmetros constantes. Porém para alguns casos essa aproximação não é satisfatória. Em casos onde as variações são muito amplas, um ganho de realimentação linear e com coeficientes constantes é incapaz de fornecer a flexibilidade necessária para atender às especificações de desempenho. Portanto surge a necessidade de um controle não linear.

Outra questão que merece destaque é o fato de muitas aplicações serem tão complexas que os princípios clássicos de controle são insuficientes, sendo necessário um controle mais inteligente e eficiente.

O controle adaptativo é um tipo de controle não linear. É uma técnica na qual o comportamento do sistema de controle é modificado de acordo com as variações do processo e dos distúrbios presentes. A figura (4.1) nos dá o exemplo de um sistema de controle adaptativo. De acordo com Karl Aström e Bjorn Wittenmark, “Um sistema adaptativo é qualquer sistema projetado sobre o conceito de adaptação. Desta forma, um controlador adaptativo é aquele capaz de modificar sua resposta em função da dinâmica dos parâmetros do processo e distúrbios” [4].

Os parâmetros podem variar devido à não linearidades de atuadores, mudanças de condições de operação do sistema, e distúrbios não estacionários que agem no processo.

Basicamente, todo sistema de controle adaptativo realiza três etapas:

- Identificação da resposta do sistema;
- Comparação com a resposta desejada;

- Regulagem dos parâmetros para obter a resposta desejada;

Sistemas com controle adaptativo possuem duas malhas de realimentação, a malha de controle convencional, que é rápida, e a malha de adaptação dos parâmetros do controlador, bem mais lenta e que introduz a não linearidade ao controle.

Atualmente a teoria de controle adaptativo está em franco desenvolvimento, onde novas técnicas surgem a cada dia. Antes do advento dos microprocessadores, sua utilização era bem restrita, mas após tal invento sua aplicação aumentou drasticamente, sendo muito utilizado em controles industriais, aeroespaciais, robóticos, entre outros. Podemos citar alguns dos métodos de sistemas adaptativos mais comuns:

- Escalonamento de ganho;
- Controlador auto-sintonizável (STR);
- Sistema adaptativo por modelo de referência (MRAS);

4.1 Escalonamento de ganho

O método do escalonamento de ganho consiste em variáveis auxiliares independentes que relacionam as alterações da dinâmica do processo. A figura (4.2) demonstra o esquema básico de um sistema adaptativo por escalonamento de ganho.

É assim chamado por alterar os ganhos de realimentação. O controlador já possui uma definição prévia das regiões de operação do processo e seu algoritmo seleciona a resposta que melhor se enquadra naquela condição específica.

Por ser adaptado em malha aberta, alguns não consideram ser um sistema de controle adaptativo, porém é inegável que esta técnica reduz os efeitos da variação dos parâmetros selecionando a melhor resposta.

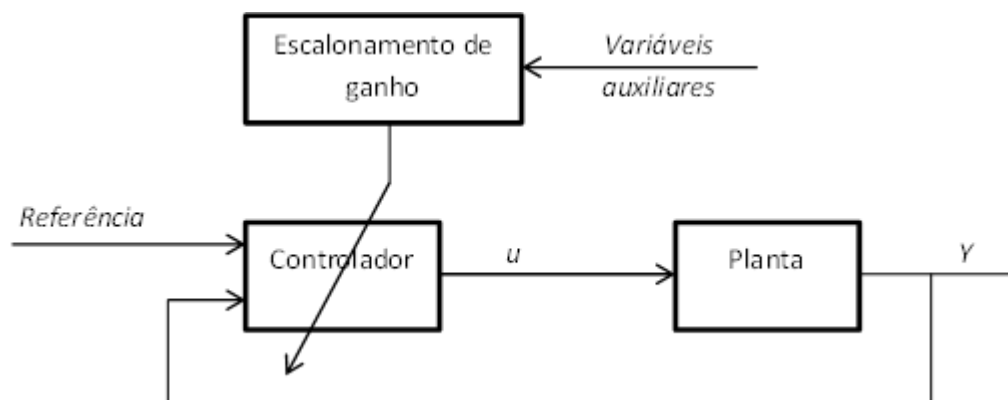


Figura 4.2 - Diagrama de blocos de um sistema com escalonamento de ganho

4.2 Controlador auto-sintonizável

O método do controlador auto-sintonizável (STR) é feito de forma *on-line*, onde os parâmetros são estimados e então é projetado um novo controlador. Essa estimação se dá a cada período amostrado.

Para a estimação, várias estruturas e técnicas podem ser utilizadas, dentre elas, mínimos quadrados, mínimos quadrados estendido e generalizado e filtro estendido de Kalman.

A figura (4.3) demonstra o esquema básico de um controlador auto-sintonizável.

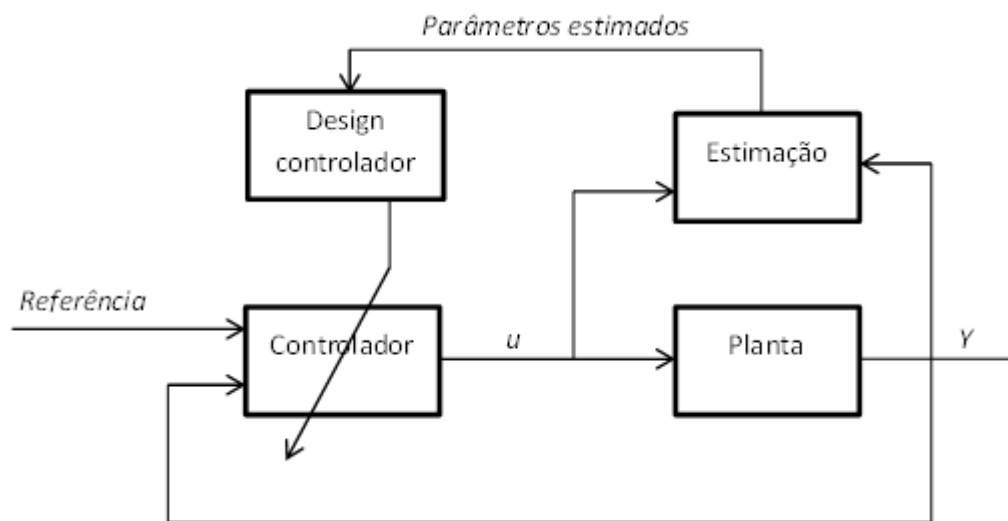


Figura 4.3 - Diagrama de blocos de um sistema com controlador auto-sintonizável

4.3 Sistema adaptativo por modelo de referência

O sistema adaptativo por modelo de referência (MRAS) é uma das técnicas mais utilizadas em controle adaptativo. Este método foi inicialmente proposto por Whitaker em 1958, visando resolver problemas em que as especificações eram dadas em função de um modelo de referência, que respondia idealmente ao sinal de entrada.

Com o controle proposto por Whitaker, duas novas ideias foram introduzidas. Primeiro, o desempenho do sistema é especificado por um modelo, e segundo, os parâmetros do controlador são ajustados com base no erro entre a resposta do modelo de referência e a resposta do processo. Este sistema pode ser também implementado em sistemas de tempo discreto.

Os parâmetros do controlador são ajustados de acordo com a diferença entre a saída do modelo de referência e a saída do sistema, que corresponde ao erro. O objetivo é minimizar o erro, tentando igualá-lo a zero. Este ajuste fará com que a função transferência de malha fechada do processo seja a mais próxima possível do modelo de referência. Isto é chamado de modelo

seguidor (*model-following*). A figura (4.4) demonstra o esquema básico de um sistema adaptativo por modelo de referência.

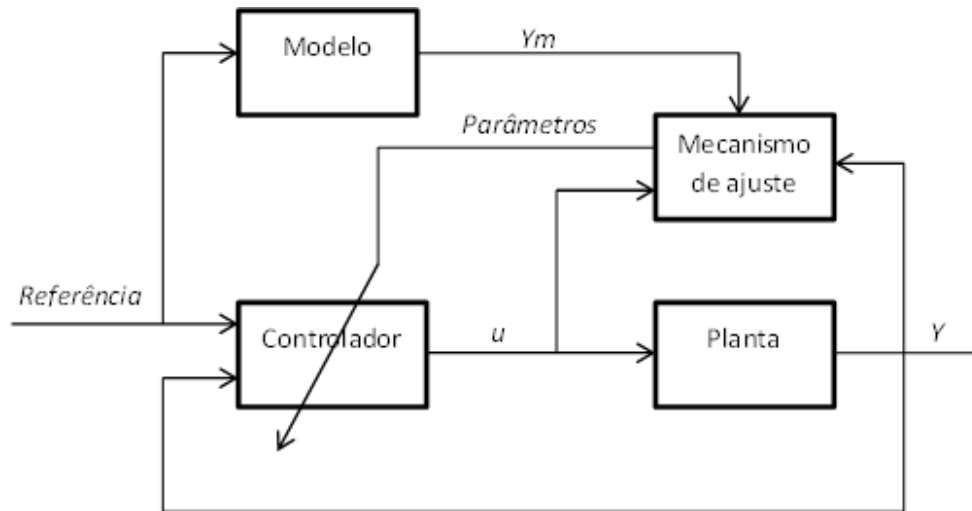


Figura 4.4 - Diagrama de blocos de um sistema adaptativo por modelo de referência (MRAS)

A principal dificuldade desta técnica é obter o mecanismo de adaptação que mantenha o sistema estável e leve o erro à zero. Isto não é trivial e requer alguns mecanismos complexos e extensos de análise e projeto do sistema. Os procedimentos mais utilizados e comentados na literatura são três:

- Método do gradiente;
- Teoria da estabilidade de Lyapunov;
- Teoria da passividade;

4.3.1 Método do gradiente

Este método é regido pelo algoritmo conhecido como regra MIT, que é assim chamada por ter sido desenvolvida na Massachusetts Institute of Technology (MIT). Ele é baseado no pressuposto de que os parâmetros variam mais vagarosamente que as outras variáveis do sistema. Isto admite um tratamento quase estacionário dos parâmetros, o que é essencial para o cálculo das sensibilidades derivadas, as quais são necessárias para o mecanismo de adaptação.

Considerando o erro e entre a resposta do processo e a resposta do modelo de referência, temos:

$$e = y_p - y_m \quad (4.1)$$

De acordo com a regra MIT, introduzimos o critério de função custo $J(\theta)$ em função dos parâmetros θ do controlador. A escolha da função custo é arbitrária, por exemplo:

$$J(\theta) = \frac{1}{2}e^2 \quad (4.2)$$

Devemos fazer com que J seja o menor possível, desta forma, é razoável variar os parâmetros na direção negativa do gradiente de J , ou seja:

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta} \quad (4.3)$$

A equação (4.3) representa o mecanismo de ajuste e é a regra MIT propriamente dita.

Se assumirmos que os parâmetros variam mais lentamente que as outras variáveis do sistema, podemos afirmar que $\frac{\partial e}{\partial \theta}$ pode ser calculado assumindo θ como constante. A derivada $\frac{\partial e}{\partial \theta}$ é dita sensibilidade derivada do erro em relação ao parâmetro θ . Ela caracteriza como o erro se comporta com o ajuste do parâmetro. γ representa a taxa de adaptação (ganho adaptativo).

A regra MIT desempenhará sua função bem se a taxa de adaptação γ for pequena em comparação com a magnitude do sinal de referência e do ganho do processo, porém não é possível fixar um limite para que a estabilidade seja garantida.

Os trabalhos realizados por Gomes, Araújo e Lima em [5] e Stefanello em [6], evidenciam a eficácia, e principalmente a eficiência da utilização do método gradiente para o projeto de sistemas adaptativos por modelo de referência em várias áreas do controle, velocidade de um motor de corrente contínua; sistema de aquecimento para secagem de grãos; projeto de filtros ativos de potência, respectivamente.

Portanto, observa-se que a regra MIT é uma ótima escolha para o ajuste de parâmetros em MRAS.

Pontos que devem ser destacados:

- Não é necessário um modelo perfeito do sistema;
- O método pode ser aplicado para sistemas parcialmente conhecidos;
- O procedimento pode ser usado em sistemas não lineares;
- Aproximações podem ser necessárias para se obter uma lei de ajuste de parâmetros realizável;

O método do gradiente não irá necessariamente resultar em um sistema de malha fechada estável. Deve-se então recorrer a outros métodos para modificar o mecanismo de adaptação, como, por exemplo, a Teoria da estabilidade de Lyapunov e a Teoria da passividade [4].

4.3.2 Ganhos de adaptação

A grande dificuldade em se projetar um sistema adaptativo por modelo de referência se dá em encontrar um bom valor para os ganhos de adaptação γ da regra MIT.

Atualmente, esta ainda é uma área pouco conhecida, tendo poucas informações a respeito de métodos e sendo na maioria das vezes imprecisa.

Rurua prega que o primeiro passo para encontrar um bom valor para γ , é encontrar uma faixa de valores para os quais o sistema se mantém estável [7].

Segundo este pensamento, uma das técnicas existentes é o segundo método de Lyapunov. Nele, deve-se selecionar uma possível função de Lyapunov e verificar para quais valores o sistema se mantém estável. Normalmente, encontra-se uma faixa muito ampla, limitando apenas os ganhos em positivo ou negativo, o que na prática, se mostra ser uma alternativa pouco viável, uma vez que será necessário recorrer à tentativa e erro para encontrar o melhor valor dentro da faixa.

Outra técnica difundida é a utilização dos critérios de erro, através da minimização de índices de desempenho de erro. Vários são os critérios que podem ser utilizados, onde alguns se demonstram mais eficientes que outros, porém os mais utilizados são ISE e ITSE. Estes critérios tendem a minimizar o erro, encontrando, portanto o valor que produz o melhor resultado.

Utilizou-se do trabalho desenvolvido por Caroline Conti em [8] para o desenvolvimento de tais critérios, assim como sua aplicação no *software* Matlab.

No caso de um sistema adaptativo por modelo de referência, queremos encontrar os valores para os ganhos de adaptação que produzam o menor erro entre o modelo de referência e a planta controlada.

O critério ISE e o critério ITSE são definidos como:

$$ISE = \int_0^T e^2(t)dt \quad (4.4)$$

$$ITSE = \int_0^T t * e^2(t)dt \quad (4.5)$$

Em ambos os casos como queremos a minimização do erro, calculamos o ponto mínimo de cada equação.

Podemos estender este conceito para a minimização entre os parâmetros do PID identificados e os parâmetros do PID do modelo de referência.

5 Fuzzy

“A operação destes sistemas inspira-se, em geral, em sistemas biológicos. A capacidade criativa dos seres humanos, de raciocinar de maneira incerta ou difusa contrasta com a forma de operar de computadores e máquinas, regidos por raciocínio binário e preciso. No momento em que estas máquinas transcendessem a esta restrição, tornar-se-iam inteligentes, podendo ‘raciocinar’ de forma difusa. Esta forma de raciocínio é conhecida em inglês por fuzzy, tendo como tradução em português nebuloso, difuso” [9].

5.1 Fundamentos da teoria Fuzzy

A teoria Fuzzy não se assemelha à teoria clássica, ela se aproxima muito da interpretação humana dos fatos, sendo nebulosa, difusa.

Em um conjunto clássico, um elemento pertence a um conjunto ou não. Dado um universo U e um elemento x , o grau de pertinência $\mu_A(x)$ com relação a um conjunto $A \subseteq U$ é dado por:

$$\mu_A(x) = \begin{cases} 1, & \text{se } x \text{ pertence a } A \\ 0, & \text{se } x \text{ não pertence a } A \end{cases}$$

Na teoria Fuzzy proposta por Lotfali Askar Zadeh, temos uma caracterização mais ampla, na medida em que alguns elementos são “mais membros” de um conjunto do que outros. Desta forma, o grau de pertinência pode ser qualquer valor no intervalo entre 0 e 1, onde 0 indica total exclusão e 1 indica completa pertinência. Esta generalização aumenta muito o poder de expressão da função característica e torna algo mais próximo do pensamento humano. Assim temos:

$$\mu_A(x): U \rightarrow [0,1]$$

Temos também na teoria Fuzzy a expressão de conceitos via uso de elementos qualitativos ao invés de valores quantitativos, por exemplo, temos: “muito”, “pouco”, “mais ou menos”, entre outros. A estes conceitos damos o nome de variáveis linguísticas. Portanto, ao invés de assumir instâncias numéricas, variáveis linguísticas assumem instâncias linguísticas.

A principal utilidade das variáveis linguísticas é inserir uma maneira sistemática para a caracterização de fenômenos complexos ou não completamente conhecidos. A fundo, a utilização de variáveis linguísticas, como os seres humanos, e não de variáveis quantificadas, como as máquinas, permite o tratamento de sistemas muito complexos para serem analisados com teorias matemáticas convencionais.

A forma mais comum de expressar o conhecimento da teoria Fuzzy é por meio de regras condição-ação. Um conjunto de condições descrevendo as entradas do processo é associado à uma ação de controle que levará o processo às condições de operação desejadas. Temos como exemplo de condição “erro grande positivo”, e exemplo de ação “aumentar velocidade”.

As regras mais utilizadas em sistemas Fuzzy são as regras do tipo se-então. Elas são frequentemente chamadas de declarações condicionais Fuzzy, ou simplesmente regras Fuzzy.

5.1.1 Funções pertinência

As funções pertinência são curvas que representam o quanto determinado elemento é membro de um conjunto Fuzzy. Várias são as formas das funções pertinência, devendo o projetista analisar a que melhor se encaixa em cada situação. Como exemplo, temos:

- Triangular

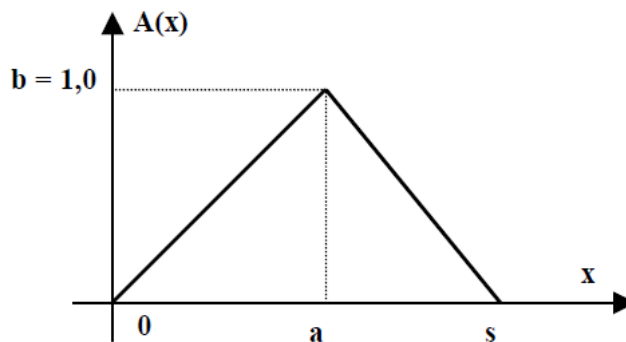


Figura 5.1 - Função de pertinência Triangular

A figura (5.1) demonstra uma função pertinência triangular, e é representada por:

$$\mu_A(x) = \begin{cases} b \frac{1 - |x - a|}{s}, & \text{se } a - s \leq x \leq a + s \\ 0, & \text{caso contrário} \end{cases}$$

- Trapezoidal

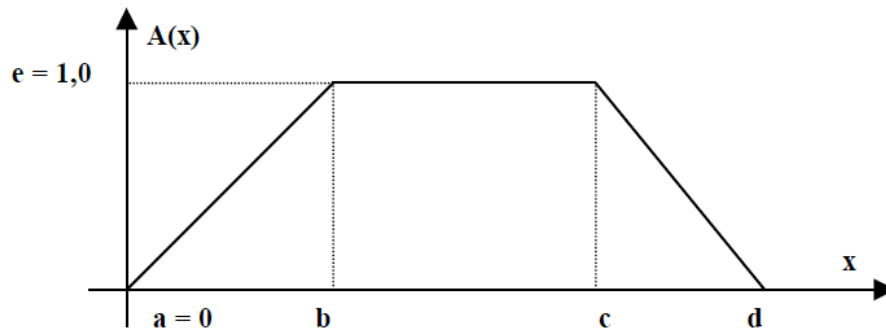


Figura 5.2 - Função pertinência Trapezoidal

A figura (5.2) demonstra uma função pertinência trapezoidal, e é dada por:

$$\mu A(x) = \begin{cases} \frac{(x-a)e}{b-a}, & \text{se } a \leq x \leq b \\ e, & \text{se } b \leq x \leq c \\ \frac{(d-x)e}{d-c}, & \text{se } c \leq x \leq d \\ 0, & \text{caso contrário} \end{cases}$$

- Gaussiana

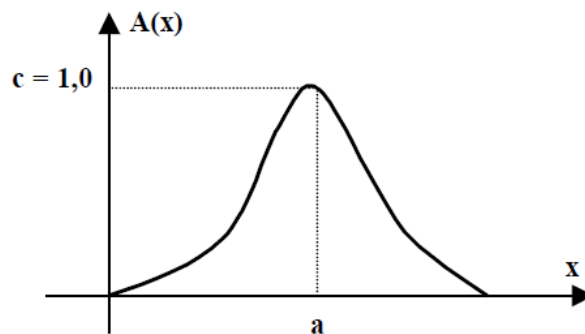


Figura 5.3 - Função pertinência Gaussiana

A figura (5.3) demonstra uma função pertinência gaussiana, e corresponde à:

$$\mu A(x) = c * e^{-\frac{(x-a)^2}{b}}$$

As funções triangulares e trapezoidais são as mais populares devido à simplicidade e ao fato de que o custo computacional adicional exigido pelos outros tipos de função não refletem, em geral, em uma melhoria significativa na qualidade dos valores de saída dos sistemas [10].

5.1.2 Operações Fuzzy

Sejam A e B dois conjuntos Fuzzy no universo U, com funções pertinência μ_A e μ_B , respectivamente. Temos definidas as seguintes operações:

$$\text{União (A} \cup \text{B): } \mu_{A \cup B} = \mu_A \& \mu_B$$

$$\text{Intersecção (A} \cap \text{B): } \mu_{A \cap B} = \mu_A \tau \mu_B$$

$$\text{Complemento (CA): } \mu_{CA} = 1 - \mu_A$$

onde, $\&$ é uma co-norma triangular (norma-s) e τ é uma norma triangular (norma-t).

Uma norma triangular é uma função $\tau: [0,1] \times [0,1] \rightarrow [0,1]$ tal que:

- $x \tau w \leq y \tau z$, se $x \leq y$, $w \leq z$;
- $x \tau y = y \tau x$;
- $(x \tau y) \tau z = x \tau (y \tau z)$;
- $x \tau 0 = 0$ e $x \tau 1 = x$;

Uma co-norma triangular é tal que $\&: [0,1] \times [0,1] \rightarrow [0,1]$, satisfazendo as três primeiras condições acima e ainda:

- $x \& 0 = x$ e $x \& 1 = 1$;

A tabela (5.1) trás exemplos de normas e co-normas, onde os mais utilizados são os tipos Zadeh e Probabilística. As figuras (5.4) e (5.5) demonstram graficamente as normas e as co-normas.

Tabela 5.1 - Principais normas, co-normas e seus nomes

Norma	Co-norma	Nome
$\text{Min}(a,b)$	$\text{Max}(a,b)$	Zadeh
$a.b$	$a \oplus b$	Probabilística
$\text{Max}(a+b-1,0)$	$\text{Min}(a+b,1)$	Lucasiewicz
$\begin{cases} a, \text{ se } b = 1 \\ b, \text{ se } a = 1 \\ 0, \text{ caso contrário} \end{cases}$	$\begin{cases} a, \text{ se } b = 0 \\ b, \text{ se } a = 0 \\ 1, \text{ caso contrário} \end{cases}$	Weber

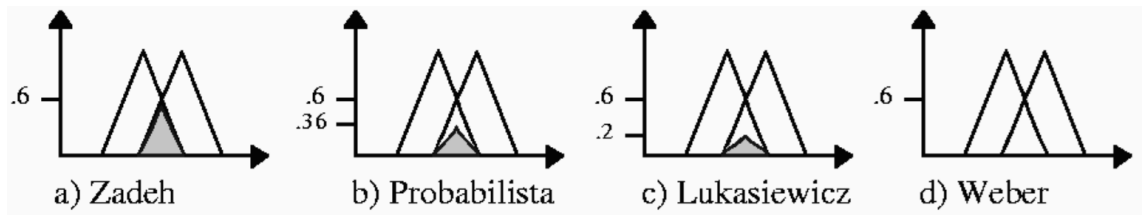


Figura 5.4 - Principais normas

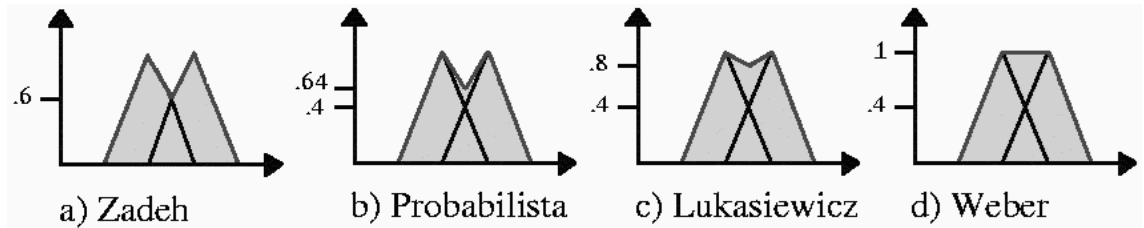


Figura 5.5 - Principais co-normas

5.1.3 Defuzzyficação

A defuzzyficação é o processo de transformar o conjunto nebuloso decorrente das regras fuzzy em um valor numérico para a saída. Existem vários métodos de defuzzyficação, porém os principais são:

- Centro de gravidade (COG):

Seu valor é calculado pela abscissa do centro de gravidade do conjunto fuzzy, como se segue:

$$u_{COG} = \frac{\int \mu_C(x_i)x_i}{\int \mu_C(x_i)}$$

onde x_i é um ponto do universo de conclusão ($i=1,2,\dots$) e $\mu_C(x_i)$ é o valor resultante da conclusão.

- Centro da área (BOA):

Seu valor é calculado pela abscissa da linha vertical que divide a área resultante em duas áreas iguais:

$$u_{BOA} = \left| \int_{i=1}^j \mu_C(x_i) - \int_{i=j+1}^{i_{max}} \mu_C(x_i) \right|, i < j < i_{max}$$

Este método é computacionalmente complexo e custoso.

- Média dos máximos (MOM):

Como o próprio nome diz, o valor é calculado pela média de todos os valores que tenham o maior grau de pertinência:

$$u_{MOM} = \frac{\int x_i}{I}$$

onde I corresponde ao intervalo que contem todos os valores que tenham o maior grau de pertinência.

- Menor dos máximos (SOM):

Seu valor é calculado pelo menor valor que apresente o maior grau de pertinência:

$$u_{SOM} = x_{min}(I)$$

- Maior dos máximos (LOM):

Seu valor é calculado pelo maior valor que apresente o maior grau de pertinência:

$$u_{LOM} = x_{max}(I)$$

5.2 Controle Fuzzy

O uso de controladores baseados na lógica Fuzzy proporciona a possibilidade de supervisão inteligente usando apenas informações qualitativas sobre a operação do sistema, não havendo necessidade de modelagem matemática. Ela propõe uma análise diferente da análise proposta pela lógica clássica.

Esta técnica foi proposta por Zadeh em 1965 e permite inferências intermediárias entre o falso (zero) e o verdadeiro (um). Zadeh analisou que, frequentemente, alguns objetos não podem ser definidos segundo um critério.

Por exemplo, a classe dos animais diferencia claramente os cachorros, peixes e pássaros. Porém, existem características que possuem ambiguidades quando se trata de diferenciação, como a altura de uma pessoa. Ela pode ser muito baixa, baixa, média, alta, muito alta e outras definições. Nestes casos, na maioria das vezes, não se consegue classificar a pessoa como somente alta ou baixa, havendo a necessidade de uma graduação. Segundo Zadeh, entre a certeza de ser e a certeza de não ser, existem infinitos graus de incerteza [11].

O controle Fuzzy, ou controle nebuloso, tem sido utilizado como uma alternativa muito interessante aos métodos clássicos no controle de sistemas não lineares, com parâmetros variáveis no tempo.

A ideia principal deste método é modelar as ações a partir de conhecimento especialista, e não modelar o processo em si. Desta forma, a abordagem é diferente das técnicas clássicas de controle, onde os mesmos são desenvolvidos matematicamente, de modo que suas ações de controle sejam função do estado do processo.

A motivação para esta abordagem diferente veio de casos nos quais o conhecimento especialista de controle era presente, mas os modelos matemáticos envolvidos eram complexos demais ou não conhecidos totalmente.

Várias são as vantagens da utilização do controle Fuzzy, dentre elas:

- Não é necessário modelo matemático do sistema para criar o controle;
- O mecanismo de ajuste pode modificar a característica do controlador Fuzzy no caso de perturbações e mudanças na dinâmica do processo;
- Conhecimento e experiência humana podem ser implantados utilizando regras linguísticas;
- Plantas não lineares podem ser controladas;
- Pode ser utilizado em processos rápidos;

Apesar de sua crescente utilização e sucesso, temos ainda algumas desvantagens:

- Funções pertinências são complexas para se determinar;
- Métodos de defuzzificação e de inferência são complexos de forma analítica;
- Variações de grande amplitude nos parâmetros, alterações estruturais ou fortes perturbações ambientais podem não ser revertidas pelo controlador, apresentando desempenho inadequado;
- Em operações complexas, fuzzyficação e defuzzyficação levam um grande tempo;

O uso de uma grande faixa de operação no controle de um processo não linear pode levar o sistema a apresentar um desempenho indesejável devido à dinâmica variável da planta com controlador Fuzzy fixo. Ainda, as regras obtidas podem não ser válidas para todos os pontos de operação da faixa desejável [12], [13] e [14].

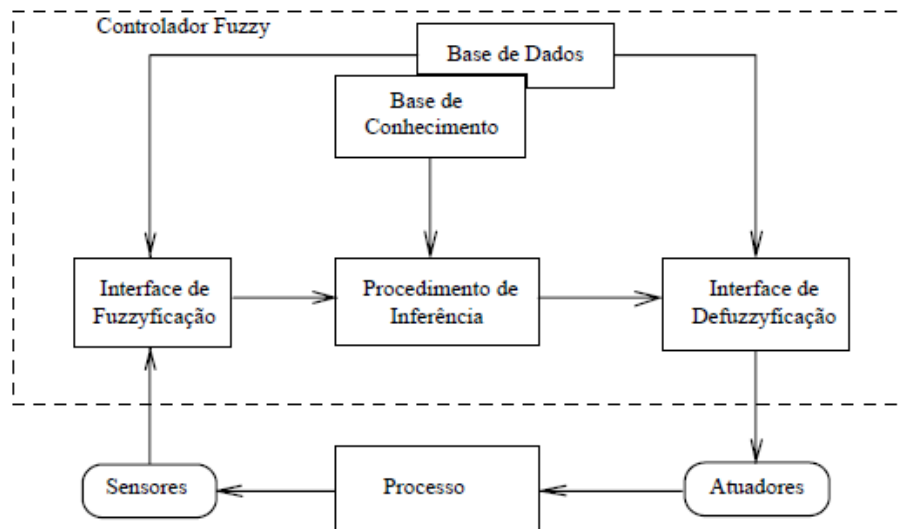


Figura 5.6 - Estrutura de um controlador fuzzy

Na figura (5.6) temos um esquema da estrutura de um controlador Fuzzy. Vale ressaltar seus componentes básicos: Interface de fuzzyficação, base de conhecimento, base de dados, procedimento de inferência e a interface de defuzzyficação.

A interface de fuzzyficação tem como função receber os valores das variáveis de entrada, condicionar estes valores a universos de discurso normalizado e fuzzyficar os valores, ou seja, transformar determinado valor em determinado conjunto fuzzy, representado por uma variável linguística.

A base de conhecimento é composta por uma base de regras, caracterizando a estratégia de controle e suas metas. Já a base de dados armazena as definições necessárias sobre discretizações e normalizações do universo de discurso, partições fuzzy dos espaços de entrada e saída e as definições das funções pertinência.

O procedimento de inferência processa os dados fuzzy de entrada, junto com as regras, de modo a inferir as ações de controle fuzzy, aplicando o operador de implicação fuzzy e as regras de inferência da lógica fuzzy.

A interface de defuzzyficação transforma as ações de controle fuzzy inferidas em ações de controle não-fuzzy. Compatibilizam-se os valores normalizados vindos do passo anterior com valores dos universos de discurso reais das variáveis.

5.3 Fuzzy no Matlab

O *software* Matlab dispõe de um *toolbox* próprio para a criação de lógicas fuzzy. Esta ferramenta chamada “Fuzzy Logic Toolbox” fornece meios computacionais para o desenvolvimento de controladores Fuzzy e apresenta uma interface gráfica muito intuitiva e amigável. Ao final do projeto, pode-se verificar seu desempenho pela ferramenta Simulink, que é atrelada ao *software* Matlab.

Para acessar o *toolbox*, é necessário apenas digitar “fuzzy” na tela de comando do Matlab. Desta forma, abre-se a o “FIS Editor”, “Fuzzy Inference System Editor”, demonstrado pela figura (5.7).

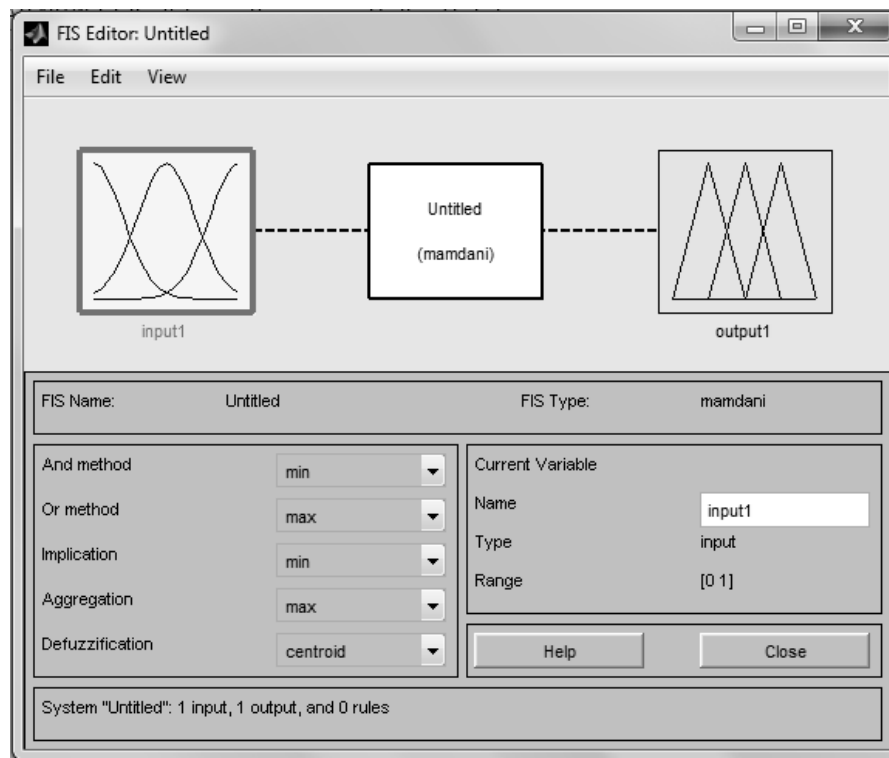


Figura 5.7 - FIS Editor

Nesta interface é possível alterar o tipo de lógica Fuzzy, Mandani ou Sugeno, acrescentar ou retirar variáveis de entrada ou saída, alterar os métodos de inferência utilizados e outras funções.

Ao clicar-se duas vezes sobre a variável de entrada ou de saída, abre-se uma nova interface, a “Membership Function Editor”, demonstrada pela figura (5.8). Nela é possível alterar todos os parâmetros das variáveis, incluindo ou excluindo funções pertinências, definindo seus limites, suas formas, e também alterar os nomes de cada faixa da função.

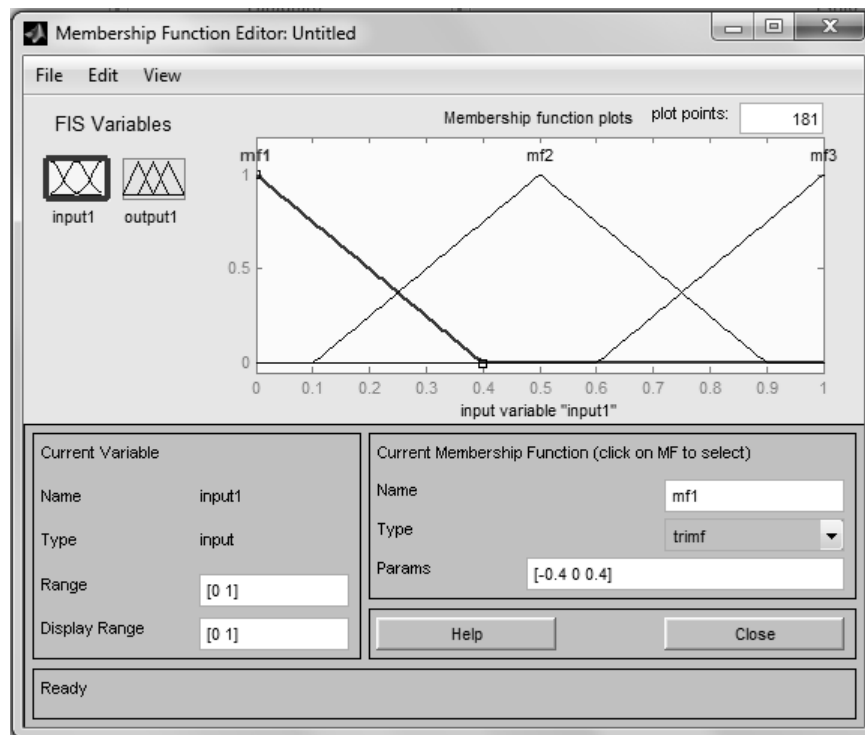


Figura 5.8 - Membership Function Editor

Outra interface a ser conhecida é a “Rule Editor”, demonstrada pela figura (5.9). Ela é acessada com um duplo clique sobre o quadrado branco central da tela “FIS Editor”. Nesta tela deve-se configurar as regras do controlador. Basta seleccionar a combinação das variáveis de entrada, com uma relação AND ou uma relação OR e seleccionar a saída correspondente.

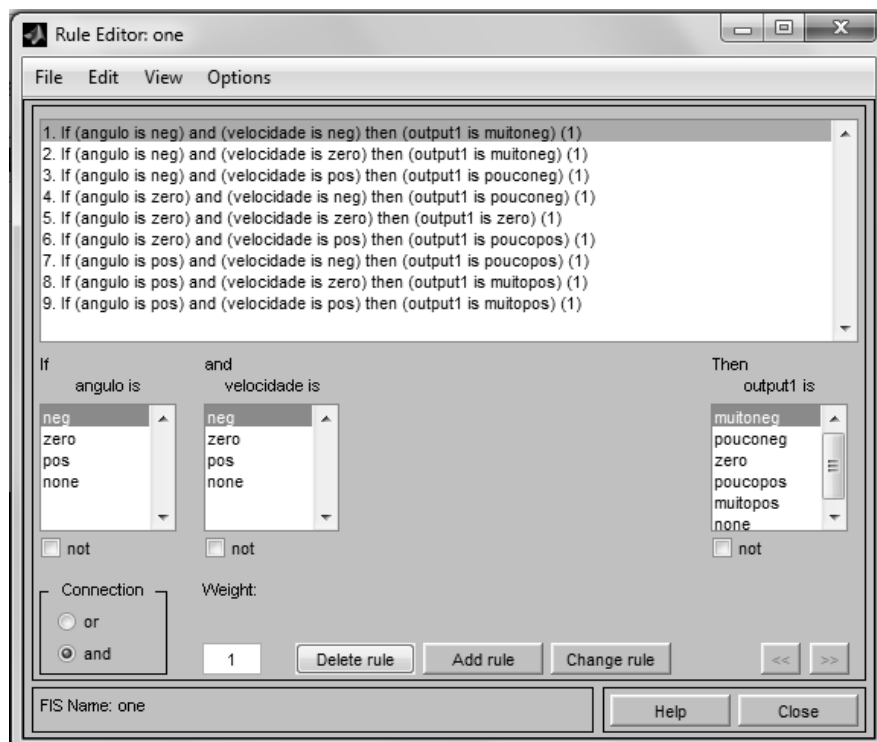


Figura 5.9 - Rule Editor

Após estas etapas pode-se conferir como o controlador Fuzzy irá reagir à determinada entrada. Basta acessar a interface “Rule Viewer”, figura (5.10) ou a interface “Surface Viewer”, figura (5.11), dispostas no menu “options” do “FIS Editor”.

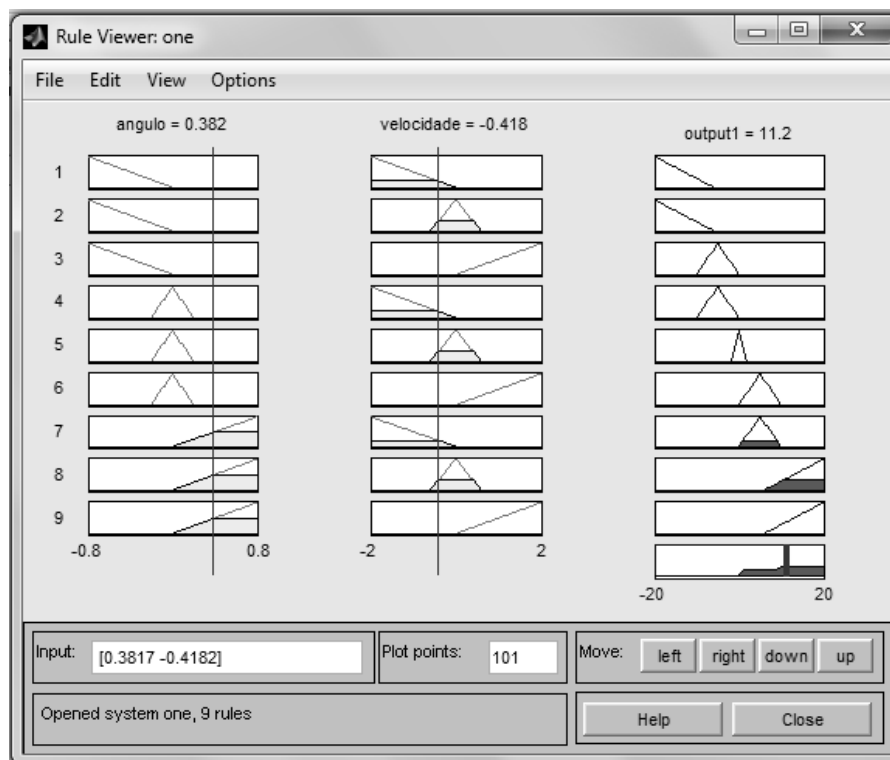


Figura 5.10 - Rule Viewer

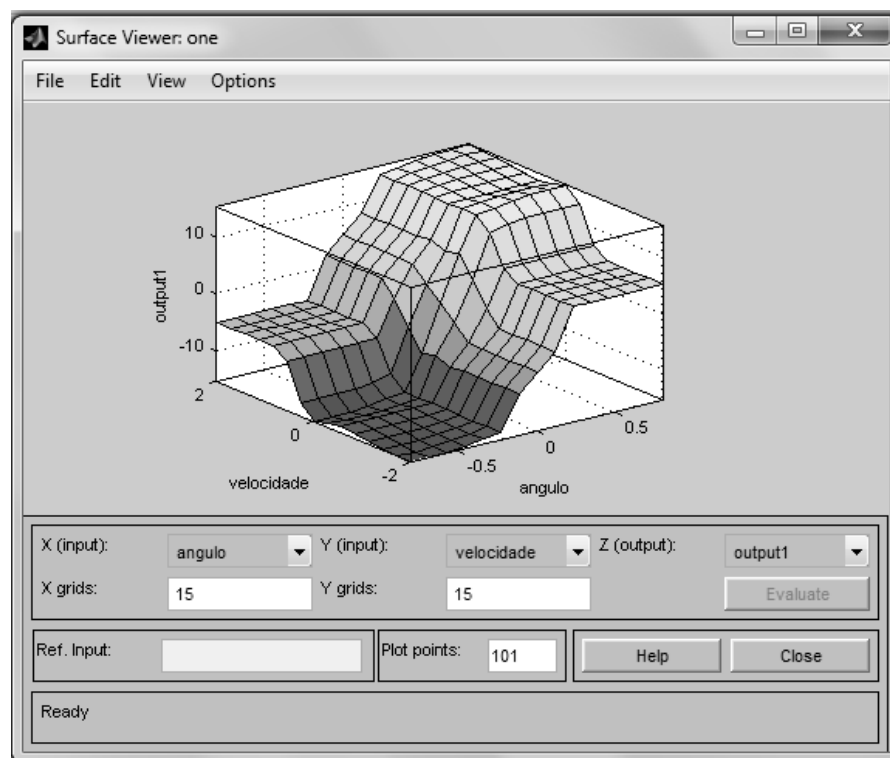


Figura 5.11 - Surface Viewer

6 Controladores

6.1 Controle PID

O controlador PID foi desenvolvido para a aplicação no sistema adaptativo por modelo de referência.

Para o desenvolvimento do controlador PID, utilizou-se a ferramenta presente no *software* Matlab, chamada de “*rltool*”.

O código desenvolvido em Matlab pode ser visto no apêndice deste trabalho.

Inicialmente alterou-se o valor da realimentação, que representa o tacogerador. O tacogerador utilizado é considerado um ganho, uma vez que converte a resposta do sistema, ou seja, a velocidade em rad/s, em tensão. Além dele, utilizamos um conformador de sinal, obtendo assim a tensão exata que corresponde à determinada velocidade.

Em seguida, para que o *rltool* ajustasse o controlador como um PID, acrescentou-se o polo real e os dois zeros reais. Feito isto, passou-se a alterar a posição dos pólos e zeros, modificando a resposta do sistema. Fez-se até que a resposta se encaixasse nas especificações.

O sugerido foi que o sistema com o controlador respondesse com metade do tempo de acomodação da resposta de malha aberta do processo e um sobre sinal máximo de 7%.

Obteve-se, portanto a seguinte função transferência para o PID:

$$G_{PID} = \frac{-1.02557 \cdot 10^{-4} s^2 + 2.392886 s + 231.098297}{s} \quad (6.0)$$

Partiu-se para a implementação de tal controlador no *software* Matlab-Simulink. Seu esquema é representado pela figura (6.1) abaixo.

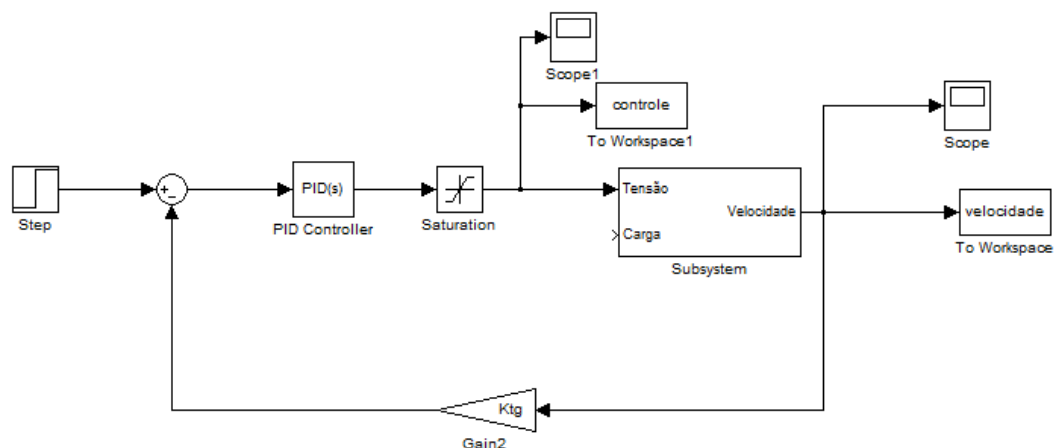


Figura 6.1 - Sistema com controle PID

6.2 Sistema adaptativo por modelo de referência

6.2.1 Mecanismo de adaptação

Como mostrado no capítulo 4, item (4.3.1), utilizaremos o método do gradiente para a determinação do mecanismo de ajuste do modelo de referência, conhecido como regra MIT. Desta forma faremos as seguintes considerações para o erro ε , resposta do processo y_p , resposta do modelo de referência y_m , função de custo J , vetor de parâmetros θ e o ganho de adaptação γ .

$$\varepsilon = y_p - y_m \quad (6.1)$$

$$J(\theta) = \frac{1}{2} \varepsilon^2 \quad (6.2)$$

$$\theta = [K_p, K_i, K_d] \quad (6.3)$$

De acordo com a regra MIT temos:

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta} \quad (6.4)$$

Portanto, aplicá-la significa minimizar a função custo, modificando os parâmetros do controlador em direção ao gradiente negativo de J .

Substituindo os parâmetros na equação (6.4), obtemos:

$$\frac{dK_p}{dt} = -\gamma_p \frac{\partial J}{\partial K_p} = -\gamma_p \left(\frac{\partial J}{\partial \varepsilon} \right) \left(\frac{\partial \varepsilon}{\partial y_p} \right) \left(\frac{\partial y_p}{\partial K_p} \right) \quad (6.5)$$

$$\frac{dK_i}{dt} = -\gamma_i \frac{\partial J}{\partial K_i} = -\gamma_i \left(\frac{\partial J}{\partial \varepsilon} \right) \left(\frac{\partial \varepsilon}{\partial y_p} \right) \left(\frac{\partial y_p}{\partial K_i} \right) \quad (6.6)$$

$$\frac{dK_d}{dt} = -\gamma_d \frac{\partial J}{\partial K_d} = -\gamma_d \left(\frac{\partial J}{\partial \varepsilon} \right) \left(\frac{\partial \varepsilon}{\partial y_p} \right) \left(\frac{\partial y_p}{\partial K_d} \right) \quad (6.7)$$

Temos que:

$$\frac{\partial J}{\partial \varepsilon} = \varepsilon, \text{ pois } J(\theta) = \frac{1}{2} \varepsilon^2 \text{ e } \frac{\partial}{\partial \varepsilon} \left(\frac{1}{2} \varepsilon^2 \right) = 2 * \frac{1}{2} \varepsilon = \varepsilon; \text{ e}$$

$$\frac{\partial \varepsilon}{\partial y_p} = 1, \text{ pois } \varepsilon = y_p - y_m \text{ e } \frac{\partial y_p}{\partial y_p} - \frac{\partial y_m}{\partial y_p} = 1 - 0 = 1$$

Desta forma, as equações (6.5), (6.6) e (6.7) se tornam respectivamente:

$$\frac{dK_p}{dt} = -\gamma_p \frac{\partial J}{\partial K_p} = -\gamma_p * \varepsilon * \left(\frac{\partial y_p}{\partial K_p} \right) \quad (6.8)$$

$$\frac{dK_i}{dt} = -\gamma_i \frac{\partial J}{\partial K_i} = -\gamma_i * \varepsilon * \left(\frac{\partial y_p}{\partial K_i} \right) \quad (6.9)$$

$$\frac{dK_d}{dt} = -\gamma_d \frac{\partial J}{\partial K_d} = -\gamma_d * \varepsilon * \left(\frac{\partial y_p}{\partial K_d} \right) \quad (6.10)$$

Considerando o modelo da planta como uma equação genérica de 2ª ordem, o controlador PID genérico, e o modelo de referência genérico, temos:

$$G_{planta}(s) = \frac{b}{s^2 + a_1 s + a_2} \quad (6.11)$$

$$G_{PID}(s) = K_p + \frac{K_I}{s} + K_D s \quad (6.12)$$

$$G_{modelo\ ref}(s) = \frac{b_{m1}s^2 + b_{m2}s + b_{m3}}{s^3 + a_{m1}s^2 + a_{m2}s + a_{m3}} \quad (6.13)$$

A função transferência em malha fechada, onde R(s) representa o sinal de entrada, se torna:

$$\begin{aligned} Yp(s) &= \frac{G_{planta}(s) * G_{PID}(s)}{1 + G_{planta}(s) * G_{PID}(s)} * R(s) \\ &\rightarrow \frac{\frac{b}{s^2 + a_1 s + a_2} * K_p + \frac{K_I}{s} + K_D s}{1 + \left(\frac{b}{s^2 + a_1 s + a_2} * K_p + \frac{K_I}{s} + K_D s \right)} * R(s) \\ &\rightarrow \frac{\frac{b(K_D s^2 + K_p s + K_I)}{s(s^2 + a_1 s + a_2)}}{\frac{s(s^2 + a_1 s + a_2) + b(K_D s^2 + K_p s + K_I)}{s(s^2 + a_1 s + a_2)}} * R(s) \\ &\rightarrow \frac{b(K_D s^2 + K_p s + K_I)}{s(s^2 + a_1 s + a_2) + b(K_D s^2 + K_p s + K_I)} * R(s) \end{aligned} \quad (6.14)$$

Realizando a decomposição da equação (6.14):

$$\frac{Yp(s)}{R(s)} = \frac{b(K_D s^2 + K_p s + K_I)}{s^3 + (a_1 + bK_D)s^2 + (a_2 + bK_p)s + bK_I} \quad (6.15)$$

Através da transformada inversa de Laplace e a da inserção do símbolo D representando $\frac{d}{dt}$, obtemos em (6.16) a equação (6.15) em função do tempo:

$$\frac{Yp(t)}{R(t)} = \frac{b(K_D D^2 + K_p D + K_I)}{D^3 + (a_1 + bK_D)D^2 + (a_2 + bK_p)D + bK_I} \quad (6.16)$$

Retornando às equações (6.8), (6.9) e (6.10), resta-nos calcular $\frac{\partial y_p}{\partial \theta}$, segue que:

$$\begin{aligned}
& \frac{\partial y_p}{\partial K_p} \\
&= \frac{(D^3 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i) * bD * R(t) - (b(K_D D^2 + K_P D + K_I)R(t)) * bD}{(D^3 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i)^2} \\
&= bD * R(t) - \frac{(b(K_D D^2 + K_P D + K_I)R(t)) * bD}{(D^3 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i)} \\
&= \frac{bD}{(D^3 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i)} * [R(t) - Yp(t)] \quad (6.17)
\end{aligned}$$

$$\begin{aligned}
& \frac{\partial y_p}{\partial K_i} \\
&= \frac{(D^3 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i) * b * R(t) - (b(K_D D^2 + K_P D + K_I)R(t)) * b}{(D^3 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i)^2} \\
&= b * R(t) - \frac{(b(K_D D^2 + K_P D + K_I)R(t)) * b}{(D^3 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i)} \\
&= \frac{b}{(D^3 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i)} * [R(t) - Yp(t)] \quad (6.18)
\end{aligned}$$

$$\begin{aligned}
& \frac{\partial y_p}{\partial K_d} \\
&= \frac{(D^3 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i) * bD^2 * R(t) - (b(K_D D^2 + K_P D + K_I)R(t)) * bD^2}{(D^3 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i)^2} \\
&= bD^2 * R(t) - \frac{(b(K_D D^2 + K_P D + K_I)R(t)) * bD^2}{(D^3 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i)} \\
&= \frac{bD^2}{(D^3 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i)} * [R(t) - Yp(t)] \quad (6.19)
\end{aligned}$$

Substituindo a equação (6.17) em (6.8), (6.18) em (6.9) e (6.19) em (6.10):

$$\frac{dK_p}{dt} = -\gamma_p \frac{\partial J}{\partial K_p} = -\gamma_p * \varepsilon * \frac{bD}{(D^3 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i)} * [R(t) - Yp(t)] \quad (6.20)$$

$$\frac{dK_i}{dt} = -\gamma_i \frac{\partial J}{\partial K_i} = -\gamma_i * \varepsilon * \frac{b}{(D^3 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i)} * [R(t) - Yp(t)] \quad (6.21)$$

$$\frac{dK_d}{dt} = -\gamma_d \frac{\partial J}{\partial K_d} = -\gamma_d * \varepsilon * \frac{bD^2}{(D^3 + (a_1 + bK_d)D^2 + (a_2 + bK_p)D + bK_i)} * [R(t) - Yp(t)] \quad (6.22)$$

O código desenvolvido em Matlab para encontrar a função de transferência do motor pode ser visto no apêndice deste trabalho.

Feitas as deduções, partiu-se para o desenvolvimento do sistema no *software* Matlab-Simulink. O diagrama de blocos utilizado como base para tal desenvolvimento foi o da figura (6.2).

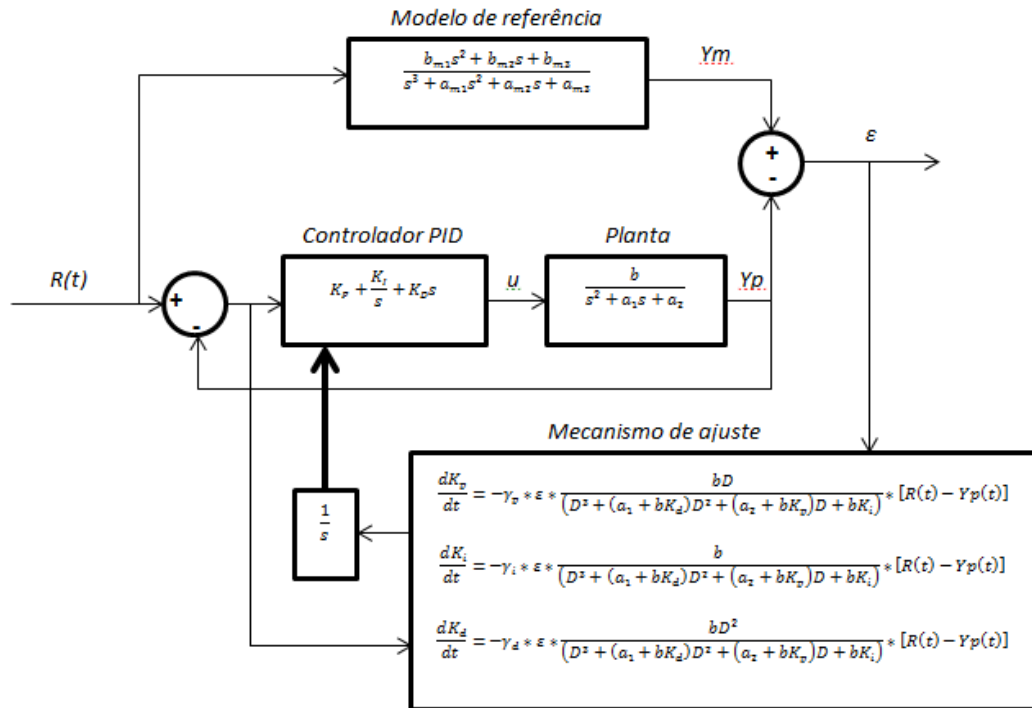


Figura 6.2 - Diagrama de blocos do sistema adaptativo por modelo de referência utilizado

No simulink, temos que as partes do mecanismo de ajuste do parâmetro proporcional, integrativo, derivativo, são respectivamente demonstradas pelas figuras (6.3), (6.4) e (6.5).

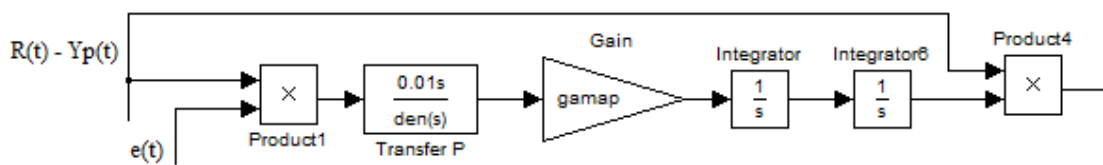


Figura 6.3 - Mecanismo de ajuste proporcional

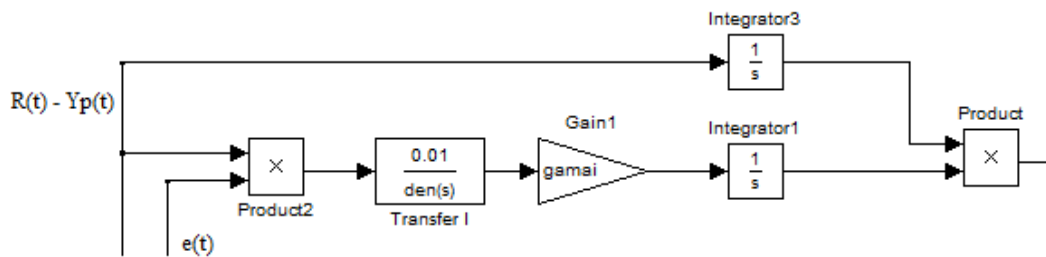


Figura 6.4 - Mecanismo de ajuste integral

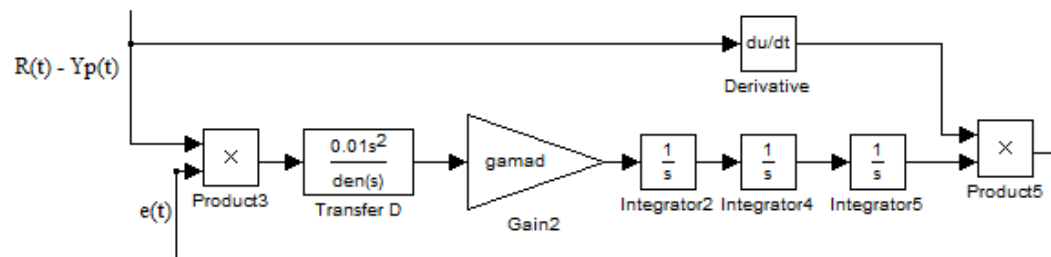


Figura 6.5 - Mecanismo de ajuste derivativo

Como resultado da união destes três mecanismos, juntamente com o modelo de referência e a planta utilizada, temos o sistema adaptativo por modelo de referência demonstrado na figura (6.6).

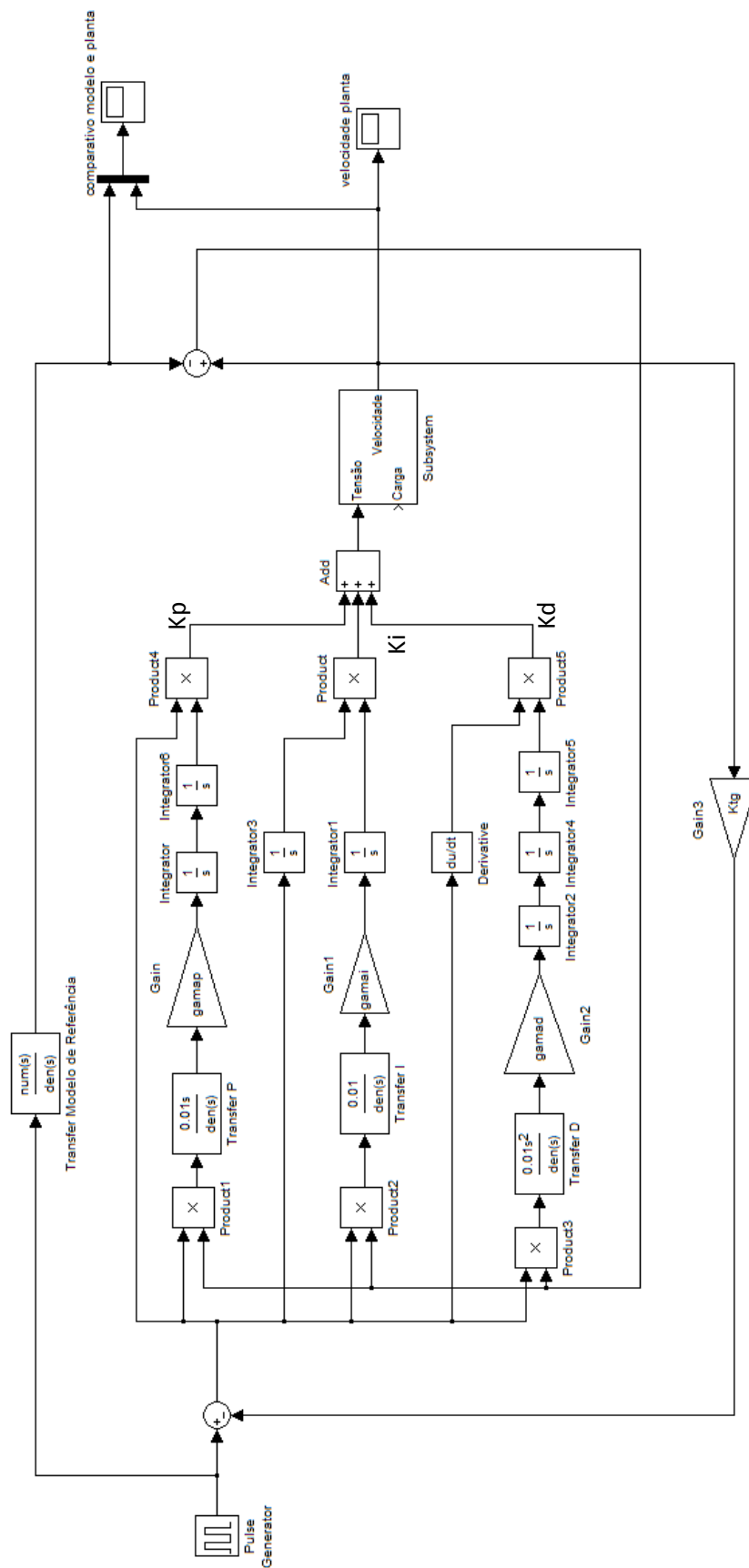


Figura 6.6 - Sistema adaptativo por modelo de referência

6.2.2 Ganhos de adaptação

Como dito anteriormente, a parte mais difícil do projeto de um sistema adaptativo por modelo de referência é encontrar bons valores para os ganhos de adaptação γ_p, γ_i e γ_d .

Com base na teoria dos critérios de minimização de erro, utilizaram-se como base as equações (4.4) e (4.5), fazendo alterações para que o erro fosse composto pelos três parâmetros pertinentes ao sistema, K_p, K_i e K_d .

Considerando K_p', K_i' e K_d' como os parâmetros presentes no modelo de referência, temos que:

$$\min_{\gamma} ISE = \min_{\gamma} \int_0^T [(K_p - K_p')^2 + (K_i - K_i')^2 + (K_d - K_d')^2] dt \quad (6.23)$$

$$\min_{\gamma} ITSE = \min_{\gamma} \int_0^T t * [(K_p - K_p')^2 + (K_i - K_i')^2 + (K_d - K_d')^2] dt \quad (6.24)$$

Utilizou-se o *software* Matlab para realizar a minimização destes critérios, através da função “*fminsearch*”. O código da rotina utilizada para o cálculo é demonstrado no apêndice deste trabalho, assim como o diagrama do Simulink utilizado.

Os valores encontrados para os ganhos de adaptação estão descritos na tabela (6.1).

Tabela 6.1 - Valores encontrados para os ganhos de adaptação

Ganhos de adaptação	ISE	ITSE
γ_p	-99,4583	-96,9402
γ_i	-9646,3516	-9472,2841
γ_d	$-5,2502 * 10^{-4}$	$-5.1476 * 10^{-4}$

6.3 Controle Fuzzy

O modelo Fuzzy foi desenvolvido através do *toolbox* do *software* Matlab para este fim. Desde o início se mostrou bem atraente devido a sua simplicidade e robustez. A principal dificuldade encontrada nesta técnica foi encontrar as funções pertinência de cada variável linguística, assim como os ganhos Kp, Ki e Kd, uma vez pretendia-se encontrar as que melhor respondiam ao processo e suas variações.

Foram utilizadas três variáveis linguísticas, duas de entrada: erro e variação do erro; e uma de saída: ação de controle. Cada uma foi dividida em cinco funções pertinência. As tabelas (6.2), (6.3) e (6.4), assim como as figuras (6.7), (6.8) e (6.9) demonstram as variáveis e suas funções pertinência.

Tabela 6.2 - Funções pertinência da variável Erro

Variável Erro		
Valor linguístico	Notação	Faixa função pertinência
Negativo Grande	NG	$[-7,5 \ -5 \ -2,5]$
Negativo Pequeno	NP	$[-5 \ -2,5 \ 0]$
Zero	Z	$[-2,5 \ 0 \ 2,5]$
Positivo Pequeno	PP	$[0 \ 2,5 \ 5]$
Positivo Grande	PG	$[2,5 \ 5 \ 7,5]$

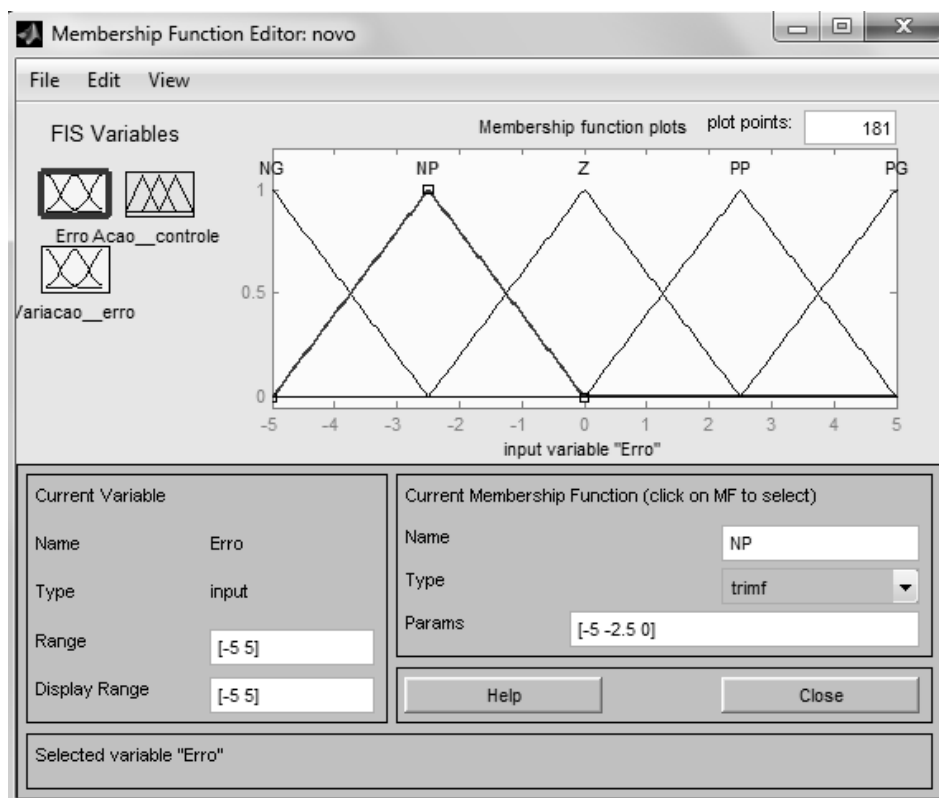


Figura 6.7 - Variável Erro no *toolbox* Fuzzy

Tabela 6.3 - Funções pertinência da variável Variação do Erro

Variável Variação do Erro		
Valor linguístico	Notação	Faixa função pertinência
Negativo Grande	NG	$[-0,015 \ -0,01 \ -0,005]$
Negativo Pequeno	NP	$[-0,01 \ -0,005 \ 0]$
Zero	Z	$[-0,005 \ 0 \ 0,005]$
Positivo Pequeno	PP	$[0 \ 0,005 \ 0,01]$
Positivo Grande	PG	$[0,005 \ 0,01 \ 0,015]$

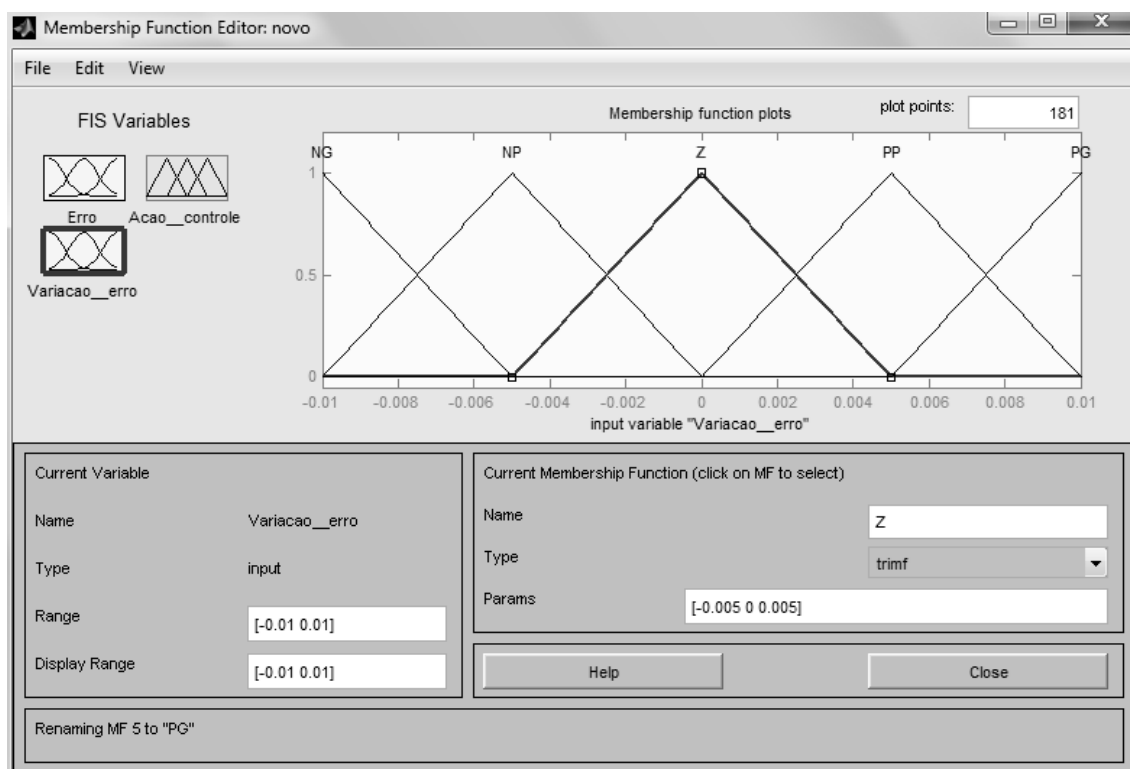


Figura 6.8 - Variável Variação do Erro no toolbox Fuzzy

Tabela 6.4 - Funções pertinência da variável Ação de Controle

Variável Ação de Controle		
Valor linguístico	Notação	Faixa função pertinência
Negativo Grande	NG	$[-6 \ -4 \ -2]$
Negativo Pequeno	NP	$[-4 \ -2 \ 0]$
Zero	Z	$[-2 \ 0 \ 2]$
Positivo Pequeno	PP	$[0 \ 2 \ 4]$
Positivo Grande	PG	$[2 \ 4 \ 6]$

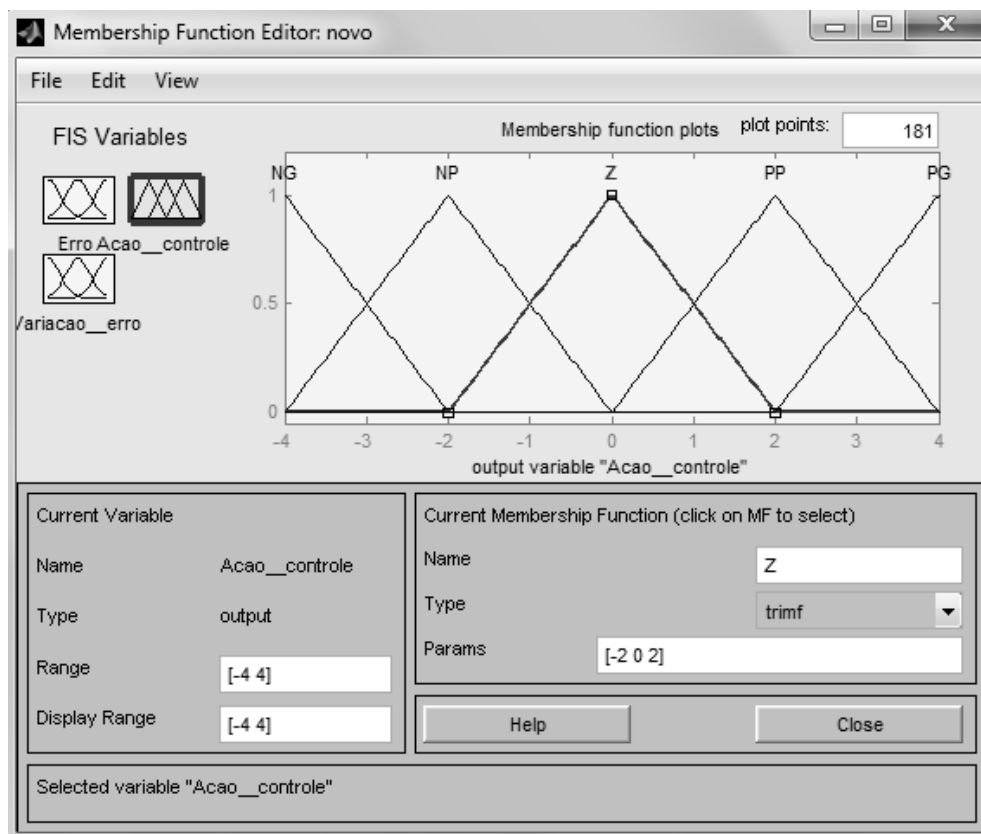


Figura 6.9 - Variável Ação de Controle no *toolbox* Fuzzy

Partiu-se então para a determinação das regras utilizadas para a tomada de decisões na inferência. A ideia principal é de que se a velocidade está baixa, ou seja, se o erro está positivo, deve-se aumentar a ação de controle, se a velocidade está alta, ou seja, se o erro está negativo, deve-se diminuir a ação de controle. O uso da variação do erro é uma forma de antecipar o comportamento do sistema, e assim se antecipar para gerar uma ação de controle mais efetiva.

Como temos cinco valores linguísticos para cada variável de entrada, temos então vinte e cinco combinações de ação de controle, todas mostradas na tabela (6.5).

Tabela 6.5 - Regras do controle Fuzzy

		Variação do Erro				
		NG	NP	Z	PP	PG
Erro	Ação de controle					
	NG	NG	NG	NP	NP	Z
	NP	NG	NP	NP	Z	PP
	Z	NP	NP	Z	PP	PP
	PP	NP	Z	PP	PP	PG
	PG	Z	PP	PP	PG	PG

O esquema montado no simulink para verificar a eficácia do método foi o demonstrado na figura (6.10).

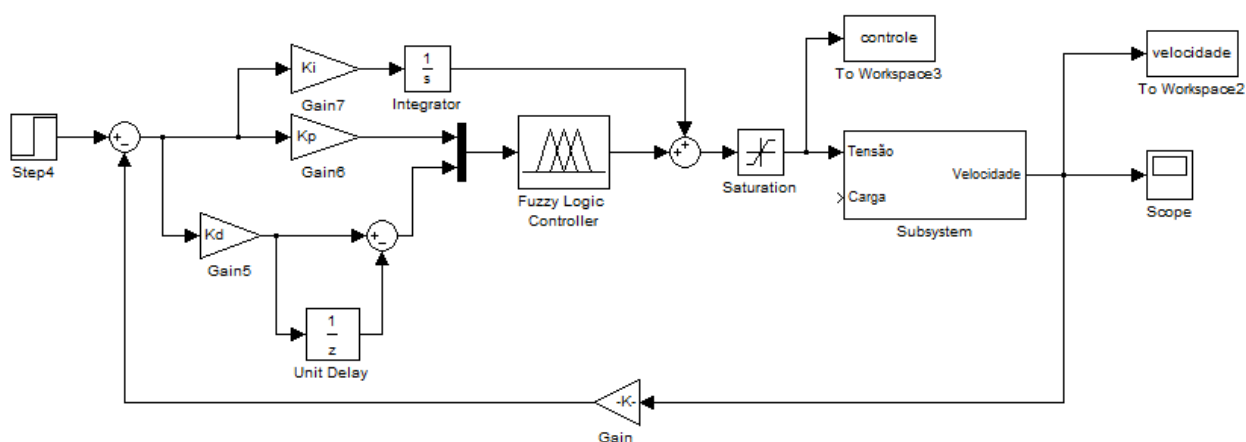


Figura 6.10 - Sistema com controle Fuzzy

Como pode se notar, este esquema é um modelo de controle PID com otimização Fuzzy. Ele conta com uma parte proporcional, uma integrativa e uma derivativa. No bloco Fuzzy temos como entrada o erro e a variação do erro, referente à diferença entre o erro e o erro no instante anterior, causado pelo bloco “*delay*”.

Os parâmetros K_p , K_i e K_d foram calculados seguindo o mesmo princípio para o cálculo dos ganhos de adaptação do item (6.2.2) deste trabalho. O código desenvolvido em Matlab para tal cálculo, assim como o diagrama do Simulink utilizado é demonstrado no apêndice deste trabalho.

Foi feita uma minimização do erro entre o sistema Fuzzy e um modelo de referência, calculando, portanto, os valores de K_p , K_i e K_d que minimizam esta diferença. Assim como para o ganho de adaptação do item (6.2.2) utilizou-se o critério ISE e ITSE, encontrando os valores demonstrados na tabela (6.6).

Tabela 6.6 - Valores para K_p , K_i e K_d encontrados segundo métodos de minimização do erro

	ISE	ITSE
K_p	3,3741	3,3739
K_i	212,1844	211,9189
K_d	$1,8225 * 10^{-3}$	$1,8183 * 10^{-3}$

7 Simulações e resultados

Várias simulações foram feitas via Simulink com o objetivo de verificar a eficiência dos métodos de controle adotados, além de verificar a adaptação dos mecanismos. São elas:

- Degraus consecutivos;
- Trem de pulsos;
- Aplicação de carga;
- Variações paramétricas;

Na simulação degraus consecutivos, inicialmente aplica-se um degrau de 1V e após meio segundo, troca-se por um degrau de 2V. Esta simulação nos dá a eficiência dos métodos em se estabilizar em diferentes *set-points*.

Em trem de pulsos utiliza-se um gerador de pulsos com período de 0,4 segundos, onde metade do período temos amplitude de 1V e a outra metade temos amplitude nula. Esta simulação é muito importante no caso do sistema adaptativo por modelo de referência, com ela conseguimos verificar a ação do mecanismo de ajuste, fazendo com que a resposta acompanhe o modelo de referência. Conseguimos verificar a convergência dos métodos em alcançar a resposta desejada ao longo do tempo.

Com a aplicação de carga, verificamos como o sistema contorna uma situação desfavorável, que tende a tirar o processo de sua condição de operação. Os métodos devem, em um tempo breve, retornarem às condições de operação. Aplica-se uma carga no motor de 0,05 Nm após meio segundo de funcionamento.

Na simulação de variações paramétricas, variamos alguns parâmetros do motor, e verificamos se os métodos conseguem contornar estas situações e manter as condições de operação. Claramente as variações aplicadas nas simulações são muito maiores que as que realmente ocorrem na prática, porém com elas, fica mais evidente o efeito de manutenção das condições de operação.

7.1 Degraus consecutivos

7.1.1 MRAS

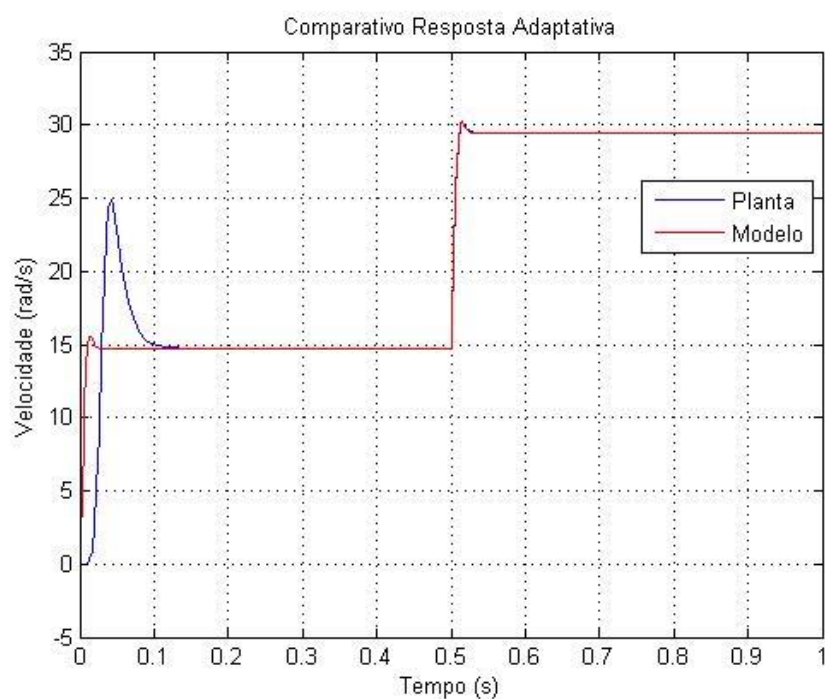


Figura 7.1 - Comparativo entre resposta do modelo de referência e do sistema adaptativo

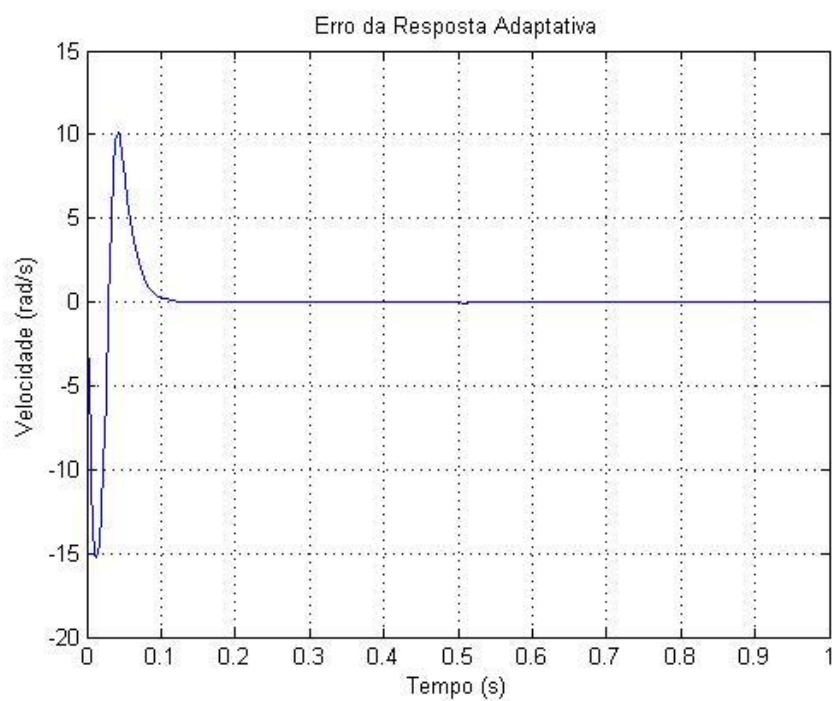


Figura 7.2 - Erro da resposta do sistema adaptativo em relação ao modelo de referência

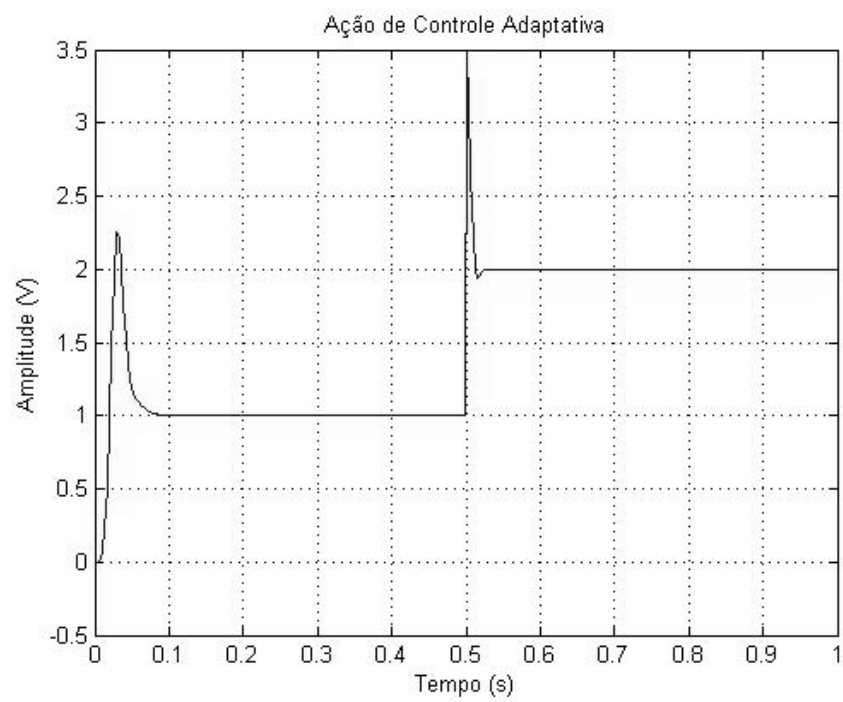


Figura 7.3 - Ação de Controle do sistema adaptativo

7.1.2 Fuzzy

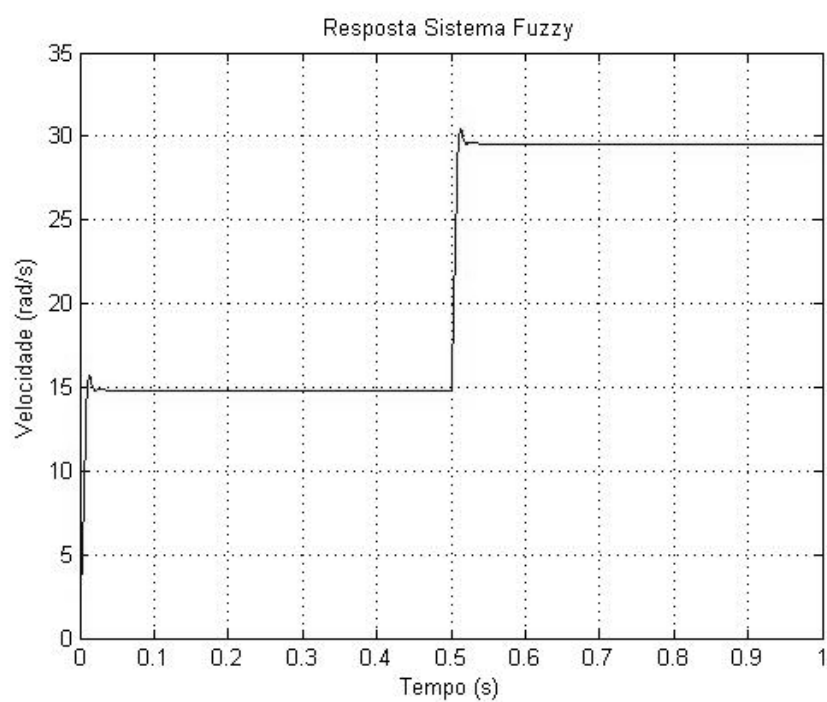


Figura 7.4 - Resposta do sistema fuzzy

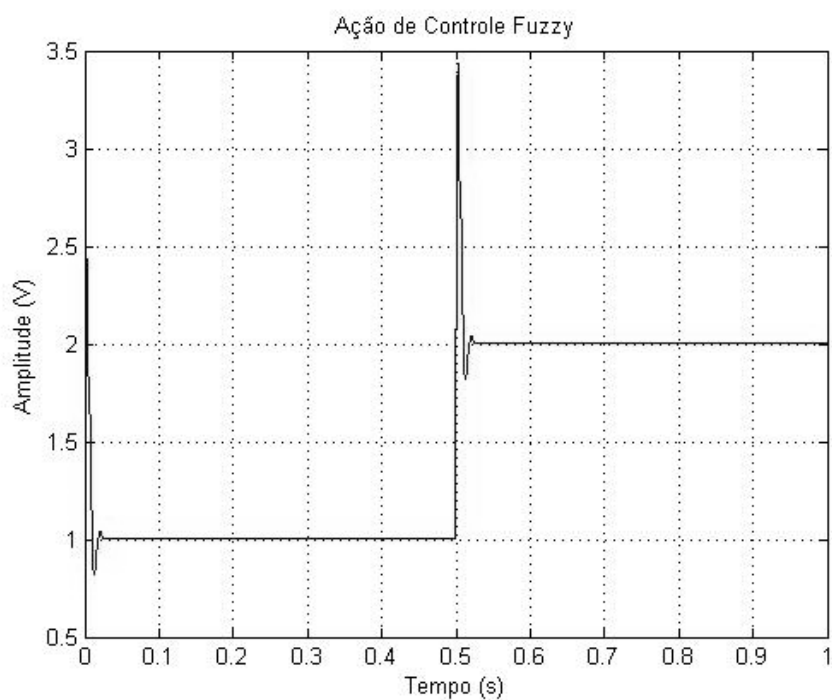


Figura 7.5 - Ação de controle do sistema Fuzzy

7.1.3 Comparativo

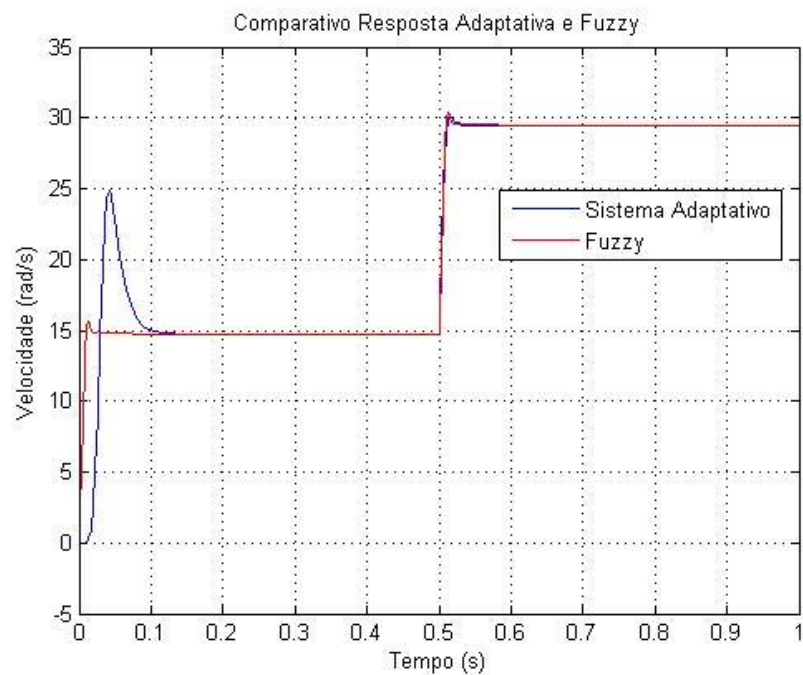


Figura 7.6 - Comparativo entre a resposta do sistema adaptativo e do sistema Fuzzy

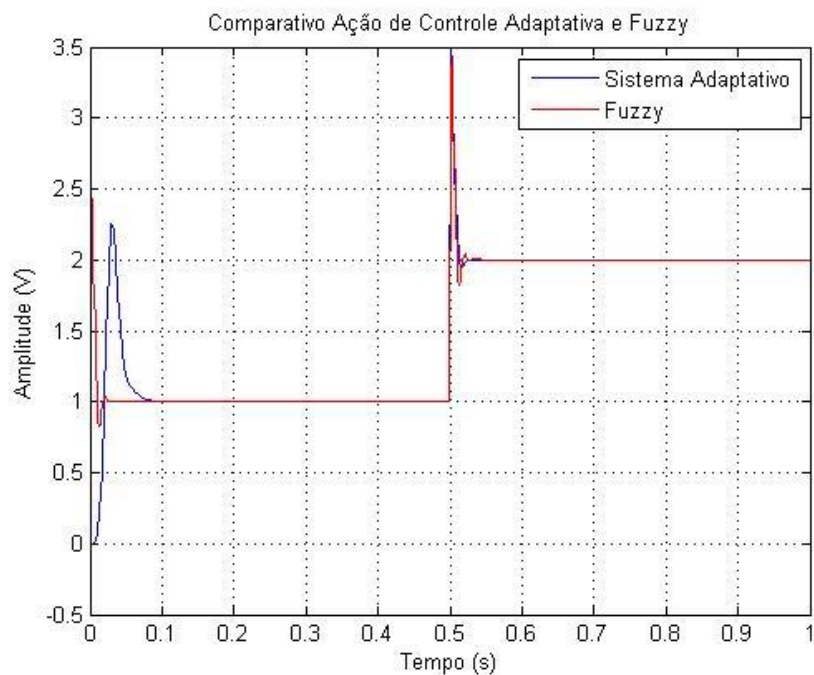


Figura 7.7 - Comparativo entre a ação de controle do sistema adaptativo e do sistema Fuzzy

Verificamos que para este tipo de aplicação, o controle fuzzy no primeiro degrau atingiu o regime mais rápido que o sistema adaptativo, que através do seu mecanismo de ajuste tentava alcançar o "set-point". Porém depois do mecanismo já ajustado, ambos responderam de forma bem semelhante. A ação de controle de cada mecanismo após o ajuste responde de forma semelhante também, uma vez que queremos a mesma resposta.

7.2 Trem de pulsos

7.2.1 MRAS

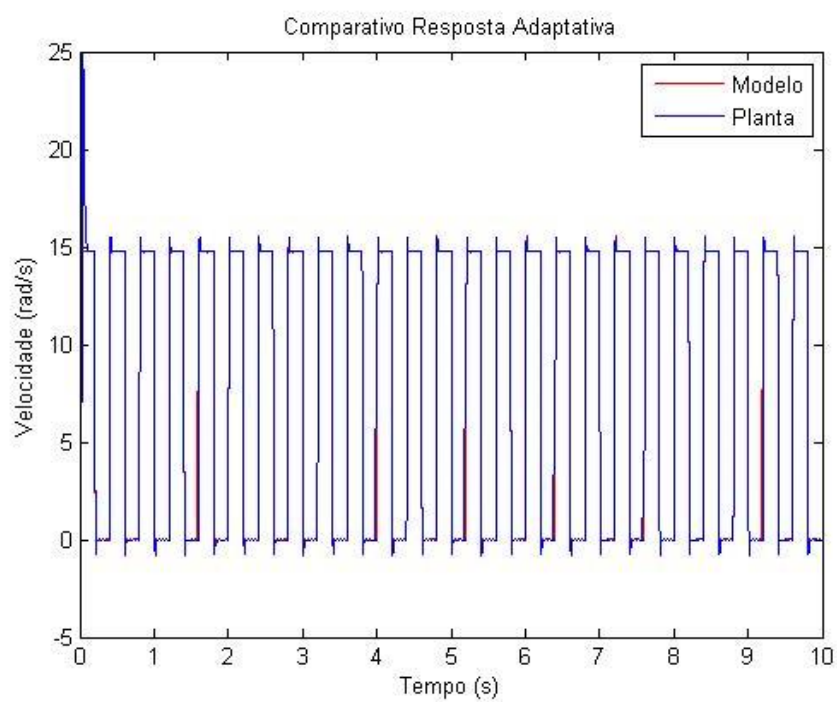


Figura 7.8 - Comparativo entre a resposta do modelo de referência e do sistema adaptativo

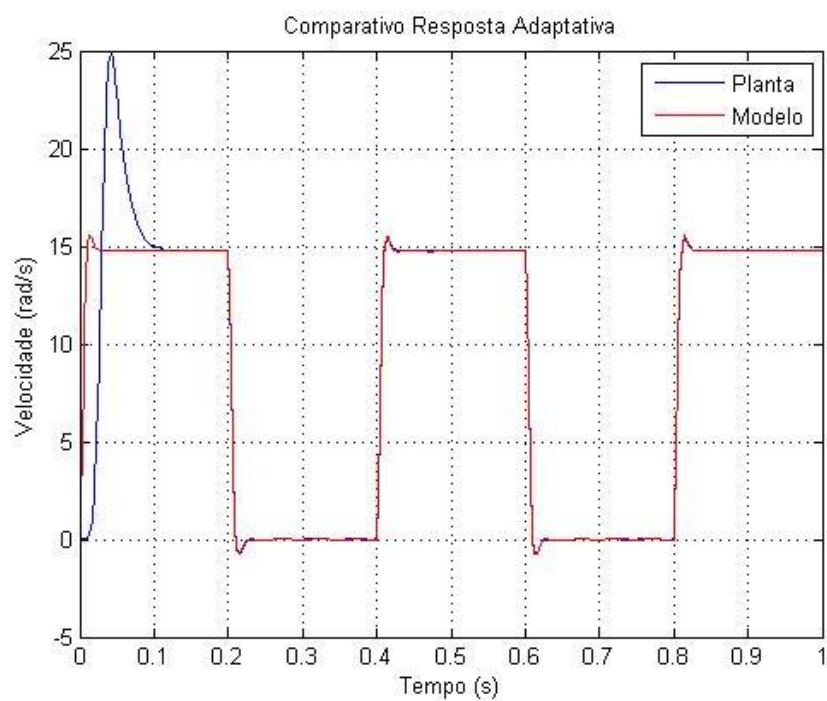


Figura 7.9 - Detalhe do comparativo em seu início

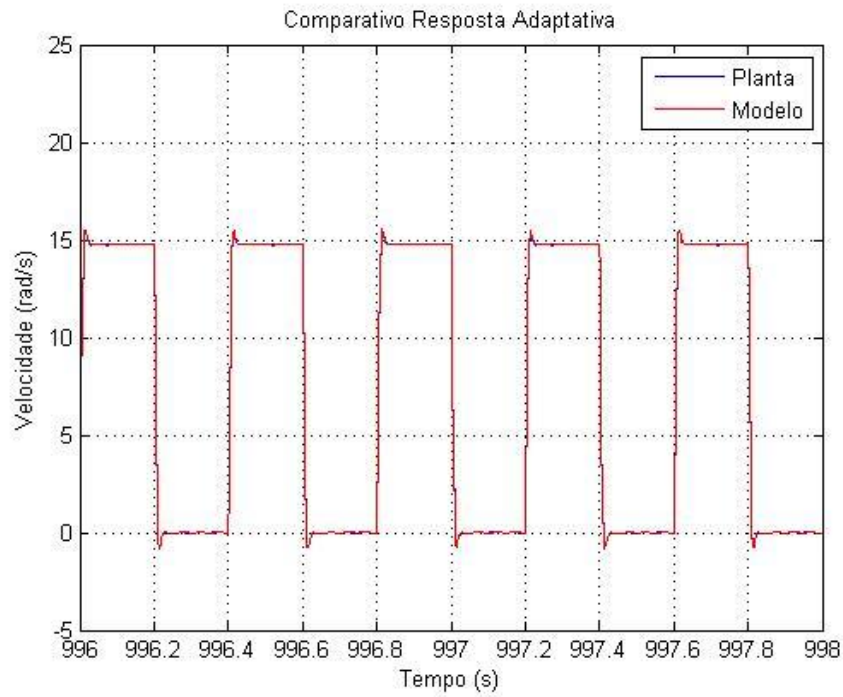


Figura 7.10 - Detalhe do comparativo ao longo da simulação

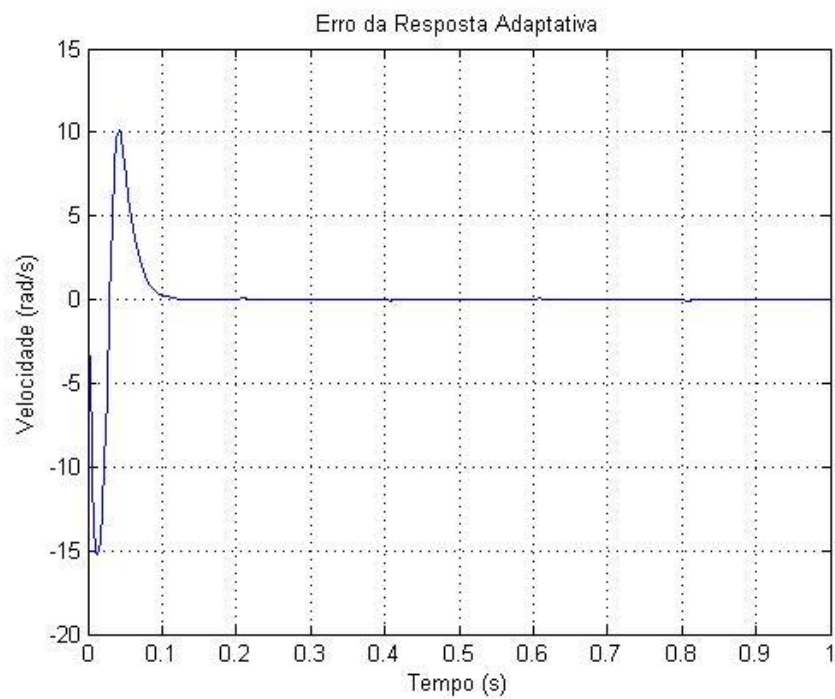


Figura 7.11 - Erro da resposta do sistema adaptativo em relação ao modelo de referência

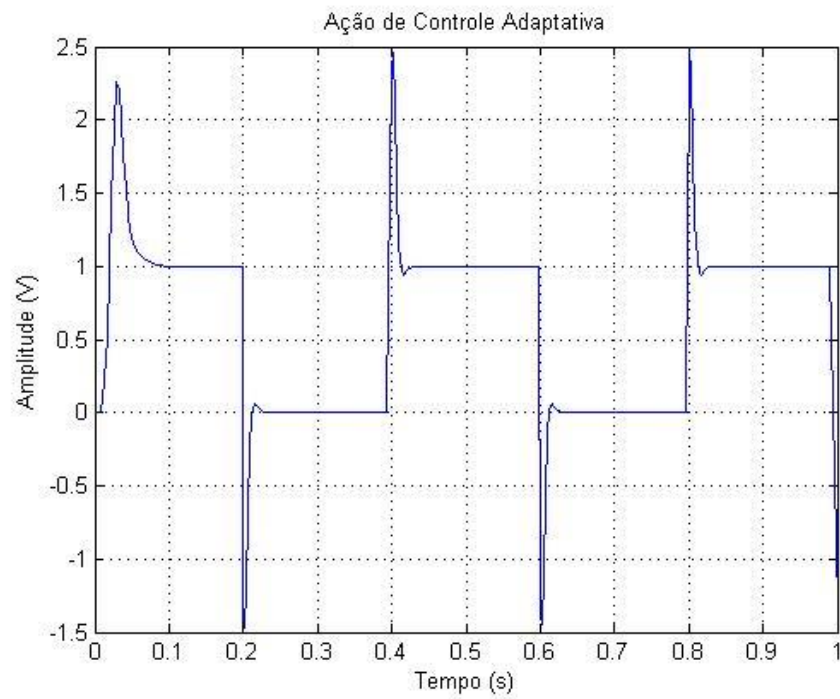


Figura 7.12 - Ação de controle do sistema adaptativo

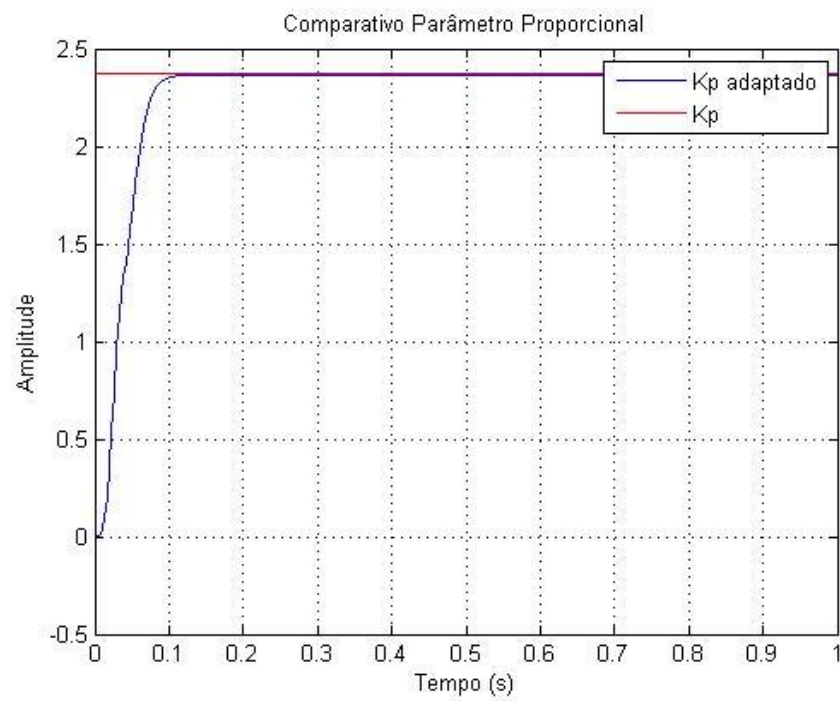


Figura 7.13 - Comparativo evidenciando a convergência do parâmetro proporcional

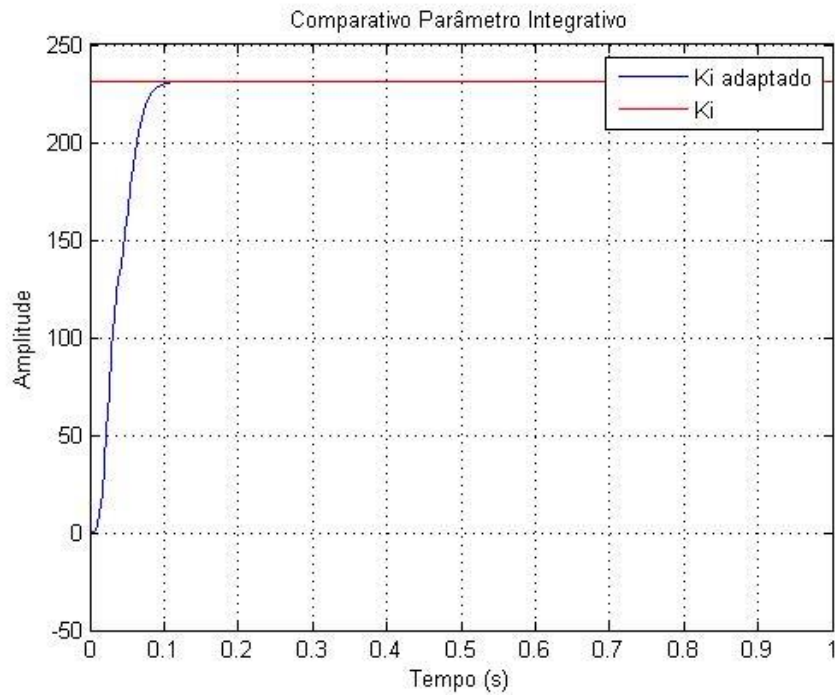


Figura 7.14 - Comparativo evidenciando a convergência do parâmetro integrativo

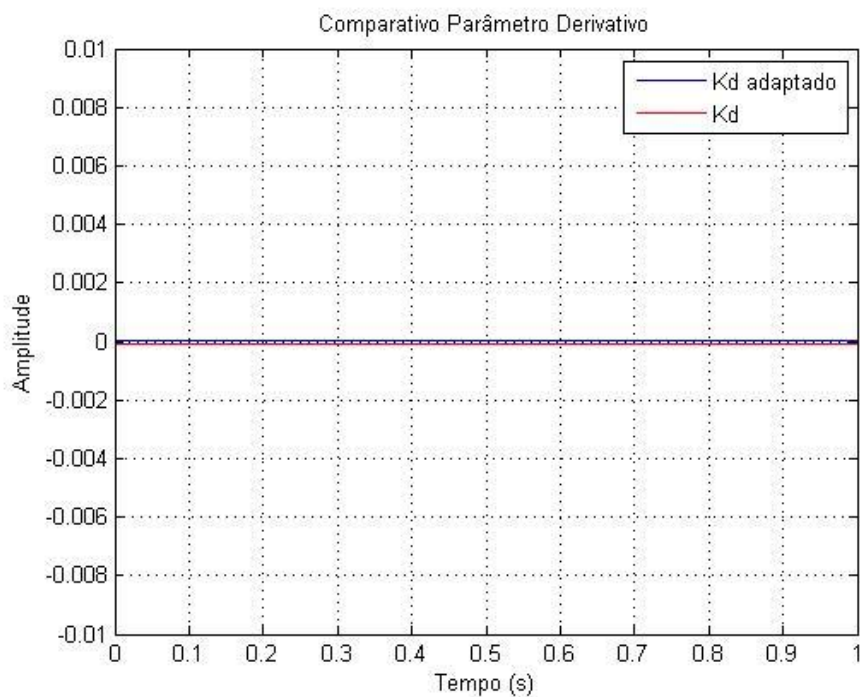


Figura 7.15 - Comparativo evidenciando a convergência do parâmetro derivativo

Observa-se que o mecanismo de ajuste está funcionando corretamente, uma vez que após passado um tempo, a resposta se mantém exatamente igual ao modelo de referência, ou seja, na resposta desejada, assim como os parâmetros K_p , K_i e K_d se igualam aos do modelo de referência, uma vez que ambos os processos são idênticos.

7.2.2 Fuzzy

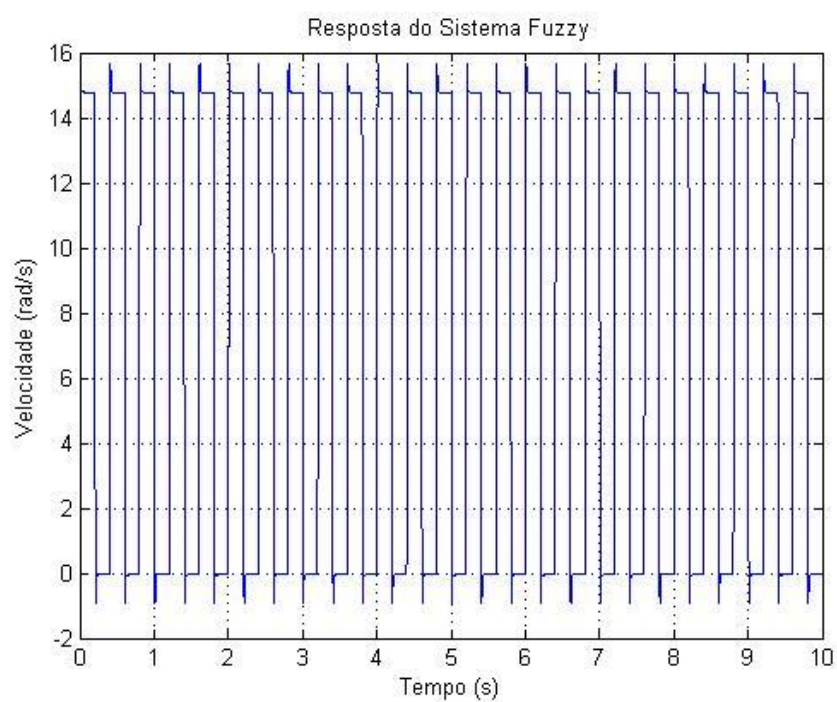


Figura 7.16 - Resposta do sistema Fuzzy

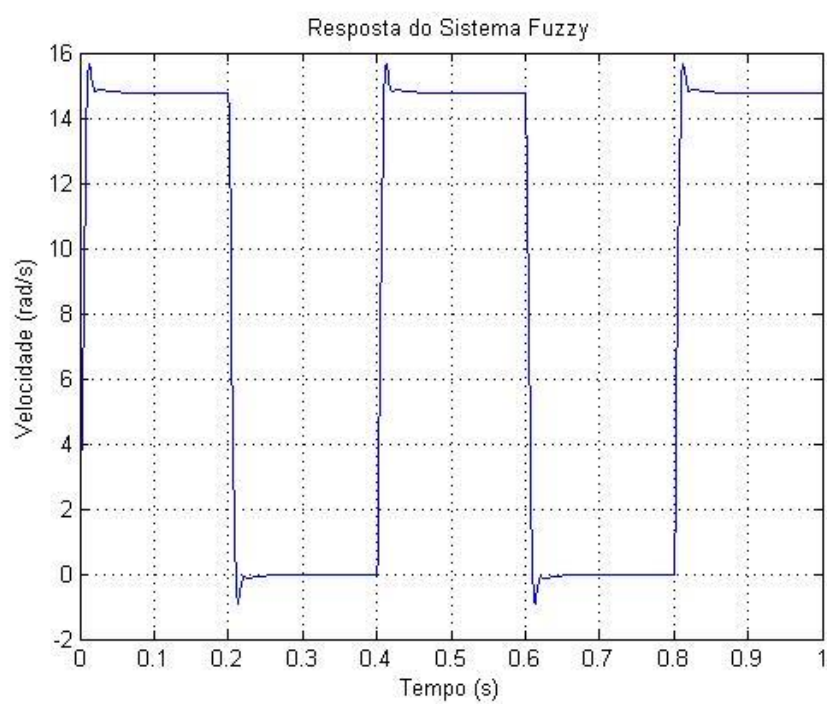


Figura 7.17 - Detalhe da resposta do sistema Fuzzy

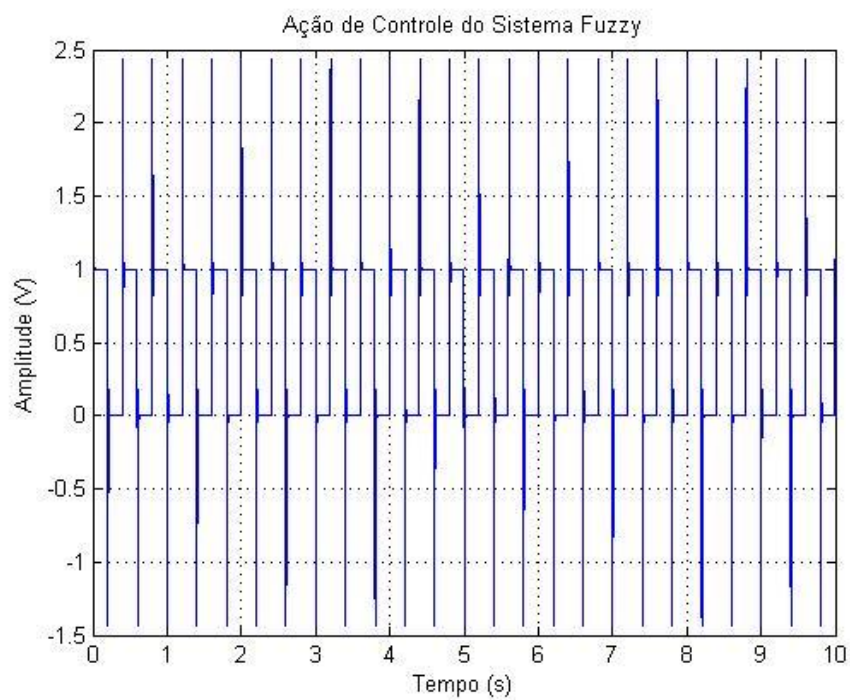


Figura 7.18 - Ação de controle do sistema Fuzzy

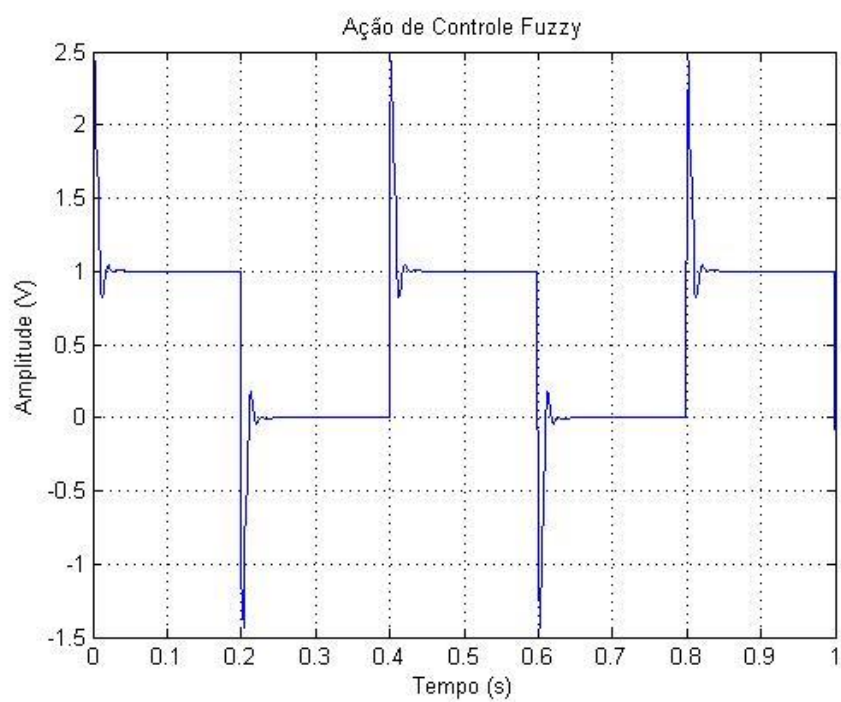


Figura 7.19 - Detalhe da ação de controle do sistema Fuzzy

7.2.3 Comparativo

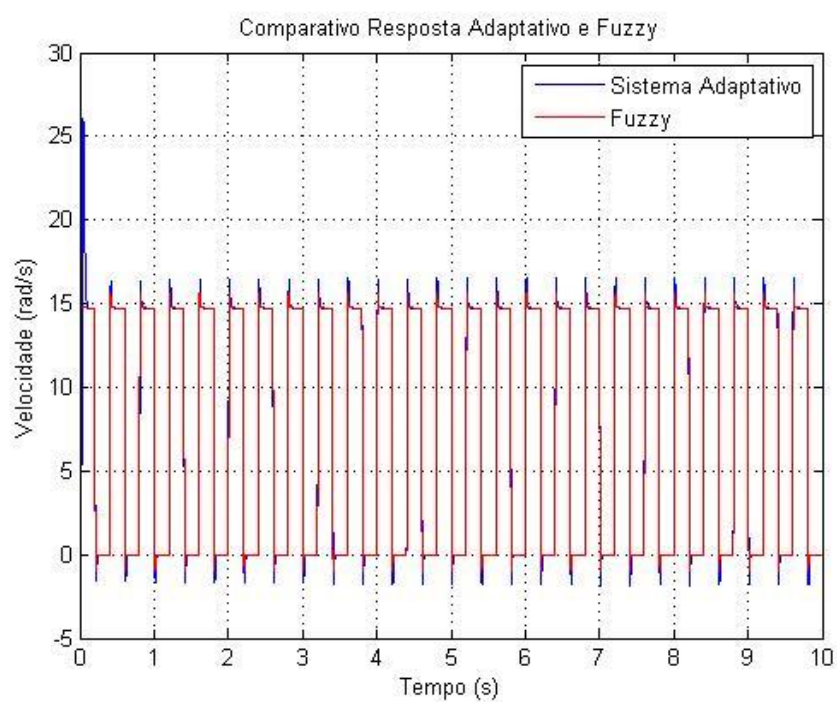


Figura 7.20 - Comparativo entre a resposta do sistema adaptativo e do sistema Fuzzy

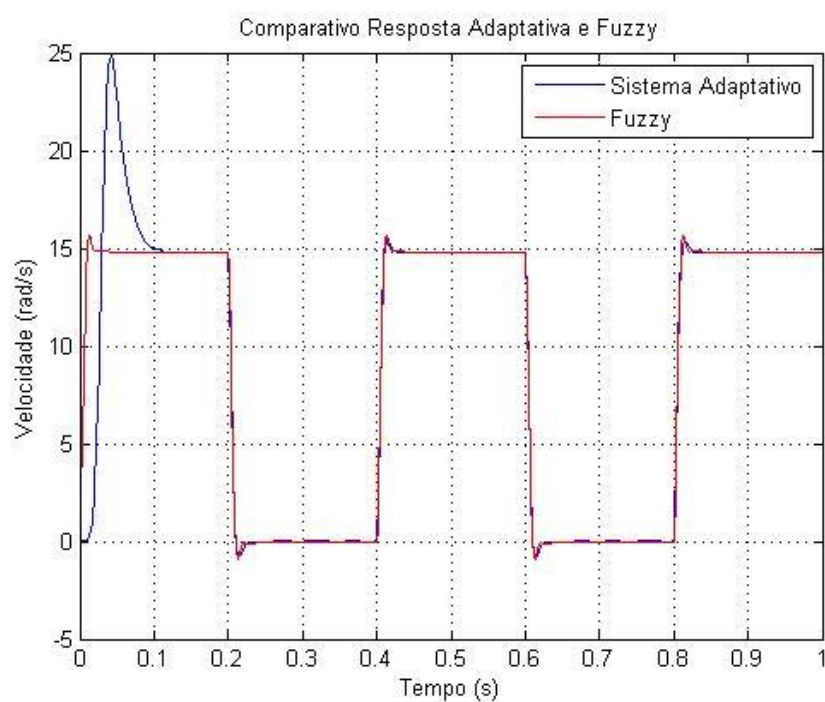


Figura 7.21 - Detalhe do comparativo da resposta

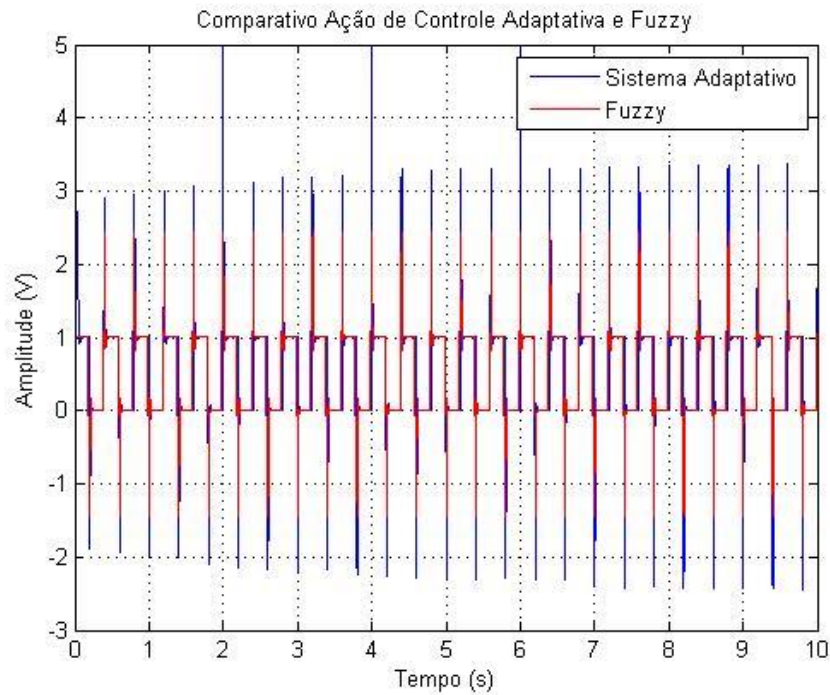


Figura 7.22 - Comparativo entre a ação de controle do sistema adaptativo e do sistema Fuzzy

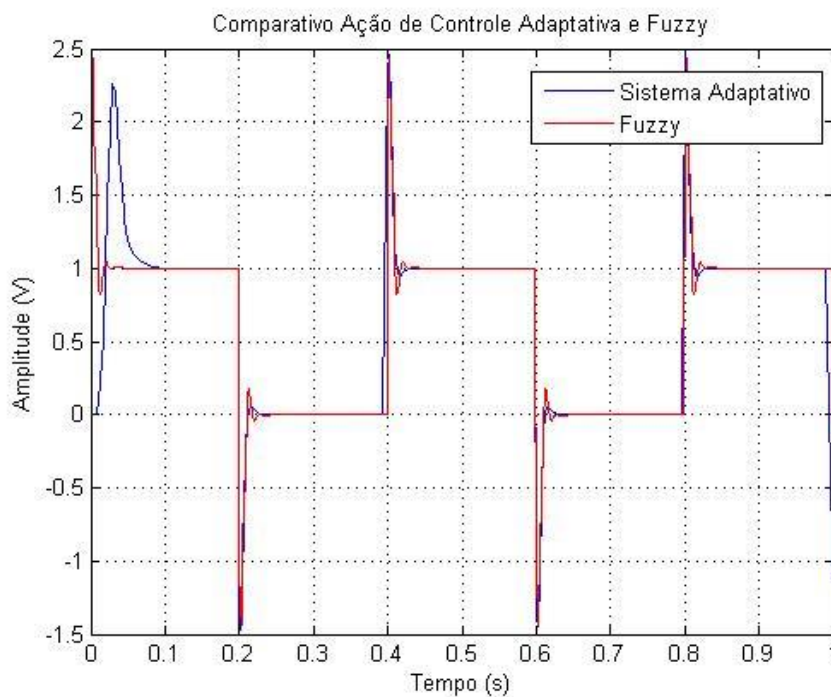


Figura 7.23 - Detalhe da ação de controle dos sistemas

Notamos novamente que a resposta do sistema Fuzzy, no primeiro instante é mais rápida, enquanto o sistema adaptativo ajusta seus parâmetros. Porém, ao longo do tempo, não temos praticamente diferença entre as respostas. Com relação a ação de controle, temos uma certa oscilação no sistema Fuzzy, porém, ainda assim, tanto a ação do sistema adaptativo, quanto a do sistema Fuzzy são relativamente parecidas após o início da simulação.

7.3 Aplicação de Carga

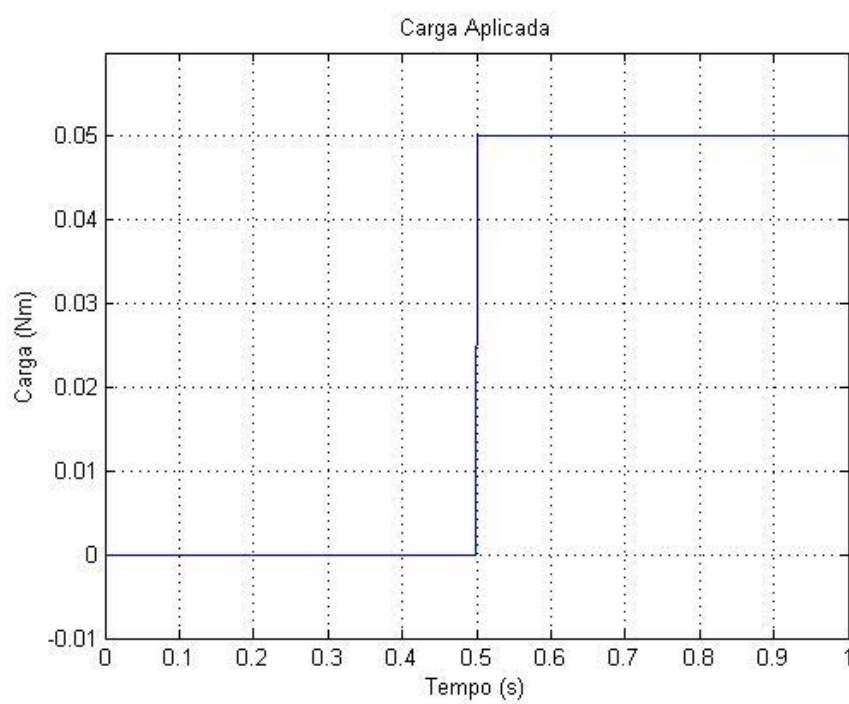


Figura 7.24 - Carga aplicada

7.3.1 MRAS

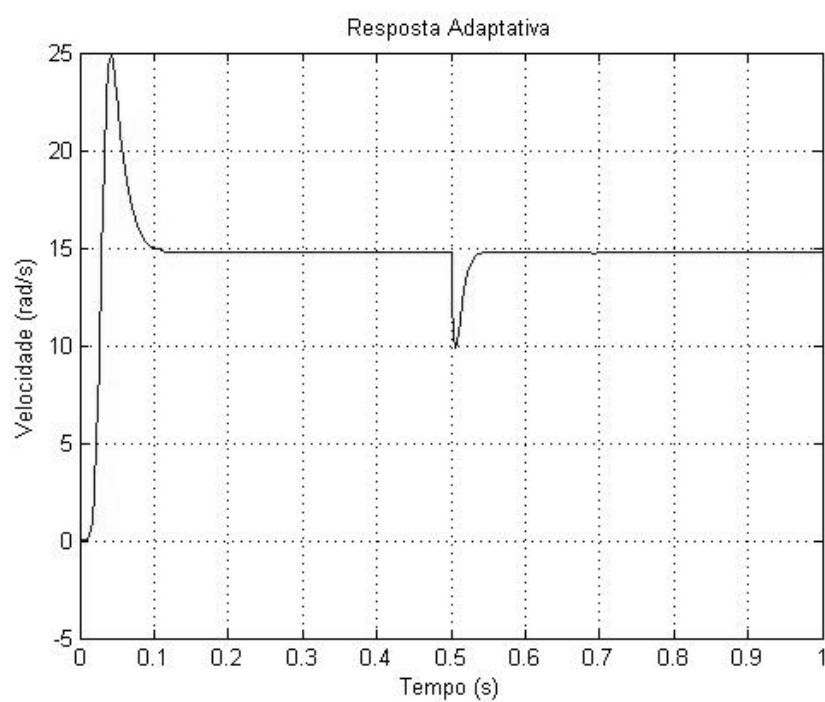


Figura 7.25 - Resposta do sistema adaptativo

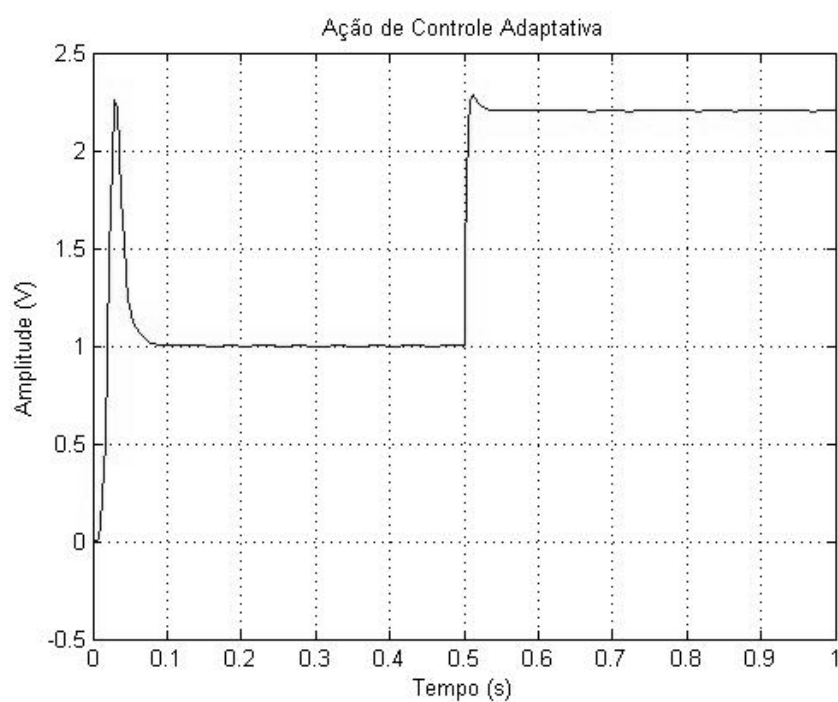


Figura 7.26 - Ação de controle do sistema adaptativo

7.3.2 Fuzzy

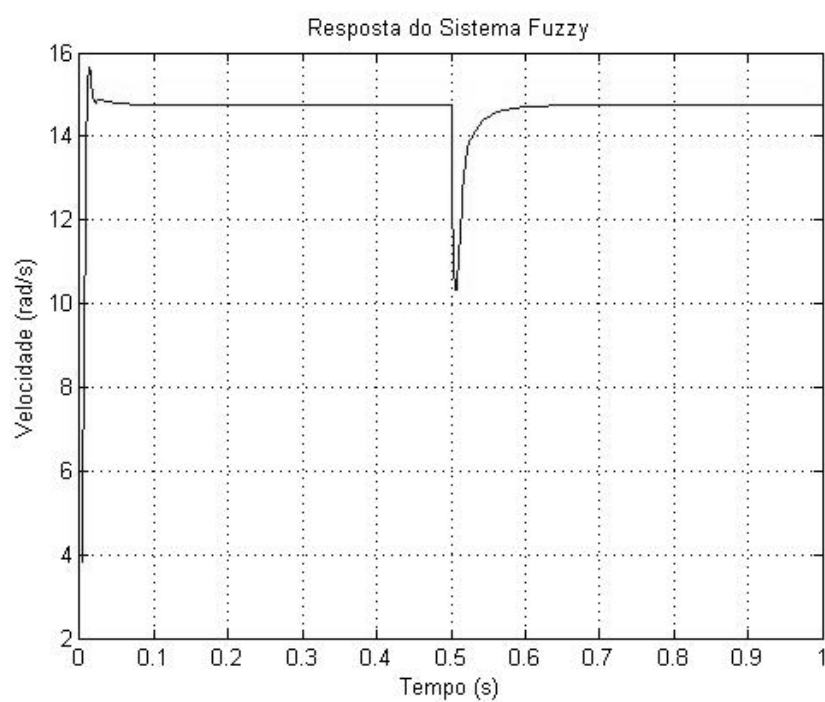


Figura 7.27 - Resposta do sistema Fuzzy

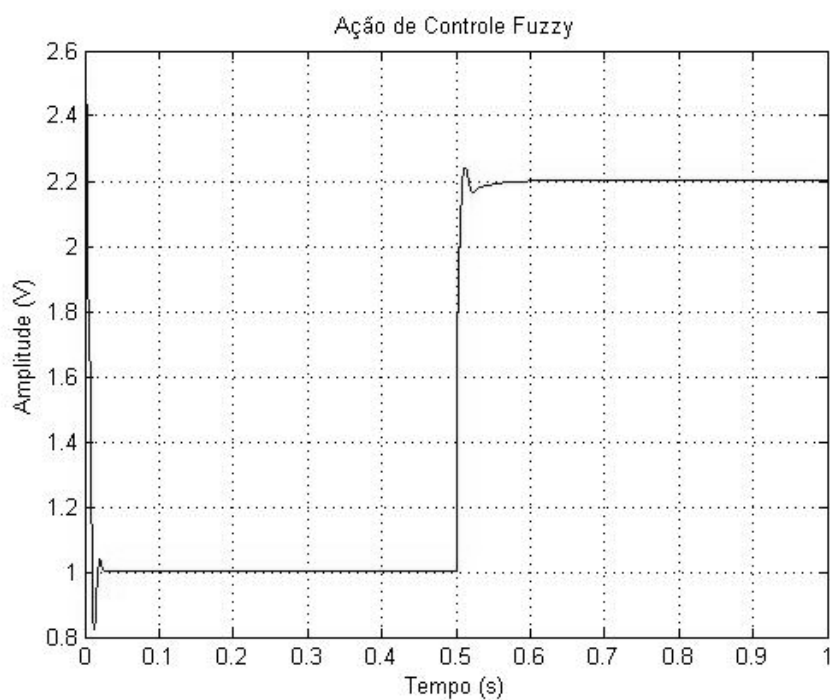


Figura 7.28 - Ação de controle do sistema Fuzzy

7.3.3 Comparativo

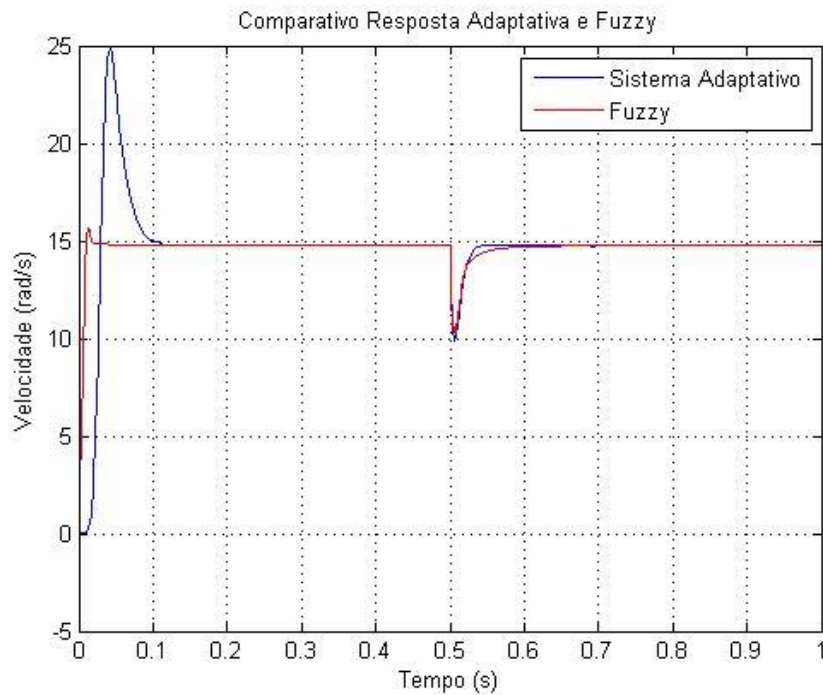


Figura 7.29 - Comparativo entre a resposta do sistema adaptativo e do sistema Fuzzy

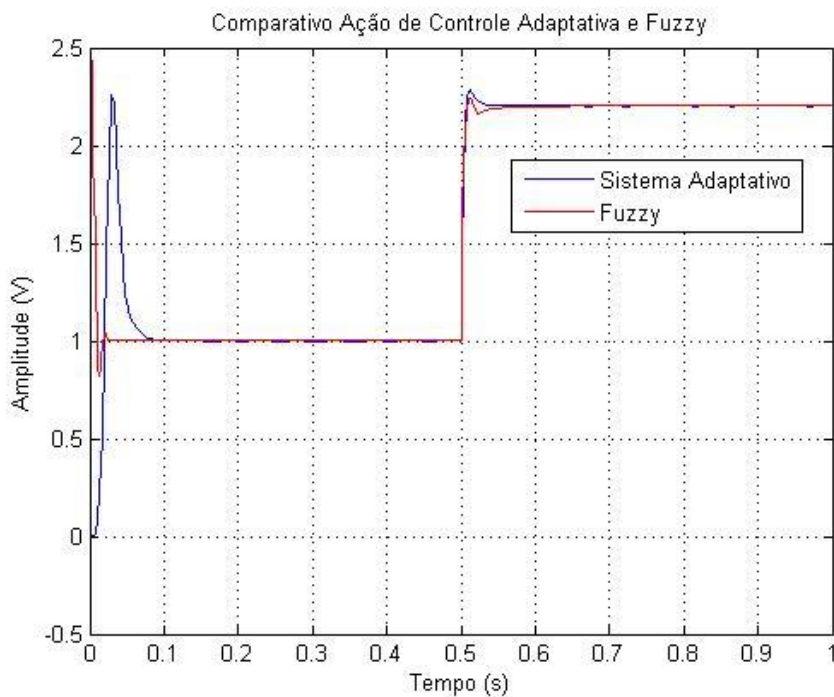


Figura 7.30 - Comparativo entre a ação de controle do sistema adaptativo e do sistema Fuzzy

A carga aplicada tende a diminuir a velocidade do motor, porém os sistemas de controle impedem isso. A resposta do sistema adaptativo à essa carga é um pouco mais rápida, o que significa que o mecanismo de ajuste após ajustado, responde melhor à variações. Em relação às ações de controle, ambas são muito parecidas.

7.4 Variações Paramétricas

7.4.1 Variação positiva de 10% na indutância e 10% na resistência

7.4.1.1 MRAS

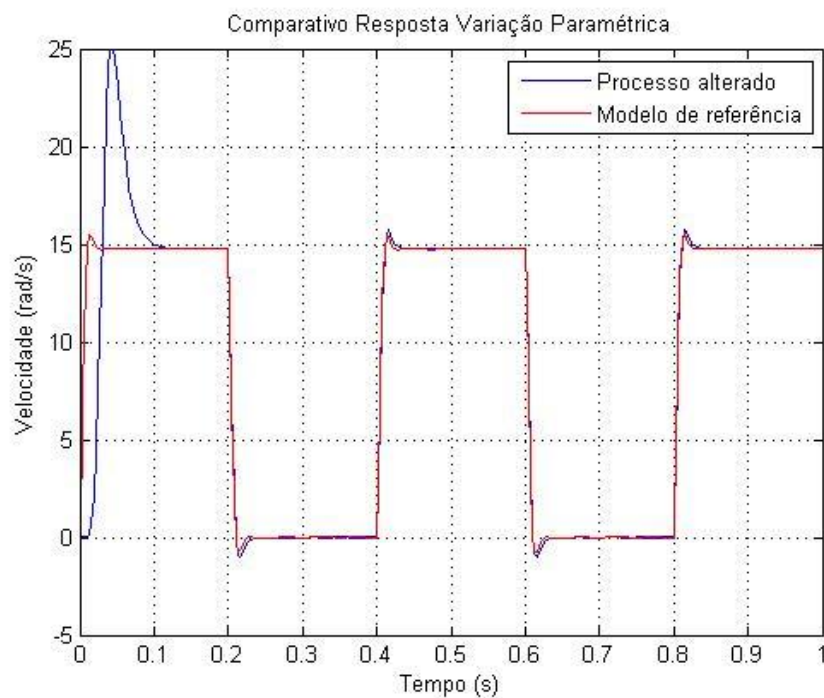


Figura 7.31 - Comparativo do início entre as resposta do modelo de referência e do processo alterado

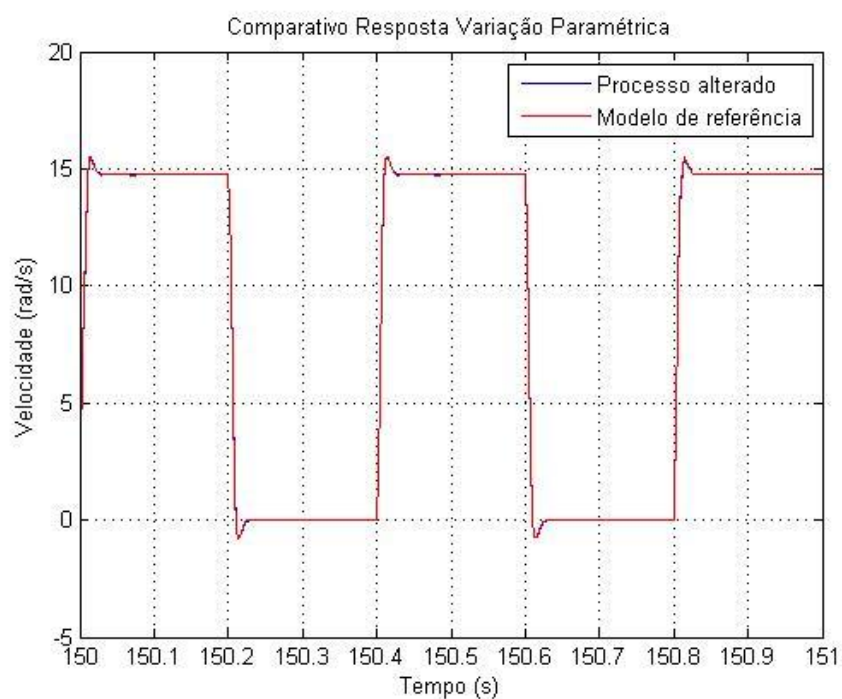


Figura 7.32 - Comparativo ao longo do tempo entre as respostas do modelo de referência e do processo alterado

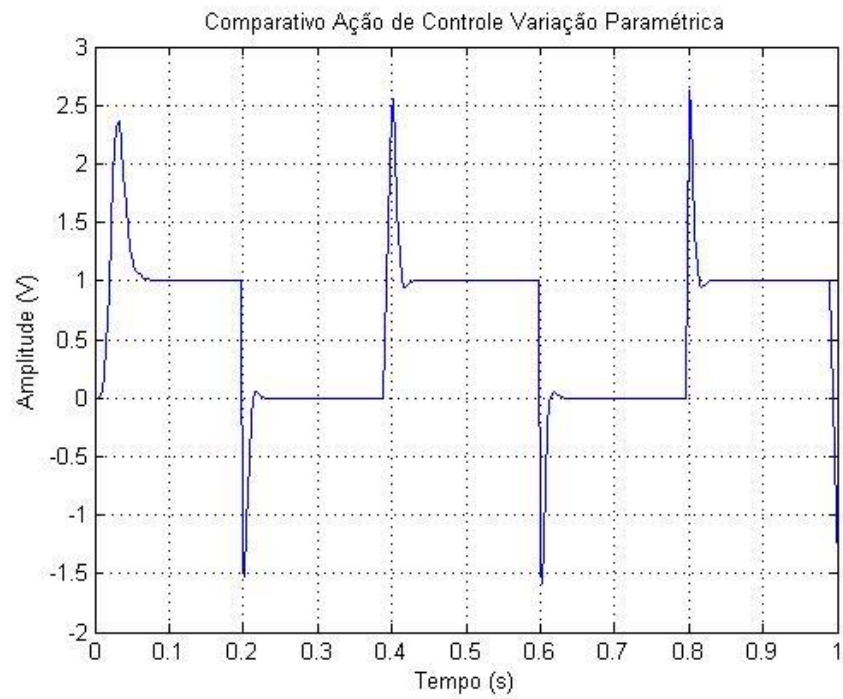


Figura 7.33 - Início da ação de controle do processo alterado

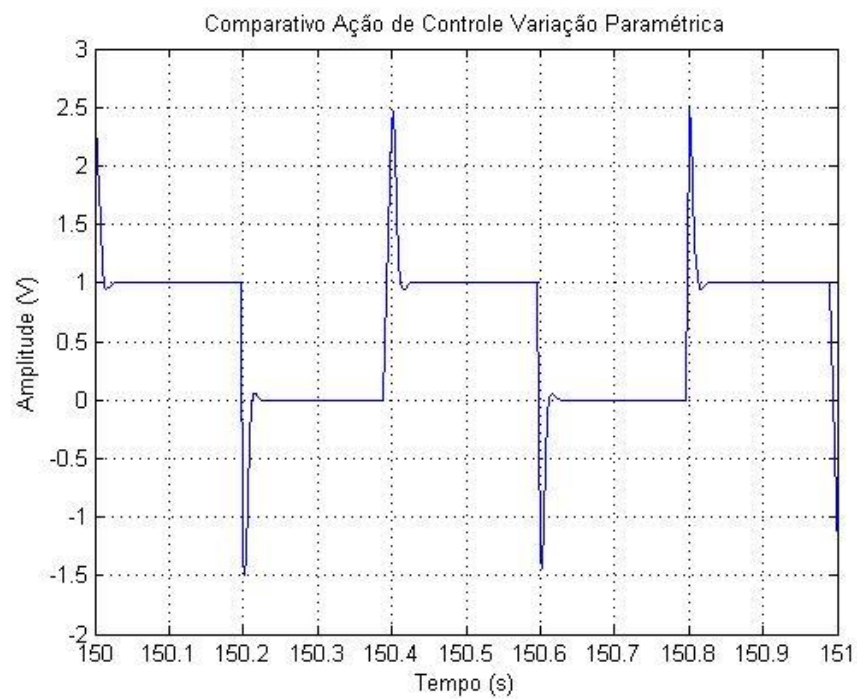


Figura 7.34 - Ação de controle do processo alterado

7.4.1.2 Fuzzy

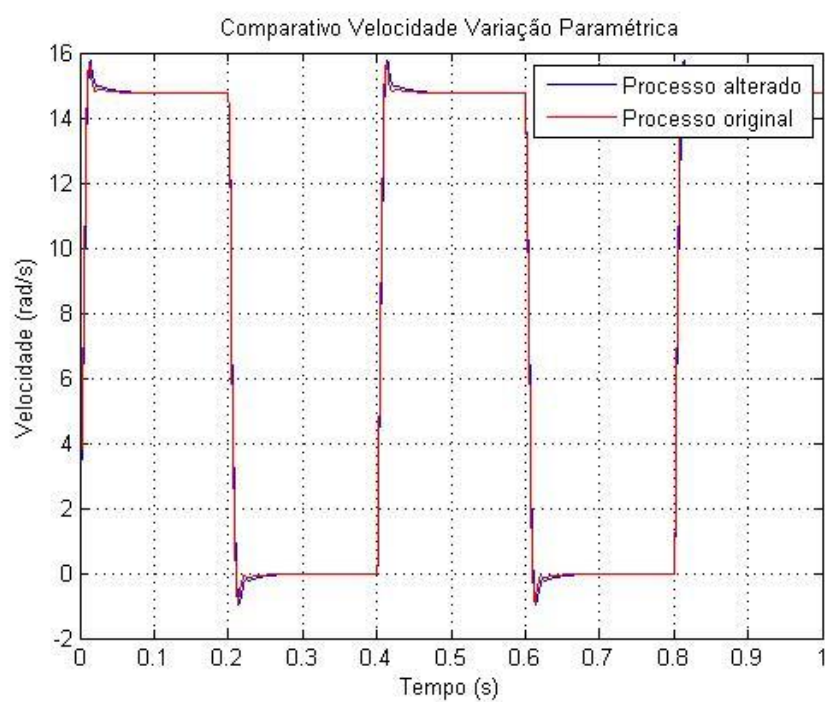


Figura 7.35 - Comparativo entre as respostas do processo sem alteração e do processo alterado

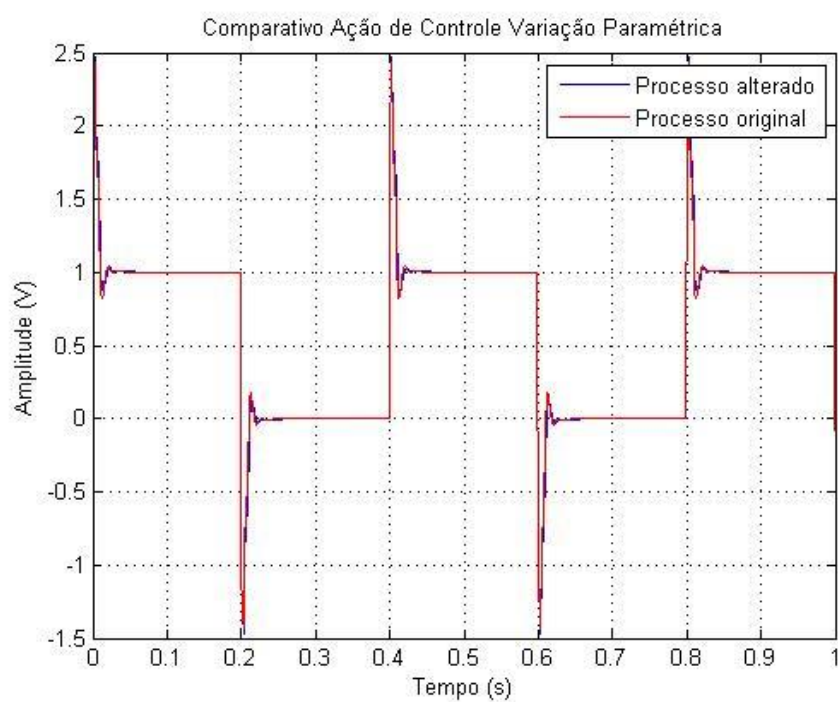


Figura 7.36 - Comparativo entre as ações de controle do processo sem alteração e do processo alterado

7.4.2 Variação positiva de 50% na inércia do motor (J) e 50% no coeficiente de atrito estático (B)

7.4.2.1 MRAS

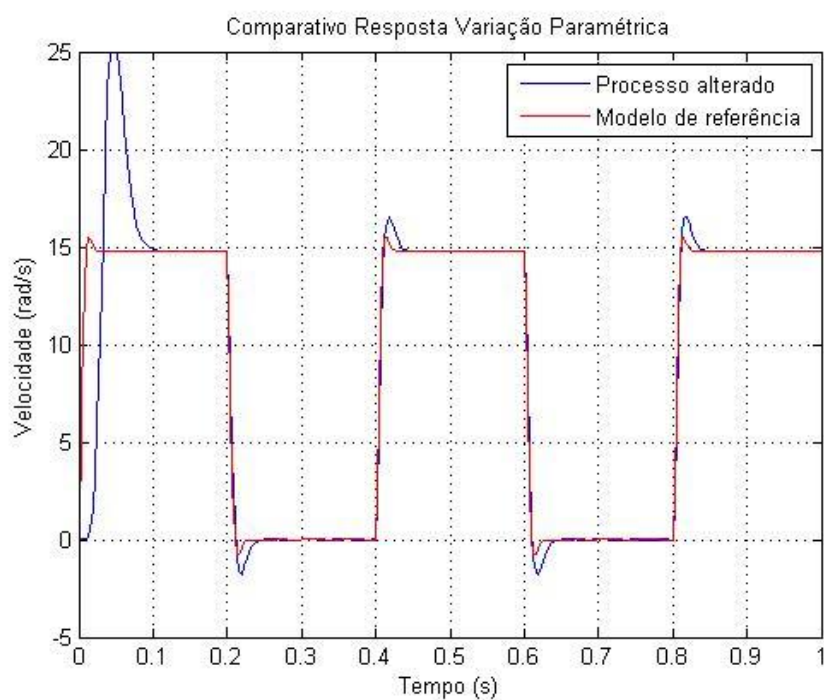


Figura 7.37 - Comparativo do início entre as resposta do modelo de referência e do processo alterado

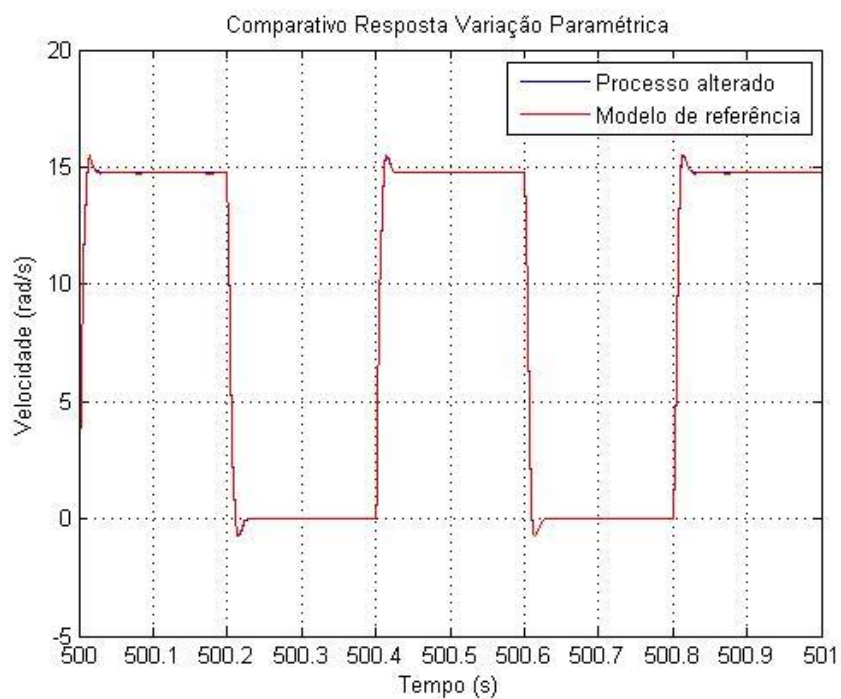


Figura 7.38 - Comparativo ao longo do tempo entre as resposta do modelo de referência e do processo alterado

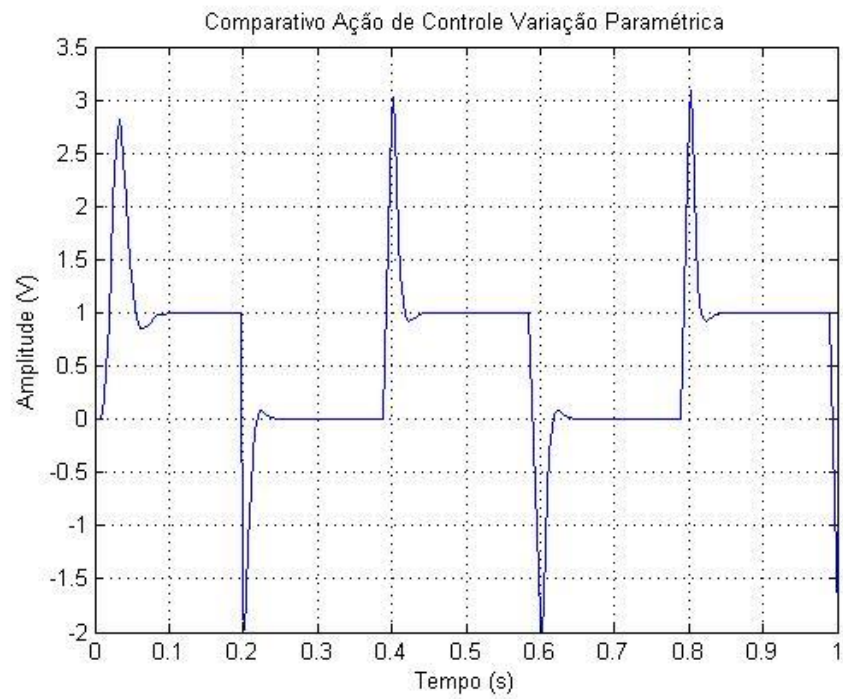


Figura 7.39 - Início da ação de controle do processo alterado

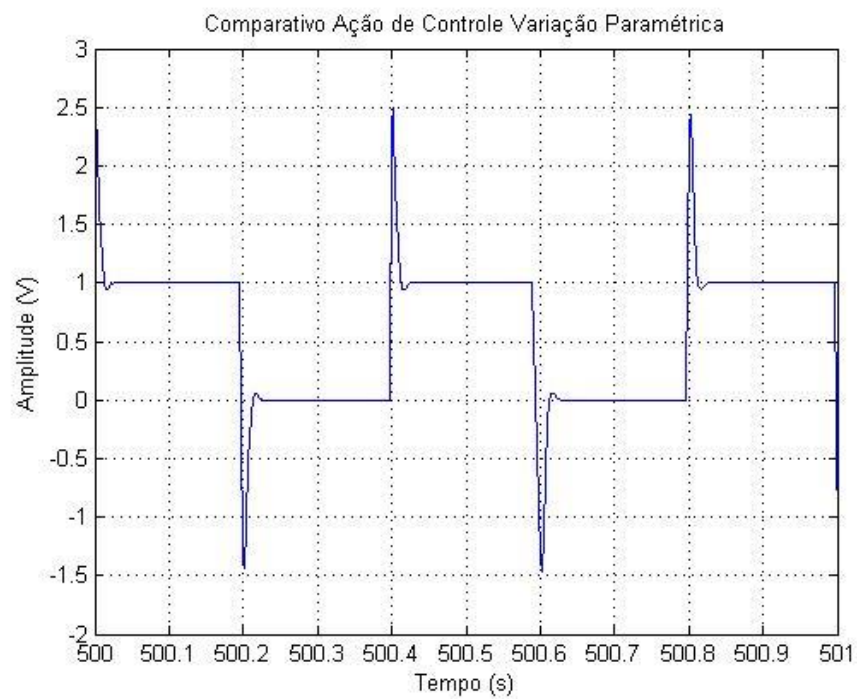


Figura 7.40 - Ação de controle do processo alterado

7.4.2.2 Fuzzy

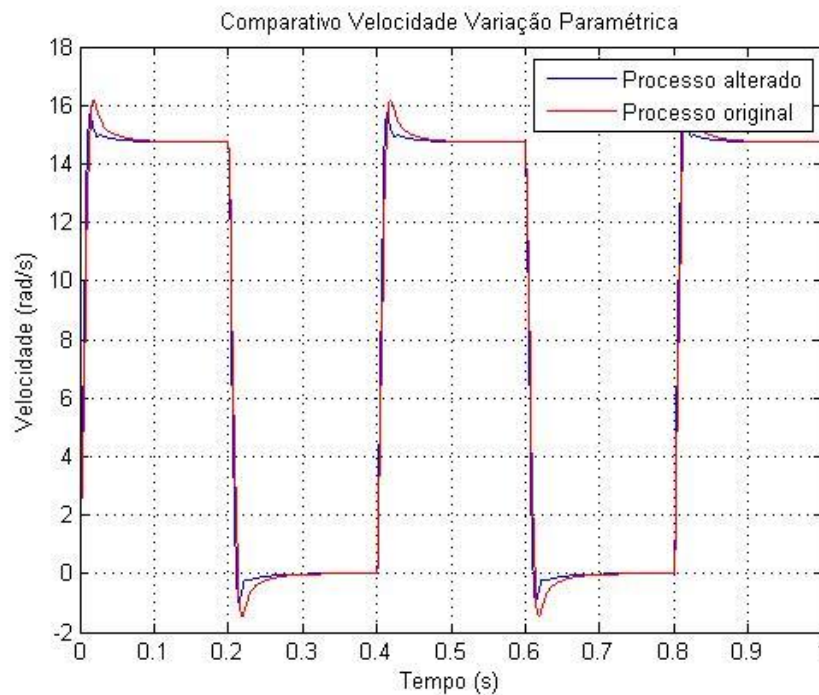


Figura 7.41 - Comparativo entre as respostas do processo sem alteração e do processo alterado

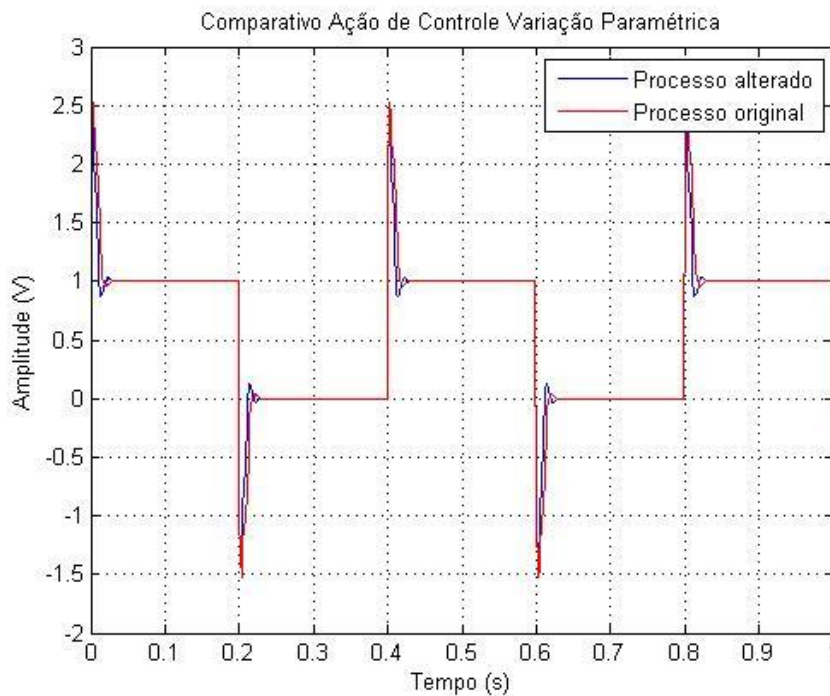


Figura 7.42 - Comparativo entre as ações de controle do processo sem alteração e do processo alterado

Como podemos observar, ambos os modelos conseguiram responder à essas variações mantendo a resposta no “set-point”. Porém, observamos que no caso do sistema adaptativo, quanto maior a variação dos parâmetros, maior o tempo que o mecanismo de ajuste atua, demorando mais para igualar a resposta com a do modelo de referência. No caso do sistema

Fuzzy, notamos que conforme temos uma variação maior, seu tempo para atingir o regime é mais longo, o que nos leva a concluir que ele apresenta uma certa limitação à variação dos parâmetros. No caso de uma variação muito grande, o sistema não Fuzzy não conseguirá acompanhar, resultando uma resposta errada.

8 Conclusão

Este Trabalho de Conclusão de Curso propôs o estudo comparativo de técnicas de controle inteligente da velocidade de um motor de corrente contínua. Utilizando como princípio o fato de controladores clássicos serem limitados, não respondendo bem à não linearidades, foram utilizadas duas técnicas relativamente novas para tal controle, o sistema adaptativo, com base no método do sistema adaptativo por modelo de referência, e o sistema Fuzzy.

Ficou claro que ambos controles são complexos, muito precisos e robustos. Eles respondem mais rapidamente que os controles clássicos, são menos sensíveis à variações e atingem sempre a resposta desejada, salvo exceções pontuais.

O controle adaptativo se mostrou matematicamente complexo, onde a regra MIT e o cálculo dos ganhos de adaptação exigem uma matemática extensa e não trivial.

Já o controle Fuzzy se mostrou muito amigável e intuitivo, porém, necessita de métodos computacionais para funcionamento.

Como diferenças principais entre as técnicas, ficou claro que o sistema Fuzzy é bem mais simples de ser implementado, seja pela não necessidade de definição do modelo exato, seja pela utilização de características mais próximas do pensamento humano e não de máquina, ou seja pela não exigência matemática do método. Porém o sistema adaptativo se mostrou menos limitado, uma vez que consegue contornar qualquer variação imposta, coisa que o Fuzzy não consegue.

Várias simulações foram realizadas a fim de comparar o comportamento dos dois controles, como degraus consecutivos, trem de pulsos, aplicação de carga e variações paramétricas. Com estas simulações, observou-se que os sistemas ao longo do tempo se comportam de forma parecida, exceto no início onde o mecanismo de ajuste do controle adaptativo ainda tenta buscar a convergência do método. Observou-se também que após o ajuste dos parâmetros do sistema adaptativo, este responde de forma um pouco mais rápida que o sistema Fuzzy.

Com as simulações, comprovou-se de fato que ambos controles são adaptativos, uma vez que sempre retornavam à resposta especificada, o que não ocorre com os controles clássicos.

Da inserção de variações paramétricas, notou-se que quanto maior a variação, ou seja, quão mais distante do processo original está o processo a ser controlado, mais tempo os sistemas levam para alcançarem o regime. No caso do sistema adaptativo, mais tempo leva para

o mecanismo de ajuste se adaptar. No caso do sistema Fuzzy, mais tempo leva para obter-se o regime.

Um fato a ser destacado é que caso as variações sejam elevadas demais, o sistema Fuzzy pode não conseguir contornar este problema, levando a uma resposta inadequada. Porém nas situações reais, essas variações dificilmente atingem tal patamar a ponto de o controle Fuzzy não responder adequadamente.

Como conclusão final, ambos os sistemas de controle utilizados apresentaram-se suficientemente eficientes e robustos, podendo ser aplicados em qualquer situação prática, devendo o projetista analisar qual dos dois se desenvolve melhor com a situação utilizada.

Como trabalhos futuros, podemos sugerir a implementação prática de tais controladores, a fim de verificar o real comportamento do sistema controlado. Sugere-se também simulações mais “agressivas”, onde o desempenho do controlador seja realmente exigido, como no caso do trem de pulsos, diminuir o período de cada pulso.

Referências

- [1] OLIVEIRA, V. A., AGUIAR, M. L., VARGAS, J. B.; Sistemas de controle: aulas de laboratório – 2005.
- [2] DORF, R. C., BISHOP, R. H.; Sistemas de controle modernos – 2009.
- [3] OGATA, K.; Engenharia de controle moderno - 2006.
- [4] ASTROM, K. J., WITTENMARK, B.; Adaptive control - 1995.
- [5] GOMES, Yuri C., ARAÚJO, Tomás V. G. P. e LIMA, Antônio M. N.; Abordagem didática com controle adaptativo - Universidade Federal de Campina Grande – 2006.
- [6] STEFANELLO, M.; Controle adaptativo robusto de estrutura variável de modelo de referência aplicado a filtros ativos de potência – Tese de doutorado – Universidade Federal de Santa Maria – 2010.
- [7] RURUA, A.; A method for finding good values of adaptation gains – University of Limerick – Ireland - 2004.
- [8] CONTI, C.; Projeto de controladores e processos de implementação – Programa Ensinar com Pesquisa – Universidade de São Paulo – 2008.
- [9] AZEVEDO, F. M., BRASIL, L. M., OLIVEIRA, R. C. L.; Redes neurais com aplicações em controle e sistemas especialistas – 2000.
- [10] YEN, J., LANGARI, R.; Fuzzy logic: Intelligence, control, and information – 1998.
- [11] ZADEH, L. A.; Fuzzy sets. Information and control – 1965.
- [12] RASHID, N. K. A., HEGER, A. S.; Fuzzy logic and control: software and hardware applications – 1993.
- [13] LAYNE, J. R., PASSINO, K. M.; Fuzzy model reference learning control – 1996.
- [14] MRAD, F., DEEB, G.; Experimental comparative analysis of adaptive fuzzy logic controllers – 2002.

Apêndice

Desenvolvimento via Matlab do Motor, PID e Modelo de Referência

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%      Universidade de São Paulo - USP      %%
%% Escola de Engenharia de São Carlos - EESC %%
%% Departamento de Engenharia Elétrica - SEL %%
%%      Trabalho de conclusão de curso      %%
%% Desenvolvido por Cyro de Araujo Gianelli %%
%%      Orientador Jerson Barbosa Vargas    %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;
close all;
clc;

%%
%Parametros do motor Eletrocraft
R = 1.63;
L = 3e-3;
J = 3.67e-5;
Ke = 0.0678;
B = 1.1224e-6;
Kt = Ke;
Kconf = 0.445;
Ktg = 0.15242;

%%
%Função transferencia do motor
tf1 = tf(1, [L R]);
g1 = Kt;
tf2 = tf(1, [J B]);
g2 = Ke;

fprintf('função transferencia da planta: \n')
Gmotor = feedback(series(g1*tf1,tf2),g2)

%%
%Aplicando degrau e impulso na entrada em malha aberta
step(Gmotor)
figure;
impulse(Gmotor)

%%
%Utilizando método lugar de raizes para construir o PID
rltool(Gmotor);

%%
%Função do PID encontrada
Gpid = tf([-0.000102557097099405 2.39288588703584 231.098297124666],[1
0])

%%
%Sistema em malha fechada - modelo de referência
fprintf('Função transferencia modelo de referencia: \n')
Gtotal = feedback(series(Gmotor,Gpid),Ktg*Kconf)

figure;
step(Gtotal)
```

Desenvolvimento via Matlab do método para encontrar os ganhos de adaptação, diagrama de simulink para o mesmo

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%      Universidade de São Paulo - USP      %%
%% Escola de Engenharia de São Carlos - EESC %%
%% Departamento de Engenharia Elétrica - SEL %%
%%      Trabalho de conclusão de curso      %%
%% Desenvolvido por Cyro de Araujo Gianelli %%
%%      Orientador Jerson Barbosa Vargas    %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;
close all;
clc;
warning off;

%%
%Parâmetros do motor Eletrocraft
R = 1.63;
L = 3e-3;
J = 3.67e-5;
Ke = 0.0678;
B = 1.1224e-6;
Kt = Ke;
Kconf = 0.445;
Ktg = 0.15242;

%%
%Configuração para a utilização da função fminsearch
options = optimset('display','iter');

global gamap gamai gamad kplinha kilinha kdlinha errototal tempo
gamap= -100;
gamai= -10000;
gamad= -0.0005;

%%
%Chamada da função fminsearch
pmin = fminsearch('desempenhonovo', [gamap gamai gamad], options)

%%
%Equação a ser minimizada por fminsearch
function [erro] = desempenhonovo(x)
global gamap gamai gamad kplinha kilinha kdlinha errototal tempo
gamap = x(1);
gamai = x(2);
gamad = x(3);
[T]=sim('adaptново',[0 10]);
erro = sum(((kplinha-2.39288588703584).^2).*tempo+((kilinea-
231.098297124666).^2).*tempo+((kdlinha+0.000102557097099405)
.^2).*tempo);
```

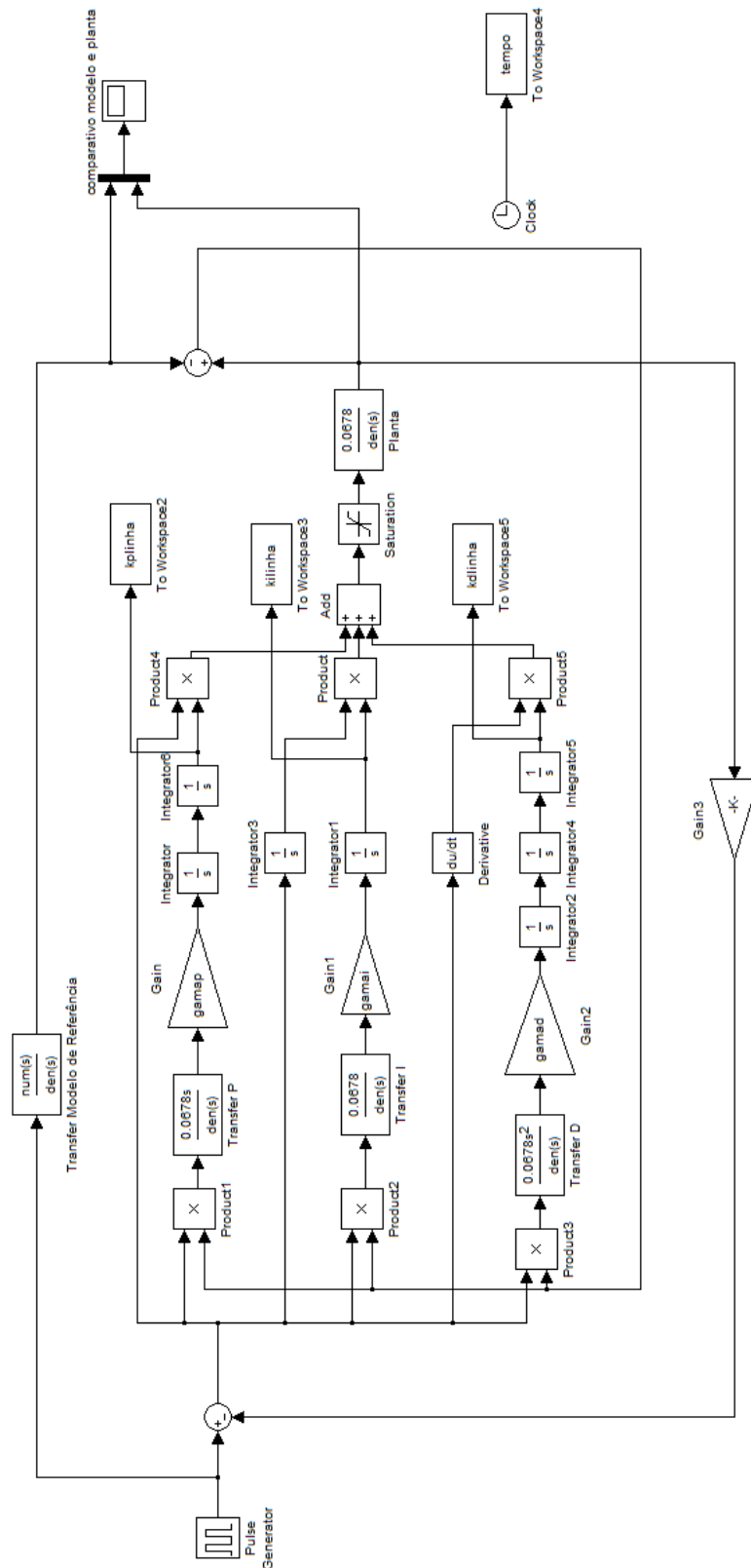


Figura I.0.1 - Diagrama utilizado para cálculo dos ganhos de adaptação

Desenvolvimento via Matlab do método para encontrar Kp, Ki e Kd do sistema

Fuzzy, diagrama de simulink para o mesmo

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%      Universidade de São Paulo - USP      %%
%% Escola de Engenharia de São Carlos - EESC %%
%% Departamento de Engenharia Elétrica - SEL %%
%%      Trabalho de conclusão de curso      %%
%% Desenvolvido por Cyro de Araujo Gianelli %%
%%      Orientador Jerson Barbosa Vargas    %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;
close all;
clc;
warning off;

%%
%Parâmetros do motor Eletrocraft
R = 1.63;
L = 3e-3;
J = 3.67e-5;
Ke = 0.0678;
B = 1.1224e-6;
Kt = Ke;
Kconf = 0.445;
Ktg = 0.15242;

%%
%Configuração para a utilização da função fminsearch
options = optimset('display','iter');

global Kp Ki Kd errototal tempo
gamap= -2.3;
gamai= -200;
gamad= -1e-4;

%%
%Chamada da função fminsearch
pmin = fminsearch('desempenho', [Kp Ki Kd], options)

%%
%Equação a ser minimizada por fminsearch
function [erro] = desempenho(x)
global Kp Ki Kd errototal tempo
Kp = x(1);
Ki = x(2);
Kd = x(3);
[T]=sim('fuzzyteste',[0 1]);
erro = sum(((errototal).^2).*tempo);
```

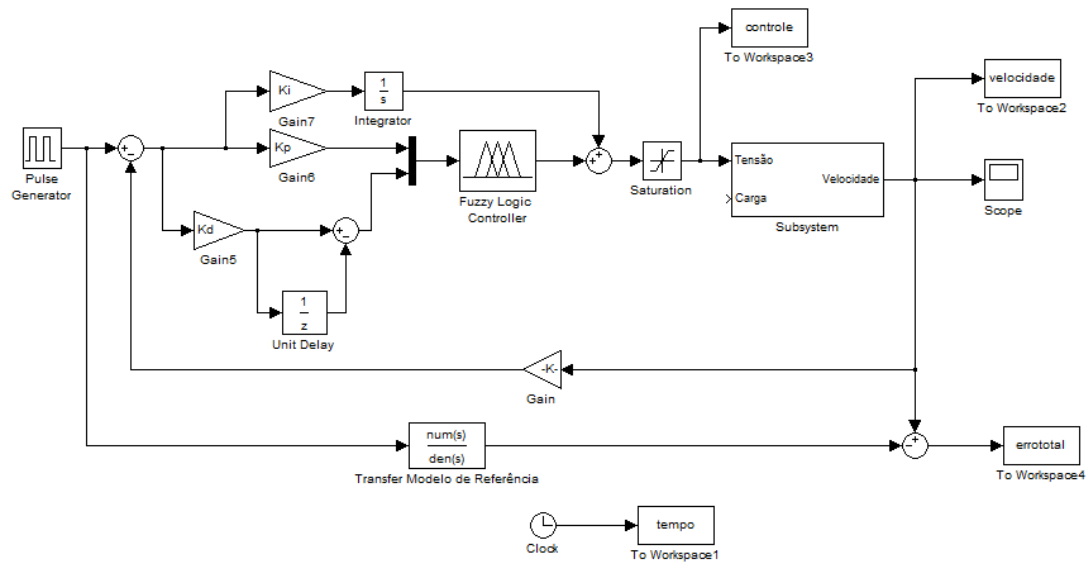


Figura I.0.2 - Diagrama utilizado para cálculo de K_p , K_i e K_d