

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



**DESENVOLVIMENTO DE UM REGULADOR AUTOMÁTICO  
DE TENSÃO E DE UM ESTABILIZADOR DE SISTEMA DE POTÊN-  
CIA UTILIZANDO LÓGICA DIFUSA**

**Carlos Eduardo dos Santos**

**Dissertação de Mestrado**

**Itajubá – MG.  
2007**

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

# **DESENVOLVIMENTO DE UM REGULADOR AUTOMÁTICO DE TENSÃO E DE UM ESTABILIZADOR DE SISTEMA DE POTÊN- CIA UTILIZANDO LÓGICA DIFUSA**

Dissertação apresentada ao  
Programa de Pós-Graduação em  
Engenharia Elétrica da UNIFEI  
como requisito parcial para obtenção  
do título de Mestre em Engenharia Elétrica

Área de Concentração  
Automação e Sistemas Elétricos Industriais

Orientador: Prof. Dr. Carlos Alberto Murari Pinheiro  
Co-Orientador: Prof. Dr. Ângelo José Junqueira Rezek

Itajubá – MG.  
Dezembro 2007

---

## DEDICATÓRIA

Esse estudo é dedicado especialmente a um pai herói que apesar de todos os infortúnios da vida sempre teve garra para conduzir a sua família pelos caminhos da honestidade e perseverança.

A uma mãe dedicada que abdicou de parte de uma carreira brilhante para educar e cuidar, sempre zelosa, de seus filhos.

E com carinho a Lidiane, uma amiga muito especial, que foi a melhor pessoa que Deus poderia ter posto em minha vida, sempre serena, me ajudou e incentivou a continuar perseguindo meus objetivos.

Dedico também a todas as pessoas que direta ou indiretamente me ajudaram e apoiaram nesse desafio.

---

## AGRADECIMENTOS

Agradeço aos professores Carlos Alberto Murari Pinheiro e Ângelo José Junqueira Rezek que foram mais que orientadores, foram amigos dedicados, por toda a ajuda que ambos prestaram no decorrer desse trabalho.

Aos colegas de mestrado pelo apóio técnico e humano durante esse tempo de estudo.

A minha família toda que sempre me incentivou, mas em especial a minha mãe Marlene, meu pai Donizette.

A **Deus** acima de tudo, por sempre estar ao meu lado ajudando na tarefa nem sempre fácil de alcançar os meus sonhos.

---

## SUMÁRIO

SUMÁRIO .....	v
LISTA DE FIGURAS .....	vii
LISTA DE TABELAS .....	x
LISTA DE SÍMBOLOS E NOMENCLATURAS .....	xi
RESUMO .....	xv
ABSTRACT .....	xvi
CAPÍTULO 1 - INTRODUÇÃO .....	1
CAPÍTULO 2 - REVISÃO BIBLIOGRÁFICA .....	3
CAPÍTULO 3 - SISTEMA DE CONTROLE DE EXCITAÇÃO .....	5
3.1. INTRODUÇÃO .....	5
3.2. TIPOS DE SISTEMA DE EXCITAÇÃO .....	6
3.3. MODELAGEM DE UM SISTEMA DE EXCITAÇÃO .....	8
CAPÍTULO 4 - CONCEITOS DE LÓGICA <i>FUZZY</i> .....	15
4.1. CONCEITUAÇÃO BÁSICA .....	15
4.2. PROPOSTA DE SISTEMA AVR – PSS UTILIZANDO LÓGICA <i>FUZZY</i> .....	17
CAPÍTULO 5 - SIMULAÇÃO .....	18
5.1. INTRODUÇÃO .....	18
5.2. MODELAGEM PSS <i>FUZZY</i> .....	18
5.3. MODELAGEM DO AVR CLÁSSICO COM PSS <i>FUZZY</i> .....	20
5.4. MODELAGEM DO AVR <i>FUZZY</i> COM PSS <i>FUZZY</i> .....	24
CAPÍTULO 6 - IMPLEMENTAÇÃO PRÁTICA DO AVR <i>FUZZY</i> .....	31
6.1. INTRODUÇÃO .....	31
6.2. BANCADAS DE ENSAIOS .....	31
6.3. MODELAGEM DA BANCADA DE ENSAIO .....	38
6.4. LINEARIZAÇÃO DO CONJUNTO CIRCUITO DE DISPARO E PONTE TIRISTORIZADA .....	41
6.5. IDENTIFICAÇÃO DO MODELO DO SISTEMA .....	43
6.6. PROJETO DE UM COMPENSADOR TIPO LEAD/LAG .....	46
6.7. PROJETO DE UM PSS CONVENCIONAL .....	49
6.8. PROJETO DE UM PSS <i>FUZZY</i> .....	51
6.9. ENSAIOS COM O COMPENSADOR AVR CLÁSSICO .....	53
6.10. PROJETO DE UM COMPENSADOR AVR <i>FUZZY</i> .....	61
6.11. ENSAIOS COM O COMPENSADOR AVR <i>FUZZY</i> .....	63
6.12. ENSAIOS DE PARALELISMO DE GERADORES COM AVR CONVENCIONAL .....	67
6.13. ENSAIOS DE PARALELISMO COM AVR <i>FUZZY</i> .....	69
CAPÍTULO 7 - CONCLUSÃO .....	71
7.1. CONCLUSÃO .....	71

---

7.2. TRABALHOS FUTUROS .....	72
BIBLIOGRAFIA.....	73
ANEXO 1: CONTROLE AUTOMÁTICO DE VELOCIDADE.....	76
ANEXO 2: TÉCNICA DO FOGO GIRANTE .....	83
ANEXO 3: INFORMAÇÕES BÁSICAS SOBRE A APLACA DE AQUISIÇÃO.....	84
ANEXO 4: PROGRAMA DE IMPLEMENTAÇÃO DO AVR TRADICIONAL EM C	87
ANEXO 5: FUNÇÃO “READER01” .....	97
ANEXO 6: FUNÇÃO “READER02” .....	100
ANEXO 7: PROGRAMA DE IMPLEMENTAÇÃO DO AVR E PSS FUZZY.....	103
ANEXO 8: MODELAGEM DOS FILTROS PASSA- ALTA E BAIXA.....	120

---

## LISTA DE FIGURAS

<b>Fig.3.1:</b>	Modelo Básico de um Sistema de Controle de Excitação. ....	6
<b>Fig.3.2:</b>	Diagrama de Blocos Básico de um Regulador de Tensão. ....	8
<b>Fig.3.3:</b>	Curva de Saturação do Excitador. ....	10
<b>Fig.3.4:</b>	Tensão de Campo/Estator. ....	11
<b>Fig.3.5:</b>	Tensão de Campo/Estator. ....	12
<b>Fig.3.6:</b>	Modelagem dos efeitos do torque e da excitação. ....	13
<b>Fig.3.7:</b>	Modelo do Gerador. ....	14
<b>Fig.4.1:</b>	Exemplo de Função de Pertinência. ....	15
<b>Fig.4.2:</b>	Proposta de um AVR e um PSS Fuzzy. ....	17
<b>Fig.5.1:</b>	Funções de pertinência da variável “e” do PSS Fuzzy. ....	19
<b>Fig.5.2:</b>	Funções de pertinência da variável “v” do PSS Fuzzy. ....	19
<b>Fig.5.3:</b>	Função de pertinência da variável “u” do PSS Fuzzy. ....	20
<b>Fig.5.4:</b>	Modelo simplificado de um AVR. ....	21
<b>Fig.5.5:</b>	Malha de controle com ARV clássico e PSS Fuzzy. ....	23
<b>Fig.5.6:</b>	Resposta da malha de controle com AVR clássico e PSS Fuzzy. ....	24
<b>Fig.5.7:</b>	Controlador Fuzzy Tipo Avanço/ Atraso de Fase. ....	25
<b>Fig.5.8:</b>	Programação do Fuzzy Logic Controller. ....	25
<b>Fig.5.9:</b>	Introdução das Regras no Fuzzy Logic Controller. ....	26
<b>Fig.5.10:</b>	Funções de Pertinência da Entrada “Level”. ....	28
<b>Fig.5.11:</b>	Funções de Pertinência da Entrada “Rate”. ....	28
<b>Fig.5.12:</b>	Funções de Pertinência Saída “Vs”. ....	29
<b>Fig.5.13:</b>	Malha de Controle com AVR Fuzzy e PSS Fuzzy. ....	30
<b>Fig.5.14:</b>	Resposta da Malha de Controle com AVR Fuzzy e PSS Fuzzy. ....	30
<b>Fig.6.1:</b>	Bancada de Ensaio. ....	32
<b>Fig.6.2:</b>	Desenho Esquemático de Montagem. ....	33
<b>Fig.6.3:</b>	Foto Ilustrativa do Sistema de Ensaio. ....	34
<b>Fig.6.4:</b>	Foto (“a”) de Circuito de Disparo de Bancada. ....	35
<b>Fig.6.5:</b>	Sistema para Controle de Velocidade. ....	36
<b>Fig.6.6:</b>	Curva de Magnetização do Sistema. ....	39
<b>Fig.6.7:</b>	Gráfico do Transdutor de Tensão. ....	40

---

<b>Fig.6.8:</b>	Processamento de Linearização do Sinal de Comando.....	42
<b>Fig.6.9:</b>	Modelo do Sistema de Regulação de Tensão.....	43
<b>Fig.6.10:</b>	Medidas para identificação de $G(s)$ .....	44
<b>Fig.6.11:</b>	Medidas para Identificação de $G_d(s)$ .....	45
<b>Fig.6.12:</b>	Diagrama de Blocos da Malha de Controle Adotada.....	46
<b>Fig.6.13:</b>	Resposta em Freqüência do Sistema Compensado.....	49
<b>Fig.6.14:</b>	Diagrama de Blocos de um <i>PSS</i> Convencional.....	50
<b>Fig.6.15:</b>	Diagrama Esquemático do <i>PSS Fuzzy</i> .....	51
<b>Fig.6.16:</b>	Detalhe dos Filtros Passa Alta e Passa Baixa do <i>PSS Fuzzy</i> .....	52
<b>Fig.6.17:</b>	Sinal de entrada (e) do <i>PSS Fuzzy</i> .....	52
<b>Fig.6.18:</b>	Sinal de entrada (v) do <i>PSS Fuzzy</i> .....	53
<b>Fig.6.19:</b>	Sinal de saída (c) do <i>PSS Fuzzy</i> .....	53
<b>Fig.6.20:</b>	<i>AVR</i> Clássico Implementado no <i>Simulink</i> .....	54
<b>Fig.6.21:</b>	Resposta Reais do <i>AVR</i> Clássico com Carga Plena.....	55
<b>Fig.6.22:</b>	Resposta do <i>AVR</i> Convencional em Meia Carga.....	56
<b>Fig.6.23:</b>	Resposta do <i>AVR</i> Clássico com Gerador a Vazio.....	57
<b>Fig.6.24:</b>	Resposta do <i>AVR</i> Clássico Implementado em “C” com Carga Plena. ..	58
<b>Fig.6.25:</b>	Resposta do <i>AVR</i> Tradicional “C” em Meia Carga.....	58
<b>Fig.6.26:</b>	Resposta do <i>AVR</i> Convencional em “C” e Gerador a Vazio.....	59
<b>Fig.6.27:</b>	Resposta do <i>AVR</i> Conv. Implementado em “C” a Variação de Carga. .	60
<b>Fig.6.28:</b>	Resposta do <i>AVR</i> Clássico com <i>PSS Fuzzy</i> .....	61
<b>Fig.6.29:</b>	Sinal de entrada (e) do <i>AVR Fuzzy</i> .....	62
<b>Fig.6.30:</b>	Sinal de entrada (v) do <i>AVR Fuzzy</i> .....	62
<b>Fig.6.31:</b>	Sinal de Controle (u) do <i>AVR Fuzzy</i> .....	62
<b>Fig.6.32:</b>	Resposta do <i>AVR Fuzzy</i> em Plena Carga.....	63
<b>Fig.6.33:</b>	Resposta do <i>AVR Fuzzy</i> com Meia Carga.....	64
<b>Fig.6.34:</b>	Resposta do <i>AVR Fuzzy</i> com Gerador a Vazio.....	65
<b>Fig.6.35:</b>	Resposta do <i>AVR Fuzzy</i> a Variação de Carga no Sistema.....	65
<b>Fig.6.36:</b>	Resposta do <i>AVR Fuzzy</i> com <i>PSS Fuzzy</i> a Variação de Carga.....	66
<b>Fig.6.37:</b>	Oscilografia do <i>AVR Fuzzy</i> com <i>PSS Fuzzy</i> com Variação de Carga. ..	67
<b>Fig.6.38:</b>	Paralelismo com <i>AVR</i> Convencional e sem <i>PSS</i> .....	68
<b>Fig.6.39:</b>	Paralelismo com <i>AVR</i> Clássico e com <i>PSS Fuzzy</i> .....	69
<b>Fig.6.40:</b>	Paralelismo com <i>AVR Fuzzy</i> e sem <i>PSS</i> .....	70
<b>Fig.6.41:</b>	Paralelismo com <i>AVR Fuzzy</i> e com <i>PSS Fuzzy</i> .....	70
<b>Fig. Anexo 1.1:</b>	Diagrama Esquemático do Regulador de Velocidade.....	76

<b>Fig. Anexo 1.2:</b>	Diagrama Frontal do Controlador Automático de Velocidade.....	77
<b>Fig. Anexo 1.3:</b>	Esquema Elétrico do Regulador de Velocidade.....	79
<b>Fig. Anexo 1.4:</b>	Lay Out da placa de circuito impresso.....	80
<b>Fig. Anexo 1.5:</b>	Mascara do Circuito Impresso.....	81
<b>Fig. Anexo 2.1:</b>	Ligação do Circuito do Fogo Girante .....	83

---

## LISTA DE TABELAS

<b>Tab.4.1:</b> Exemplo de Regras <i>Fuzzy</i> .....	16
<b>Tab.5.1:</b> Parâmetros para Simulação. ....	20
<b>Tab.5.2:</b> Conjunto de Regras Nebulosas.....	27
<b>Tab. Anexo 1.1:</b> Sinais do Regulador de Velocidade.....	78
<b>Tab. Anexo 1.2:</b> Lista de Materiais.....	82

---

**LISTA DE SÍMBOLOS E NOMENCLATURAS**

Kg	Ganho do gerador
Ve	Tensão de Entrada do Compensador
Td	Constante de Tempo do Derivador
Ta	Constante de Tempo do Excitador.
TR	Constante de Tempo do Filtro do Retificador
Tg	Constante de Tempo do Gerador
CLead/Lag	Controlador <i>Lead/Lag</i>
CPI	Controlador Proporcional-Integrador
Iexc	Corrente de Excitação
Wcg	Freqüência de Cruzamento de Ganho
SE	Função de saturação do excitador
Kd	Ganho do Derivador
Ka	Ganho do Excitador
KR	Ganho do Filtro do Retificador
Kg	Ganho do Gerador
tp	Instante de Pico
Mp	Máximo valor de ultrapassagem
a0, a1, b1	Parâmetros do Controlador Lead/Lag
ts	Tempo de Acomodação
td	Tempo de Atraso
tr	Tempo de Subida
EFD	Tensão de Campo
Vexc	Tensão de Excitação
VREF	Tensão de Referência
VR	Tensão de Saída do Regulador/Compensador
Vger.ff	Tensão Gerada Fase-Fase
Vdc	Tensão Retificada
Vt	Tensão Terminal da Máquina Síncrona

---

Tb, Tc	Constantes de Tempo do bloco Avanço e Atraso de Fase
Kp, Ki	Ganhos do Controlador Proporcional-Integrador
A/D	Conversor Analógico/Digital
Ag, Bg	Parâmetro para representação da saturação em pu
AVR	Automatic Voltage Regulators
C(s)	Controlador
CPSS	PSS convencional
D	Coeficiente de amortecimento
D/A	Conversor Digital/Analógico
e	Sinal de Erro
E'q	Tensão proporcional ao enlace de fluxo do campo em pu
E $\infty$	Tensão da barra infinita em pu
Efd	Tensão de campo em pu
Eq	Tensão atrás da reatância síncrona em eixo em quadratura em pu
ESP	Estabilizadores de Sistemas de Potência
f	Frequência de operação do sistema de potência
GESP(s)	Função de transferência do gerador + excitatriz + sistema de potência
H	Constante de Inércia em segundos
H(s)	Transdutor
Id	Corrente no eixo direito em pu
Ifd	Corrente de campo em pu
Iq	Corrente no eixo em quadratura em pu
Ir	Corrente ativa em pu
k1	Representa a mudança no torque elétrico para uma pequena variação no ângulo do rotor no eixo-d, ou seja, o coeficiente de torque sincronizante
k2	Representa a mudança no torque elétrico para uma pequena variação na tensão E'q, mantendo o ângulo do rotor constante
k3	É um fator de impedância que leva em conta o efeito da carga da impedância externa
k4	Parâmetro que representa o efeito desmagnetizante de uma variação no ângulo
k5	Parâmetro que representa a variação da tensão terminal para uma variação do ângulo do rotor com enlace de fluxo no eixo direito constante

---

$k_6$	Parâmetro que representa a variação da tensão terminal para uma variação do enlace de fluxo no eixo direito com ângulo do rotor constante
$k_e$	Ganho da excitatriz em pu
Lead/Lag	Avanço/atraso de fase
MF	Margem de Fase
$M_p$	Valor de Máximo Pico
$n$	Rotação da máquina em rpm
$p$	Pólos do gerador
$P(s)$	Processo
$P_e$	Potência elétrica da máquina em pu
$P_m$	Potência mecânica da máquina em pu
<i>PSS</i>	<i>Power System Stabilizer</i>
<i>PSS(s)</i>	Função de transferência do estabilizador de sistema de potência (ESP)
$s$	Operador de Laplace
SEP	Sistemas Elétricos de Potência
sp	Set Point.
$T'd_0$	Constante de tempo transitório a circuito aberto segundo o eixo direito em segundos
$T_{am}$	Conjugado de amortecimento em pu
$T_e$	Constante de tempo da excitatriz em segundos
$T_{mec}$	Conjugado mecânico em pu
$\omega$	Velocidade angular da máquina em rad/s
$\omega_n$	Frequência natural de oscilação da máquina em rad/s
$\omega_0$	Frequência de oscilação da máquina contra barra infinita em rad/s
$X'd$	Reatância transitória de eixo direito em pu
$X_d$	Reatância síncrona do eixo direito em pu
$X_e$	Reatância equivalente do sistema em pu
$X_q$	Reatância síncrona do eixo em quadratura em pu
$y$	Sinal de Saída
$\zeta$	Ângulo de carga da máquina em radianos
$\Theta$	Coefficiente de amortecimento

---



## RESUMO

Um procedimento fundamental para garantir a qualidade de energia em sistemas elétricos é a regulação de tensão. Contudo essa não é uma atividade trivial devido ao grande número de variações de estado que um sistema elétrico é submetido. A compensação eficaz de eventuais distúrbios é crucial para garantir a operação correta de um sistema de energia elétrica.

No sentido de melhorar este processo fundamental existem algumas estratégias que empregam reguladores e estabilizadores convencionais, e outras que utilizam técnicas mais elaboradas.

Devido as características não lineares intrínsecas a sistemas de geração de energia elétrica, nem sempre os controladores convencionais apresentam comportamentos satisfatórios em condições de operação de um sistema real.

Este trabalho usará lógica nebulosa para implementar um regulador automático de tensão e um estabilizador de sistema de potência para geradores elétricos, objetivando melhorar o processo de geração em sistemas elétricos.

Resultados obtidos através de simulações computacionais e de experimentos reais indicaram uma melhoria em relação à utilização de sistemas convencionais.

---

## ABSTRACT

A fundamental procedure to guarantee the energy quality in electric systems is the tension regulation. However, it is not a trivial activity due to the great number of state variations that an electric system is submitted. The effective compensation of eventual disturbances is crucial to guarantee the correct operation of an electric power system.

In the sense of improving this fundamental process exist some strategies that use regulators and conventional stabilizers, and other that use techniques more elaborated.

Due the non-linear characteristics of electric systems, not always conventional controllers have satisfactory behaviors in conditions of real operation in power systems.

This work will use Fuzzy Logic to implement an automatic regulator of tension and a power system stabilizer for electric generators, aiming to improve the generation processes in electric systems.

Results obtained through computational simulations and of real experiments indicated an improvement regarding the utilization of conventional systems.

---

## CAPÍTULO 1 - INTRODUÇÃO

Em sistemas elétricos de potência mudanças em estados operacionais são comuns, principalmente na comutação de linhas, de cargas, de grupos geradores, etc. Essas alterações devem ser compensadas de forma eficaz garantindo a estabilidade e padrões de qualidade do sistema elétrico. Nas máquinas elétricas que compõem um sistema real, as variações de tensão decorrentes dessas perturbações são corrigidas e compensadas por Reguladores Automáticos de Tensão (*Automatic Voltage Regulator - AVR*).

Para garantir determinadas condições de estabilidade são empregados os Estabilizadores de Sistema de Potência (*Power System Stabilizer - PSS*).

As estruturas usuais destes reguladores e estabilizadores utilizam compensadores convencionais como controladores PID (Proporcionais, Integrais e Derivativos), ou de avanço ou atraso de fase (*Lead/Lag*).

Esse trabalho propõe a utilização de lógica nebulosa (*Fuzzy*) na implementação de um AVR e um PSS com características de avanço ou atraso de fase, visando a compensação de características não-lineares de sistemas de geração reais.

O trabalho está organizado nos capítulos descritos a seguir. O capítulo 2 contém uma revisão bibliográfica sobre reguladores de tensão e estabilizadores de sistema de potência em geral. Neste capítulo serão comentadas as principais dificuldades existentes no controle de tensão, as formas de abordagem tradicionais e a referencia de algumas técnicas mais elaboradas de tendências atuais.

O capítulo 3 mostra os conceitos envolvidos na modelagem dos reguladores de tensão, os principais componentes de uma máquina síncrona, os tipos usuais de sistemas de excitação e a modelagem clássica desses sistemas.

No capítulo 4 está apresentada a conceituação da lógica nebulosa, teoria proposta por Zadeh em 1965, e algumas aplicações no contexto deste trabalho.

No capítulo 5 é mostrado como foi elaborado o AVR e o PSS *Fuzzy* propostos e suas respectivas simulações no contexto citado.

No capítulo 6 tem-se o desenvolvimento do trabalho prático realizado para comprovar as vantagens da utilização da estratégia proposto em relação a compensadores convencionais.

As conclusões do trabalho são mostradas no capítulo 7, assim como as sugestões para possíveis trabalhos futuros.

---

Ao final da dissertação estão anexos e descrições com folhas de dados dos sistemas utilizados, esquemas de montagens e os programas desenvolvidos na elaboração do trabalho.

---

## CAPÍTULO 2 - REVISÃO BIBLIOGRÁFICA

Para entender a função de um regulador automático de tensão aplicado a um gerador elétrico é importante conhecer primeiro as características básicas das máquinas síncronas. O que inclui estabelecer um modelo adequado com detalhes suficientes para efetuar as análises e estudos necessários ao contexto proposto. Os principais modelos de sistema de excitação e regulação de tensão são padronizados pelo IEEE que estabeleceu os modelos de referência em 1968. A obtenção de modelos práticos geralmente inclui informações mais precisas, conforme descrito em Leandro (2004). Alguns textos tratam estas questões com informações detalhadas como em Anderson e Fouad (1993).

Esse é um assunto atual e complexo e ainda hoje é motivo de inúmeras pesquisas. Um sistema elétrico de potência requer um balanço contínuo entre a geração de energia e as demandas variáveis de carga, sem perder o controle da frequência e dos níveis de tensão desejados para o sistema. Esse controle não é trivial, pois as variações presentes em uma rede elétrica real são diversas e muitas vezes ocorrem em uma ampla faixa e de forma não balanceada. Muitas variáveis influenciam esse comportamento, como por exemplo, o número de unidades geradoras e linhas de transmissão em operação, o que resulta em um sistema altamente complexo e com dinâmicas não-lineares (Venayagamoorthy, 2003).

Em sua maioria, os reguladores de tensão tradicionais operam com parâmetros constantes, sendo influenciados por ações não-lineares de um processo real. Usualmente são utilizadas técnicas de controle clássico baseadas em modelos linearizados. Essas técnicas são adequadas para regiões de operações restritas. Caso exista um distúrbio que faça uma máquina ultrapassar o limite dessas regiões operacionais, os sistemas de controle convencionais deixam de ser eficientes. Para conseguir melhores resultados no controle de sistemas não-lineares foram desenvolvidos outros tipos de controladores, como por exemplo, os controladores adaptativos. Estes têm suas leis de controle ou seus ganhos ajustados por métodos de adaptação que utilizam técnicas de identificação de sistemas. No entanto essas técnicas de estimação de parâmetros resultam em implementações de controles mais elaboradas e complexas (Harris e Billings, 1981). Outra forma de me-

---

lhorar o controle de sistemas não-lineares é utilizando técnicas de lógica difusa (Pinheiro, 2000).

Com a dificuldade de elaborar um modelo que represente todas as características de um processo, a maior parte dos sistemas de controle utiliza as propriedades mais relevantes dos modelos. Contudo para sistemas não-lineares a modelagem clássica não descreve bem a dinâmica dos sistemas, e sim apenas uma determinada vizinhança em torno do ponto de operação desejado no projeto. Assim, nos casos onde o sistema pode operar em regiões distantes do ponto de operação, os modelos tradicionais não são, geralmente, adequados.

Os sistemas que utilizam lógica nebulosa constituem uma possível solução para esse tipo de problema. A idéia básica consiste da descrição aproximada de um sistema não-linear que represente o comportamento da planta em diferentes pontos de operação (Junior, W. B., 2005).

O desenvolvimento de novas áreas da teoria de controle, assim como de novas tecnologias, permite que novas metodologias de compensação possam ser aplicadas a sistemas elétricos de potência, visando comportamentos mais estáveis e melhorias na eficiência destes sistemas.

O modelo de sistema de controle que esta dissertação propõe é a utilização de um controlador nebuloso que apresente características de compensação de avanço ou atraso de fase, a ser aplicado a um regulador de tensão e a um estabilizador de sistemas de potência.

O objetivo é obter um melhor desempenho frente aos sistemas convencionais de compensação que geralmente não apresentam boas respostas a variações de parâmetros e não linearidades inerentes a sistemas elétricos de potência.

Resultados experimentais mostrarão que a estrutura proposta é eficaz e pode melhorar a eficiência de sistemas de geração.

Nos próximos capítulos será apresentada conceituação sobre modelagem de máquinas síncronas, reguladores automáticos de tensão e sistemas de estabilização. Será apresentada também a proposta de um compensador nebuloso com características de avanço ou atraso de fase, que será utilizado para implementar um AVR e um PSS.

---

## CAPÍTULO 3 - SISTEMA DE CONTROLE DE EXCITAÇÃO

### 3.1. Introdução

Um sistema de controle de excitação é estruturalmente composto pelo regulador automático de tensão (*AVR*), excitatriz e máquina síncrona. Geralmente nas referências técnicas um sistema de excitação se difere de um sistema de controle de excitação por não incluir a máquina síncrona na sua representação.

O gerador síncrono é responsável pela produção de energia elétrica, sendo normalmente acionado por uma turbina. A turbina, por sua vez, tem a velocidade de rotação controlada por um regulador de velocidade que atua basicamente nas válvulas de admissão de gás, vapor ou água, conforme o tipo de turbina considerada.

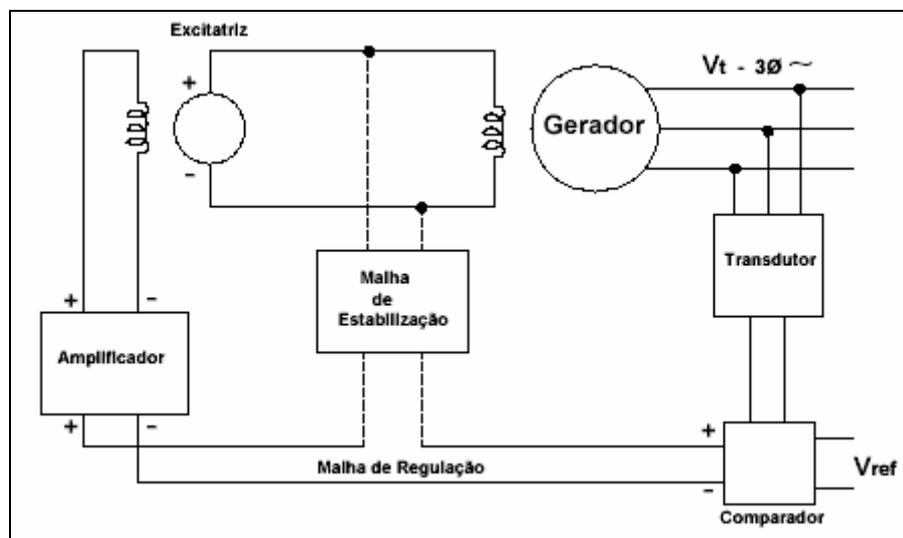
Os geradores síncronos, de uma forma geral, são compostos por duas partes magnéticas principais: o estator e o rotor. No estator são dispostos enrolamentos que conectam a máquina ao sistema elétrico de potência. Por eles circulam correntes elétricas que permitem suprir a potência demandada pelo sistema. No rotor são encontrados dois tipos de enrolamentos: o enrolamento de campo, que é responsável pela excitação da máquina e conseqüentemente pela produção do fluxo magnético de excitação; e os enrolamentos amortecedores, que têm papel importante no amortecimento de oscilações de origem mecânica e na qualidade do sinal de tensão obtido no estator.

No enrolamento de campo da máquina circula uma corrente contínua, suprida pelo sistema de excitação. A ação desta corrente juntamente com o movimento do rotor, produz uma tensão induzida no estator. Caso a máquina esteja ligada ao sistema, irá circular corrente alternada pelos enrolamentos do estator. Esta corrente produz um fluxo de reação no próprio estator que tende a se opor ao fluxo magnético de excitação. A composição destes dois fluxos tem efeitos diferentes nos enrolamentos do estator e do rotor. O fluxo magnético de excitação, que é produzido pelo rotor, e o fluxo magnético de reação do estator têm magnitudes constantes e giram com a velocidade do rotor. Assim, a composição dos dois fluxos dá origem a um fluxo resultante, que é estacionário com relação ao rotor, mas que gira com velocidade síncrona com relação ao estator.

---

O regulador de tensão tem a finalidade de alimentar o enrolamento de campo do gerador com corrente contínua necessária para a produção do fluxo magnético dentro da máquina, assim como controlar a tensão terminal gerada.

Os tipos de sistemas de excitação podem ser classificados em sistema de excitação rotativo e sistema de excitação estático. Um sistema do tipo rotativo considera a presença de excitatriz, sendo esta um gerador de corrente contínua ou um alternador. Já um sistema de excitação do tipo estático considera a utilização de chaves estáticas controladas, dispensando unidades rotativas. Para esse estudo será utilizado o sistema estático de excitação. No item 3.2 serão apresentados os sistemas de excitação mais utilizados. A Figura 3.1 ilustra o diagrama básico de um sistema de controle de excitação típico.



**Fig.3.1:** Modelo Básico de um Sistema de Controle de Excitação.

### 3.2. Tipos de Sistema de Excitação

Os sistemas de excitação são comumente classificados pelo tipo de excitatriz. A excitatriz é a componente responsável por fornecer a corrente de campo necessária para magnetizar os pólos do gerador síncrono, (Almeida, 2000).

De forma geral podemos dividir os sistemas de excitação em quatro tipos: de excitatriz rotativa; estática; brushless; auto-reguladas.

Em máquinas com excitações rotativas exista um gerador de corrente contínua auto-excitado montado geralmente sobre o eixo da própria máquina. A corrente de campo pode ser controlada através de um reostato ligado em série com o campo do gerador.

A excitatriz estática é um dos tipos mais utilizados na atualidade. Este sistema não utiliza geradores de corrente contínua, e o regulador de tensão é normalmente montado em conjunto com os componentes da excitatriz estática. O campo do gerador principal é conectado através de escovas e anéis a uma ponte conversora controlada. De forma geral, os sistemas de regulação empregam o valor atual da tensão dos terminais da armadura que é comparado com uma tensão de referência. Se houver uma diferença resultante (erro), o sistema de comando da ponte conversora agirá no sentido de diminuir o erro processado.

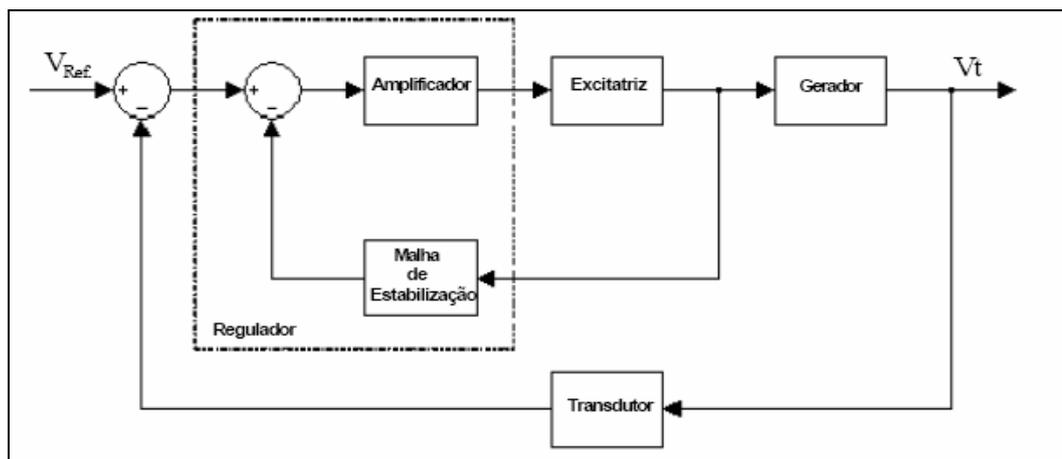
A terceira forma de construção de uma excitatriz é a brushless, que é na verdade uma máquina composta por um alternador de pólos fixos e armadura girante, mais uma ponte retificadora. Assim como nas máquinas rotativas, esse conjunto fica montado no próprio eixo do gerador. Como a armadura da excitatriz, ponte retificadora, os pólos do gerador principal e seus enrolamentos giram em conjunto, não existe a necessidade de escovas e anéis de comutação, o que é uma significativa vantagem em termos de manutenção. O regulador automático de tensão controla a corrente de campo através do controle do valor da corrente de excitação que circula na excitatriz.

Outro método de excitação utilizado são as máquinas com excitatriz auto-reguladas onde a é inserido um circuito composto de diodos e resistores que ligam a bobina de campo aos terminais de saída da armadura do gerador. Como a tensão dos terminais de saída fornece energia ao circuito de campo, pode-se estabelecer o valor final da tensão que será gerada. Naturalmente essa regulação não é linear e depende muito da faixa de operação da máquina síncrona.

---

### 3.3. Modelagem de um Sistema de Excitação

Como visto anteriormente, o sistema de excitação é um conjunto de equipamentos utilizados em uma central de geração de energia, objetivando garantir os valores de tensão adequados nos terminais de saída das unidades reguladoras. A figura 3.2 ilustra o diagrama de blocos de um regulador de tensão.



**Fig.3.2:** Diagrama de Blocos Básico de um Regulador de Tensão.

O Regulador de tensão tem seu funcionamento baseado no controle da corrente de campo da excitatriz do gerador. Variando a corrente de excitação das máquinas é possível controlar a tensão gerada.

Nos reguladores de tensão são estabelecidas as funções de controle e proteção, para garantir o desempenho operativo dos geradores e dos sistemas de potência aos quais estão conectados.

Na Figura 3.2 a entrada  $V_{Ref}$  é a tensão de referência da malha de controle e a tensão  $V_t$  é a tensão nos terminais do gerador. Observando a figura é possível verificar que os sistemas de excitação são compostos basicamente por quatro componentes:

- O excitador, que fornece corrente para o enrolamento de campo;
- O conjunto de amplificadores que têm por objetivo amplificar os sinais vindos da malha de controle;

- A malha de controle, que tem a finalidade de tornar o sistema estável;
- O transdutor que mede a tensão na saída do gerador e realimenta esta informação a malha de controle.

É importante lembrar que normalmente os sistemas de controle de tensão são extremamente complexos. Além dos sistemas básicos existem ainda outros como os estabilizadores de tensão que injetam sinais adicionais no regulador, no intuito de diminuir os efeitos de oscilações de baixa frequência no sistema. Outro sistema indispensável e que não consta no diagrama de blocos é a parte referente as proteções.

É mais fácil realizar a modelagem de um sistema de excitação subdividindo o mesmo nos seus blocos básicos e nas suas funções constituintes.

### **Transformador de Medida e Retificador de Instrumentação**

A tensão retificada  $V_{dc}$  é diretamente proporcional à tensão terminal  $V_t$  e depende também da relação  $K_R$  do transformador de medida. A função de modelagem resultante é dada por (3.1), onde  $T_R$  é a constante de tempo do filtro do retificador.

$$V_{dc} = \frac{K_R \cdot V_t}{T_R \cdot s + 1} \quad (3.1)$$

### **Subtrator**

O sinal  $V_e$  entregue ao compensador de tensão é proporcional à diferença entre o sinal de referência  $V_{Ref}$  e a informação de realimentação  $V_{dc}$  conforme (3.2), onde  $K$  é um ganho (geralmente unitário).

$$V_e = K \cdot (V_{Ref} - V_{dc}) \quad (3.2)$$


---

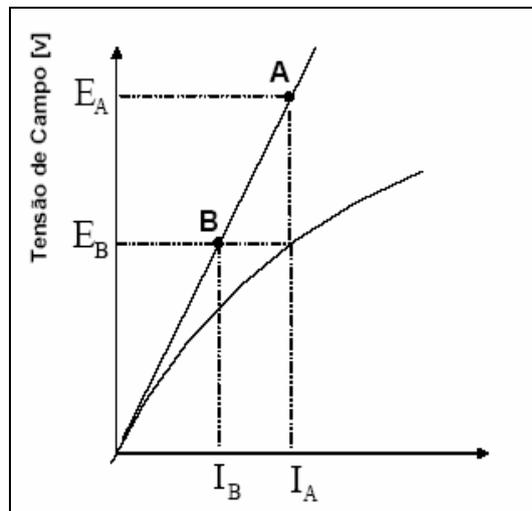
## Excitator

A equação da tensão de campo  $F_{DE}$  é dada por (3.3), onde  $S_E$  é a função (3.4) de saturação do excitador.

$$E_{FD} = \frac{(V_R - E_{FD} \cdot S_E)}{T_a \cdot s + 1} \cdot K_a \quad (3.3)$$

$$S_E = \frac{(I_A - I_B)}{I_B} \quad (3.4)$$

A figura 3.3 mostra a curva de saturação do excitador.



**Fig.3.3:** Curva de Saturação do Excitador.

A reta definida pelos pontos A e B representa o modelo linear do processo. A outra curva traduz a resposta real do mesmo. Uma das possíveis não-linearidades presentes em um gerador, a curva de magnetização, deve a efeitos de saturação magnética.

Para efeito de simplificações considerou-se efeitos lineares nas situações operacionais, e por isso os efeitos da saturação não foram considerados para efeito de cálculos em geral.

## Compensador

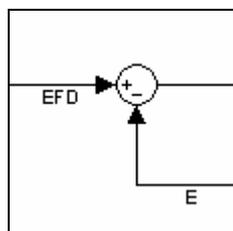
A equação (3.5) ilustra o modelo de um compensador *Lead/Lag* clássico.

$$C(z) = \frac{\tau_c s + 1}{\tau_B s + 1} \quad (3.5)$$

## Gerador

Segundo Anderson e Fouad (1993), a análise e modelagem de máquinas síncronas é baseada nas equações diferenciais que descrevem o comportamento destes sistemas no tempo. O modelo típico consiste de uma máquina síncrona conectada a um barramento infinito através de uma reatância. Para modelar o gerador usa-se como referência a tensão de campo (EFD) e a tensão dos terminais do estator (E). Contudo, existe um tempo de atraso entre uma alteração na tensão de campo e a resposta encontrada nos terminais do estator. Esse tempo é designado como  $T_{do}$ . A relação (3.6) engloba estas grandezas, onde  $E'_q$  é a tensão no eixo em quadratura da máquina e está representada na Figura 3.4.

$$E'_q = \frac{EFD - E}{T_{do}} \quad (3.6)$$



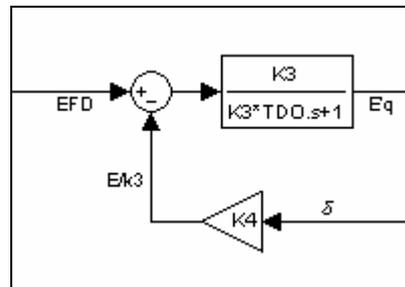
**Fig.3.4:** Tensão de Campo/Estator.

Contudo esses sinais não têm a mesma escala de grandeza. Para resolver esse problema é necessário usar fatores de correção que são definidos por (3.7) e (3.8).

$$k_4 = \frac{1}{k_3} \cdot \frac{E}{\delta} \quad (3.7)$$

$$T_{do} \cdot E'_q = \frac{k_3}{1 + k_3 T_{do}} \cdot (EFD - k_4 \cdot \delta) \quad (3.8)$$

Uma vez definidos estes parâmetros tem-se na Figura 3.5 uma representação mais adequada.  $K_3$  é um fator que leva em conta o efeito da carga da impedância externa.  $K_4$  está relacionado ao efeito desmagnetizante de uma mudança no ângulo do rotor, e  $\delta$  é o ângulo do sistema elétrico.



**Fig.3.5:** Tensão de Campo/Estator.

Para estimar o valor do ângulo é necessário encontrar a velocidade angular do rotor do gerador. A equação que representa essa velocidade é dada por (3.9), onde  $T_e$  é o torque elétrico do gerador,  $T_m$  é o torque mecânico da turbina,  $D$  é o amortecimento, e  $H$  é a energia armazenada em razão da velocidade do rotor.

$$\dot{\omega} = \frac{(T_m - T_e) \cdot \delta \cdot \omega_R}{2 \cdot H} \quad (3.9)$$

Para a potência elétrica podemos considerar (3.10).

$$P_e = V_d \cdot \vec{I}_d + V_q \cdot \vec{I}_q \quad (3.10)$$

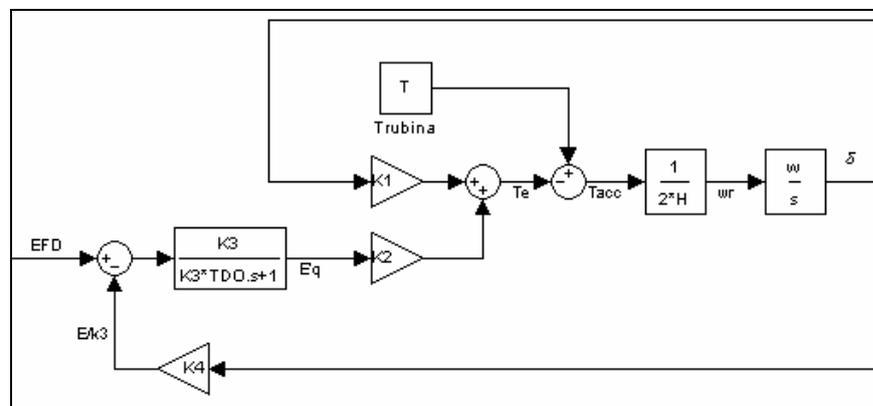
Considerando  $V_d = -x_q \cdot I_q$  e  $V_q = x_d \cdot I_d + E'_q$  resulta no torque elétrico incremental expresso por (3.11):

$$T_e = k_1 \cdot \delta + k_2 \cdot E'_q \quad (3.11)$$

Nesse caso  $k_1$  representa a mudança no torque elétrico para uma pequena variação no ângulo do rotor, enquanto  $k_2$  representa a mudança no torque elétrico para uma pequena variação na tensão no eixo de quadratura.

Com o torque elétrico definido e o torque mecânico sendo fornecido pela turbina, conclui-se que o torque acelerante será a diferença entre eles. A partir do torque acelerante é possível encontrar a velocidade angular e também o ângulo de regulação da máquina síncrona, como em (3.12). A Figura 3.6 ilustra o modelo resultante decorrente das modelagens citadas.

$$T_{acc} = J \cdot \frac{d^2 \cdot \delta}{dt^2} = T_m - T_e \quad (3.12)$$

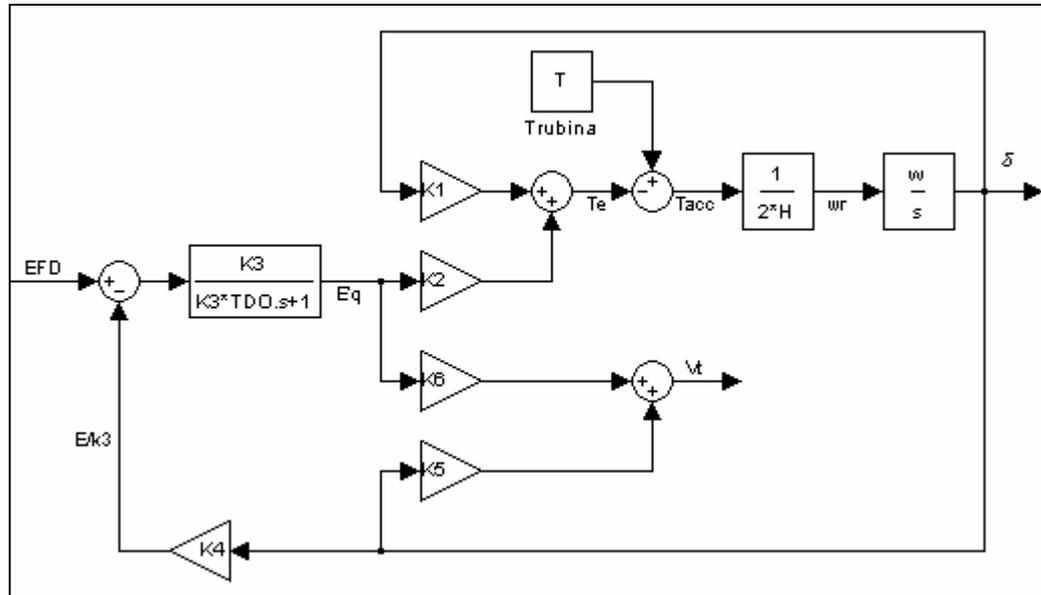


**Fig.3.6:** Modelagem dos efeitos do torque e da excitação.

A equação que modela a tensão terminal na máquina síncrona é dada por (3.13).  $k_5$  representa uma modificação na tensão terminal para uma pequena mudança no ângulo do rotor.  $k_6$  expressa a mudança na tensão terminal para uma pequena variação no eixo de quadratura.

$$V_t = k_5 \cdot \delta + k_6 \cdot E'_q \quad (3.13)$$

De maneira similar ao procedimento anterior, deve-se acrescentar esses parâmetros ao digrama de blocos do modelo do gerador. Assim, a representação completa do gerador está indicada na Figura 3.7.



**Fig.3.7:** Modelo do Gerador.

## CAPÍTULO 4 - CONCEITOS DE LÓGICA FUZZY

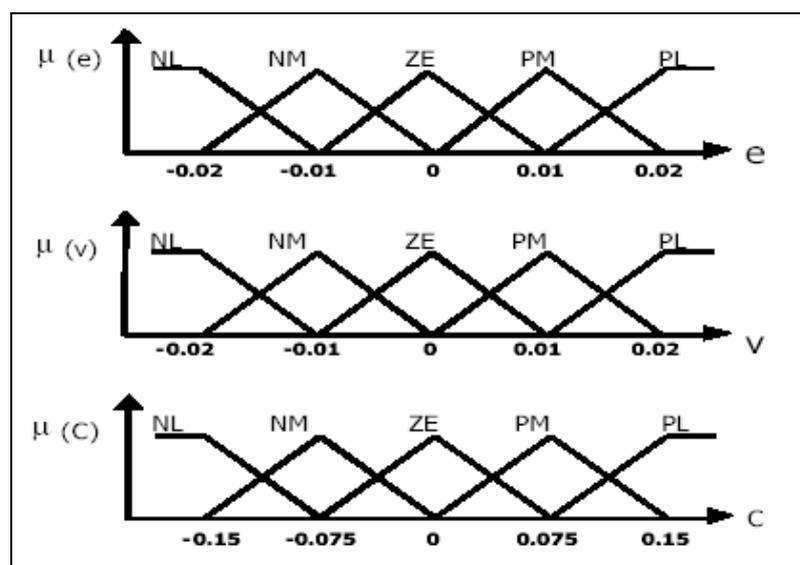
### 4.1. Conceituação Básica

Os conceitos sobre lógica difusa foram estabelecidos a partir de meados da década de sessenta, quando Lofti A. Zadeh (1965) formalizou a teoria dos conjuntos nebulosos (*Fuzzy Sets*). Mas foi a partir da década de setenta que a utilização da lógica nebulosa tornou-se muito difundida em projetos de sistemas de controle, principalmente por sua capacidade de compensar características não-lineares de sistemas de controle de uma forma mais simples e eficaz do que com controladores clássicos.

A primeira aplicação da teoria de Zadeh em sistemas de controle foi feita no trabalho de Mamdani e Assilian (1974), onde foi empregada para controlar uma caldeira a vapor, nas quais os controladores convencionais apresentavam dificuldades de sintonia devido aos efeitos não-lineares existentes no processo.

Em Lee (1990) é apresentada uma revisão ampla de conceitos sobre conjuntos nebulosos, Lógica *Fuzzy* e suas aplicações em sistemas de controle em geral.

A Figura 4.1 ilustra um modelo de função de pertinências com seus universos de discurso correspondentes. A escolha correta das funções de pertinências, bem como as das regras nebulosas e ganhos de escalonamento, definem as características de um sistema de controle difuso.



**Fig.4.1:** Exemplo de Função de Pertinência.

Para formar as regras nebulosas, as variáveis utilizadas são combinadas através de proposições condicionais do tipo “*if then*”, e são associadas através de conectivos lógicos que executam operações básicas nestes conjuntos. Como exemplo pode-se citar os operadores **MIN** e **MAX** que operam os valores mínimo e máximo, respectivamente, nos valores de pertinência correspondentes.

Outro operador bastante utilizado é o **Produto**, que como o nome indica realiza o produto das pertinências processadas. As regras nebulosas podem ser representadas conforme indicado em (4.1).

$$R_n: \text{IF } e \text{ IS } A \text{ AND } v \text{ IS } B \text{ THEN } C \text{ IS } D \quad (4.1)$$

Onde “*e*” e “*v*” são as variáveis dos antecedentes das regras, “*u*” o conseqüente e A, B, C os correspondentes conjuntos nebulosos. A conversão das variáveis nos correspondentes valores de pertinência  $\mu(\cdot)$ , é conhecida como fuzificação.

A equação (4.1) também pode ser representada de forma tabular (Tabela 41), onde os conjuntos difusos são designados linguisticamente como negativo largo (NL), negativo médio (NM), zero (ZE), positivo médio (PM) e positivo largo (PL).

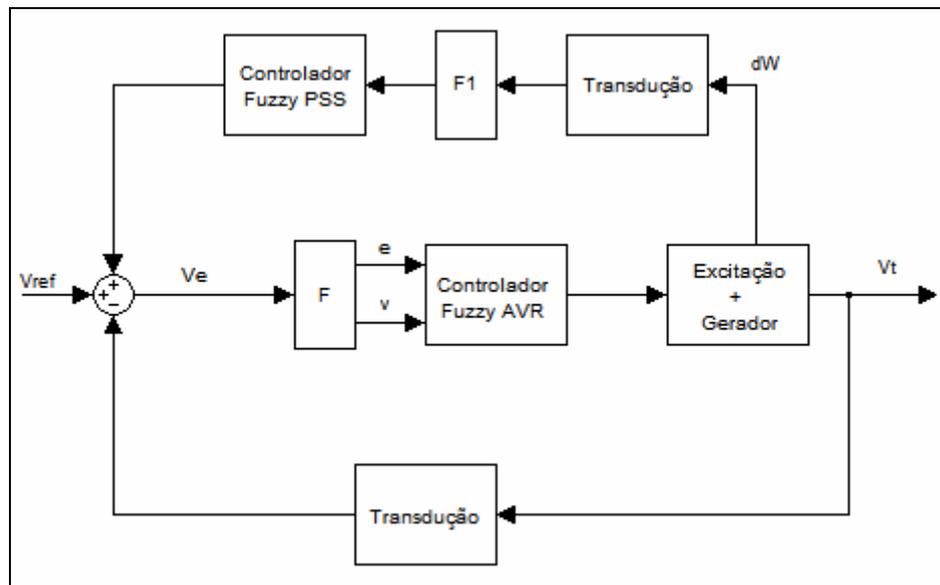
**Tab.4.1:** Exemplo de Regras Fuzzy.

c		e				
		NL	NM	ZE	PM	PL
v	NL	NL	NL	NL	NM	ZE
	NM	NL	NL	NM	ZE	PM
	ZE	NL	NM	ZE	PM	PL
	PM	NM	ZE	PM	PL	PL
	PL	ZE	PM	PL	PL	PL

Realizada a computação das regras é efetuada a etapa de defuzificação, onde as variáveis lingüísticas são transformadas em valores numéricos novamente. Para obter um valor que melhor represente a composição das regras ativas, são empregados vários métodos de defuzificação, entre os quais pode-se citar o valor máximo, o método do centróide, etc.

#### 4.2. Proposta de Sistema AVR – PSS utilizando Lógica Fuzzy

A proposta desta dissertação é utilizar regras difusas para implementar um AVR e um PSS eficaz. A Figura 4.2 ilustra a proposição. Serão utilizados compensadores difusos com características de atraso e avanço de fase. Os sinais de entrada são fornecidos através filtros (F e F1) com funções passa-altas e passa-baixas. A figura citada mostra a estrutura adotada. O filtro F fornece os sinais de entrada para o compensador AVR Fuzzy que será responsável por comandar a excitação do gerador. A grandeza controlada será a tensão no terminal da máquina síncrona ( $V_t$ ) realimentada pelo um transdutor correspondente, que será subtraída do sinal de referência ( $V_{Ref}$ ) gerando a informação do erro ( $V_e$ ) da malha. Para o PSS Fuzzy os sinais de entrada são provenientes da informação da variação da frequência (ou rotação) do conjunto gerador. Essa informação da entrada no filtro F1 é processada pelo compensador difuso, depois é somada junto com a entrada de referência da malha de controle compondo o sinal estabilizante do sistema.



**Fig.4.2:** Proposta de um AVR e um PSS Fuzzy.

## CAPÍTULO 5 - SIMULAÇÃO

### 5.1. Introdução

Neste capítulo será apresentada a simulação de um sistema de controle desenvolvido com lógica nebulosa, aplicado no controle de tensão de um gerador síncrono. O objetivo é projetar um regulador de tensão *fuzzy* que apresente um desempenho melhor que os reguladores tradicionais elaborados com controladores convencionais, onde os parâmetros de controlador são fixos de maneira a garantir um determinado desempenho em torno de um ponto de operação nominal do sistema. O desempenho geral pode não se manter, pois devido a características não-lineares do processo, o sistema pode não oferecer o mesmo desempenho quando o ponto de operação é diferente daquele para o qual a malha de controle foi sintonizada.

### 5.2. Modelagem *PSS Fuzzy*

Parte da estabilidade de um sistema elétrico de potência é comumente estabelecida com a utilização de um *PSS*. Sistemas de geradores interligados podem apresentar oscilações não desejadas que devem ser minimizadas. Oscilações em baixa frequência (entre 0,2 e 2 Hz) geralmente ocorrem após perturbações em sistemas elétricos interligados. Contudo grandes perturbações como a entrada de novas cargas ou saídas de unidades geradoras de grande porte, podem gerar perturbações de características muito diversificadas, não sendo possível compensá-las sem o uso de um controle específico e de atuação mais incisiva no processo. Em Féo (2004) foi apresentado o projeto de um compensador difuso com características de avanço/atraso de fase na estratégia de controle de um *PSS*.

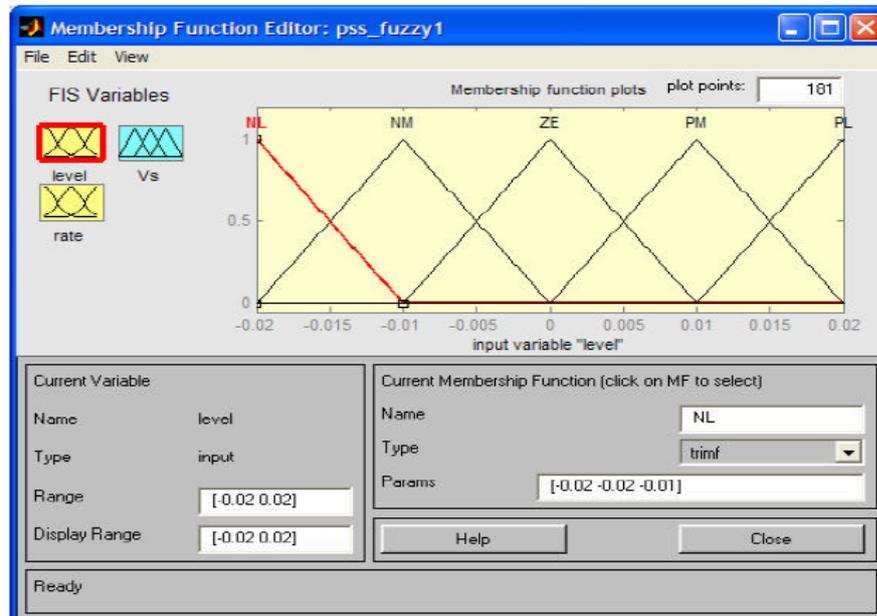
O próximo item deste capítulo mostrará a simulação da estrutura de controle difuso com características de avanço e atraso de fase para realizar um *AVR* convencional e um *PSS Fuzzy*. A ferramenta de simulação a ser utilizada será o software *MatLab*.

Para realizar a simulação de um *PSS Fuzzy* foi utilizado o modelo matemático mostrado no capítulo 3 (figura 3.7 Modelo do Gerador) com um compensador tra-

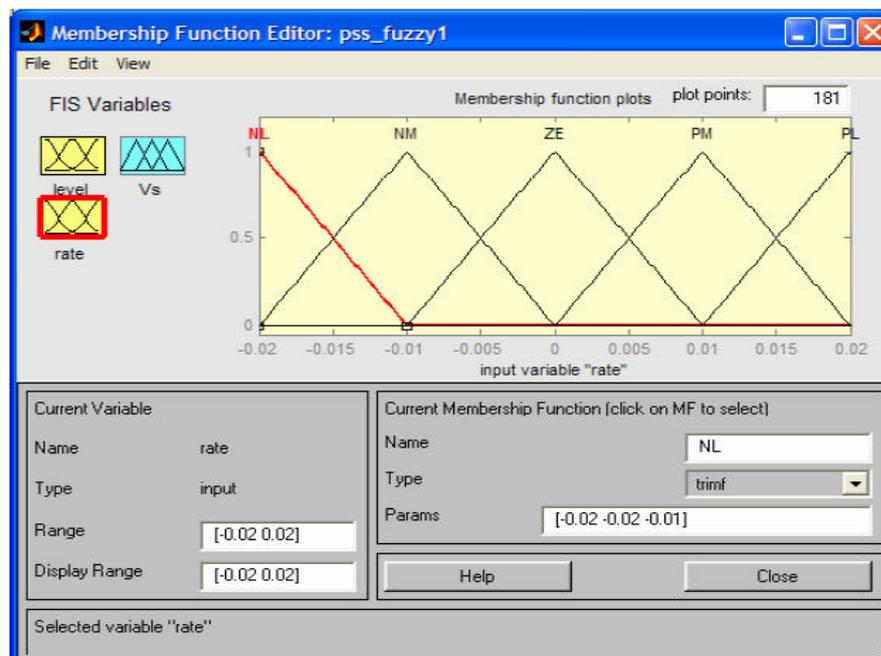
---

dicional. Os valores dos parâmetros usados no exemplo a ser simulado estão descritos na Tabela 5.1 e são típicos de um sistema de excitação.

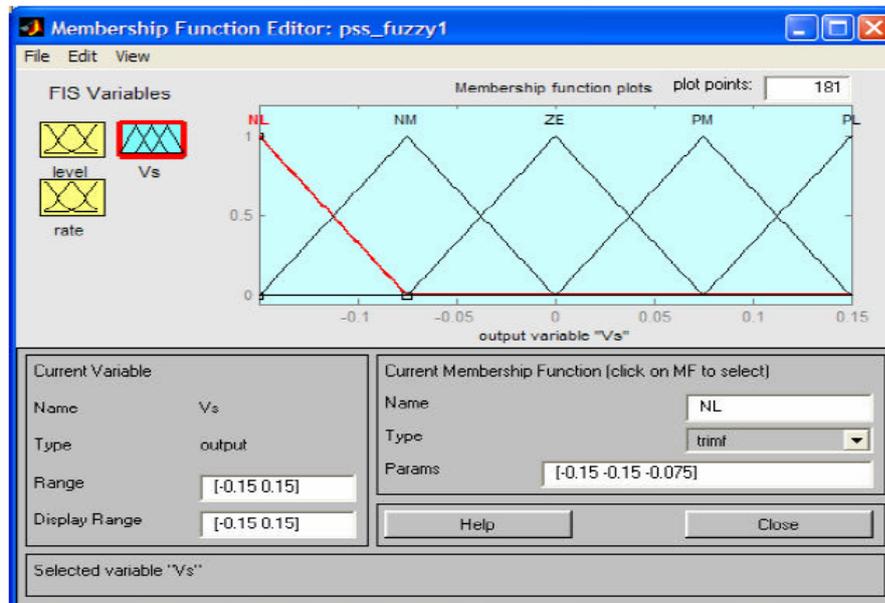
Na implementação do *PSS Fuzzy* foram utilizadas as funções de pertinência ilustradas nas Figuras 5.1 e 5.2 para as variáveis de entradas do controlador. Para a saída do *PSS* nebuloso foi utilizada a pertinência mostrada na Figura 5.3.



**Fig.5.1:** Funções de pertinência da variável “e” do *PSS Fuzzy*



**Fig.5.2:** Funções de pertinência da variável “v” do *PSS Fuzzy*.



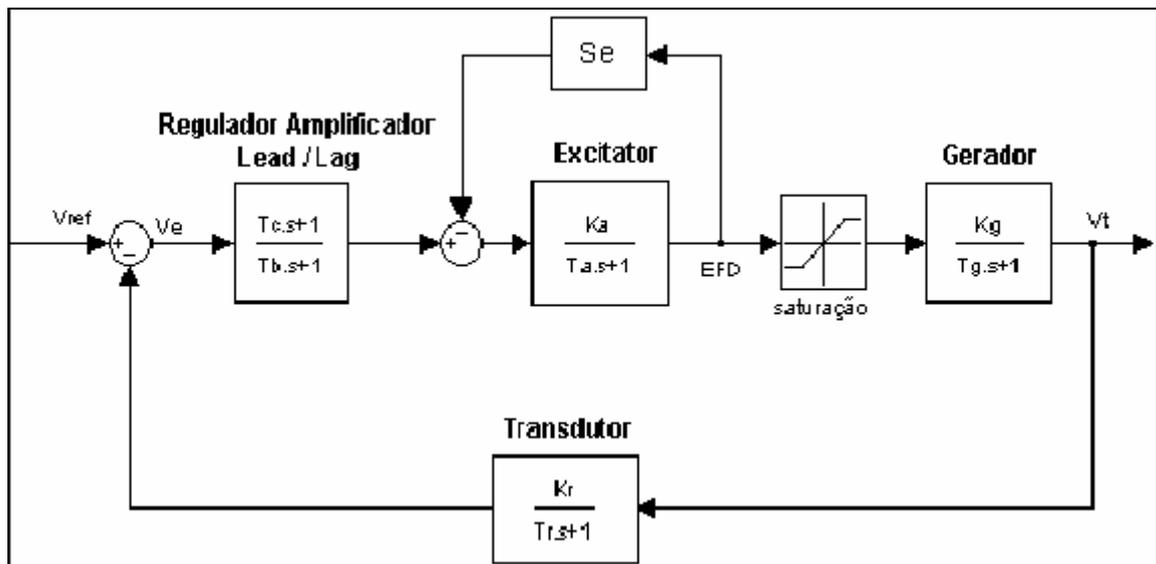
**Fig.5.3:** Função de pertinência da variável “u” do *PSS Fuzzy*.

**Tab.5.1:** Parâmetros para Simulação.

$T_a=0.05$	$T_E=0.95$	$T_{DO}=5.9$	$T_F=1.0$	$T_R=0.0$	$H_2=4.74$
$K_1=1.4479$	$K_2=1.3174$	$K_3=0.3072$	$K_4=1.8052$	$K_5=0.0294$	$K_6=0.5257$
$K_a=400$	$K_E=0.17$	$K_F=0.04$	$K_R=1.0$	$D=2.0$	$F=60$
		$T_M=1.0[\text{pu}]$ em $T=8[\text{s}]$			

### 5.3. Modelagem do AVR Clássico com *PSS Fuzzy*

Para facilitar a modelagem do compensador *AVR* tradicional, foi utilizado um modelo simplificado do gerador e da excitatriz, onde ambos podem ser aproximados por funções de transferência de primeira ordem (Anderson e Fouad, 1993), e a função de saturação (figura 5.4).



**Fig.5.4:** Modelo simplificado de um AVR.

Nas funções (5.1) e (5.2) estão as equações do gerador e a excitatriz.

$$G_1 = \frac{K_a}{(T_a \cdot s + 1)} \quad (5.1)$$

$$G_2 = \frac{K_g}{(T_g \cdot s + 1)} \quad (5.2)$$

No cálculo dos parâmetros dos compensadores, pólos dominantes podem ser especificados (5.5). Para um determinado fator de amortecimento ( $\zeta$ ) especificado está associado um máximo pico ( $M_p$ ) associado (5.3) a resposta da malha de controle para uma entrada em degrau. A relação entre o tempo de acomodação ( $T_a$ ) da malha de controle e a especificação da frequência natural de oscilação ( $\omega_n$ ) é dada por (5.4).

$$M_p = e^{-\frac{\pi \cdot \zeta}{\sqrt{1-\zeta^2}}} \quad (5.3)$$

$$T_a = \frac{4}{\zeta \cdot \omega_n} \quad (5.4)$$

$$S_1 = -\xi \cdot \omega_n \pm j \cdot \omega_n \cdot \sqrt{1 - \xi^2} \quad (5.5)$$

A função básica de um compensador *Lead/Lag* convencional é dada por (5.6). O valor de  $K_c$  pode ser calculado por meio de (5.7) e as constantes de tempo  $T_z$  e  $T_p$  pelas equações (5.8) e (5.9), onde o termo  $G(s)H(s)$  nas expressões (5.7), (5.8) e (5.9) é equivalente a  $G_1(s)G_2(s)$ . Os parâmetros  $T_z$  e  $T_p$  resultantes equivalem aos parâmetros  $T_b$  e  $T_c$  do compensador *AVR* clássico representado na Figura 5.4. Os fatores de escala  $G_e$  e  $G_v$  são os parâmetros de sintonia do controlador *Fuzzy* e foram ajustados com os valores 2 e 0,1 respectivamente.

$$C(s) = K_c \frac{T_z s + 1}{T_p s + 1} \quad (5.6)$$

$$K_c \lim_{s \rightarrow 0} s^1 G(s)H(s) = \frac{1}{e_{ss}} \quad (5.7)$$

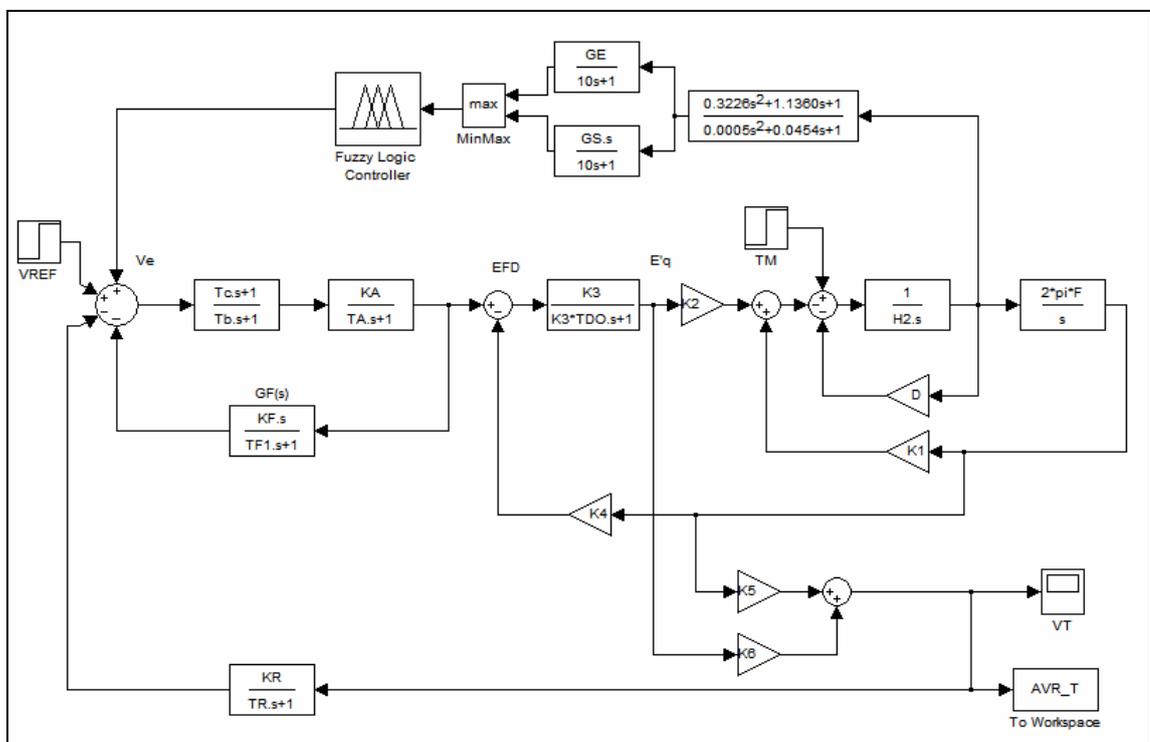
$$T_z = \frac{\sin \angle S_1 - K_c |G(S_1)H(S_1)| \operatorname{sen}(\angle G(S_1)H(S_1) - \angle S_1)}{|G(S_1)H(S_1)| K_c |S_1| \operatorname{sen} \angle G(S_1)H(S_1)} \quad (5.8)$$

$$T_p = \frac{K_c |G(S_1)H(S_1)| + \operatorname{sen} \angle S_1 \cdot \operatorname{sen}(\angle G(S_1)H(S_1) + \angle S_1)}{|S_1| \operatorname{sen} \angle G(S_1)H(S_1)} \quad (5.9)$$

Para esse trabalho foi convenicionado que o erro em regime estacionário ( $e_{ss}$ ) será de 0,0125 para entrada degrau. Usando os dados da Tabela 5.1 e as fórmulas acima, foram determinados os valores  $K_c=1$ ,  $T_z = 1$  e  $T_p = 12$ . Textos como Phillips e Harbor (1996) mostram a formulação e utilização destas equações no projeto de compensadores *Lead/Lag* em geral.

Com os parâmetros de compensação calculados, pode-se montar o modelo de simulação como indicado na Figura 5.5. Para a simulação foi utilizado o modelo completo de uma máquina síncrona.  $V_{REF}$  representa a tensão desejada (em

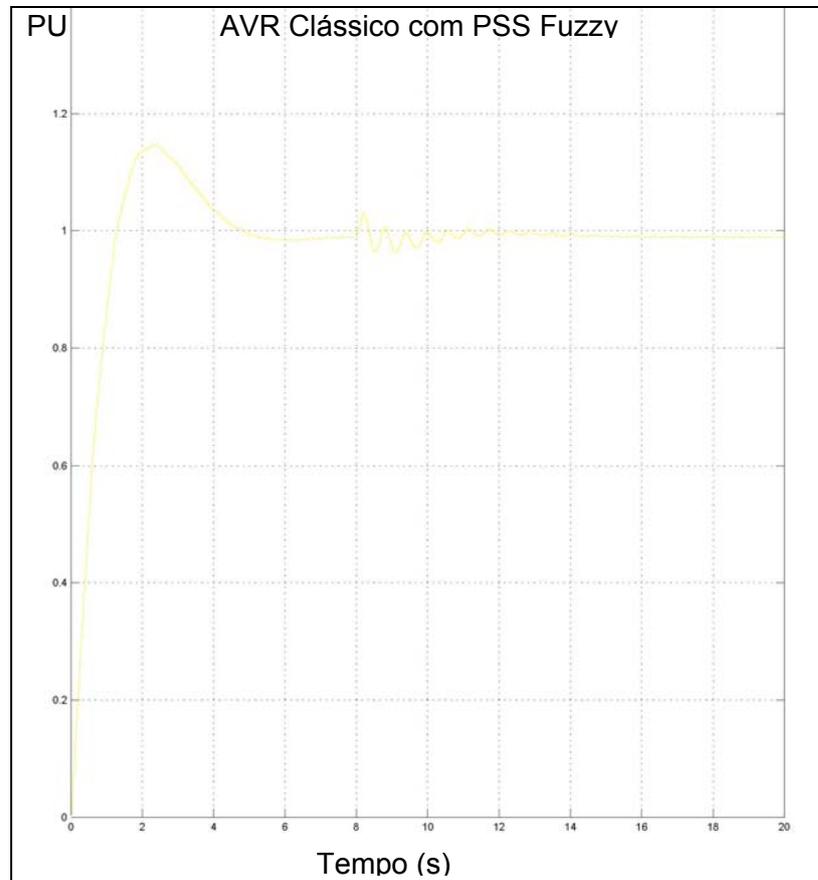
p.u.) e  $V_T$  é o valor nos terminais da máquina síncrona em p.u. A entrada de distúrbio é representada por TM que insere um sinal na planta simulando uma variação de carga. No modelo apresentado a velocidade da máquina é utilizada para realimentar o sinal estabilizante do *PSS Fuzzy*, cujo sinal é somado ao valor de referência de tensão desejada para a malha de controle de tensão do ARV clássico. A função de transferência em série com o estabilizado nebuloso foi adicionada para complementar o processamento do sinal que informa o desvio da frequência do sistema devido variações de carga do processo.



**Fig.5.5:** Malha de controle com ARV clássico e *PSS Fuzzy*.

A Figura 5.6 ilustra a resposta obtida para uma entrada degrau unitário. Em seis segundos ocorre uma perturbação de carga no sistema. Analisando o gráfico verifica-se que o máximo pico foi de 16% e que o tempo de acomodação ficou próximo de 4 [s]. Esses resultados são reproduções dos valores obtidos por Féo

(2004), onde foram efetuadas comparações com malha de controle que empregava um *PSS* convencional.

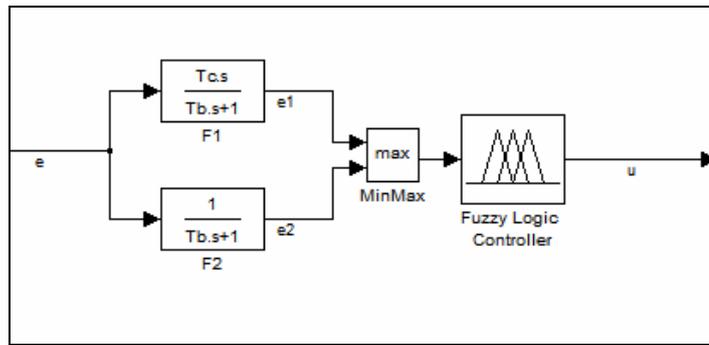


**Fig.5.6:** Resposta da malha de controle com AVR clássico e *PSS Fuzzy*.

#### 5.4. Modelagem do AVR Fuzzy com *PSS Fuzzy*

Agora será mostrada a implementação da malha de controle com um *AVR Fuzzy* em conjunto com o *PSS Fuzzy* citado anteriormente. A Figura 5.7 ilustra a estrutura do compensador do *AVR Fuzzy*. F1 é um filtro passa-altas e F2 um filtro passa-baixas (sendo similares aos filtros F e F1 citados anteriormente). As ações destes filtros promovem efeitos de avanço ou atraso de fase na compensação do controlador.

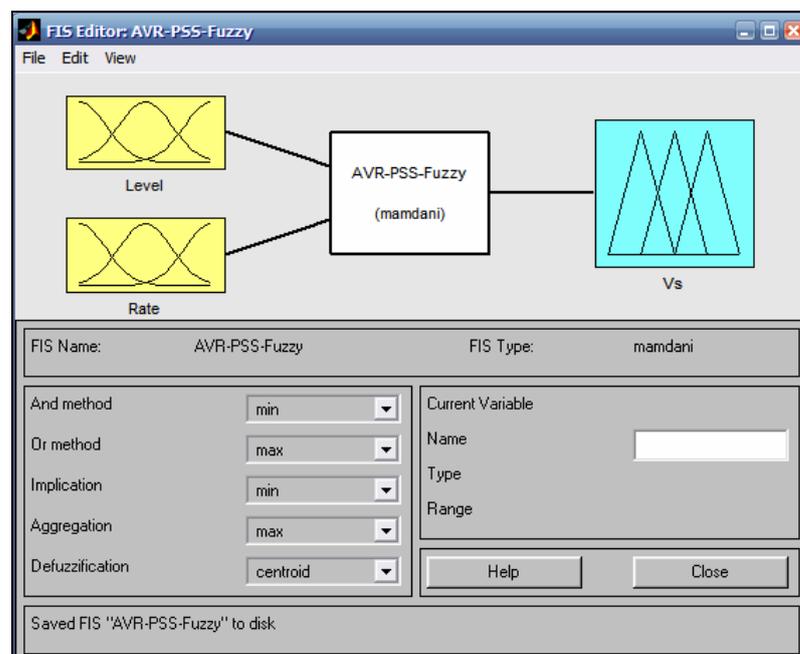
---



**Fig.5.7:** Controlador *Fuzzy* Tipo Avanço/ Atraso de Fase.

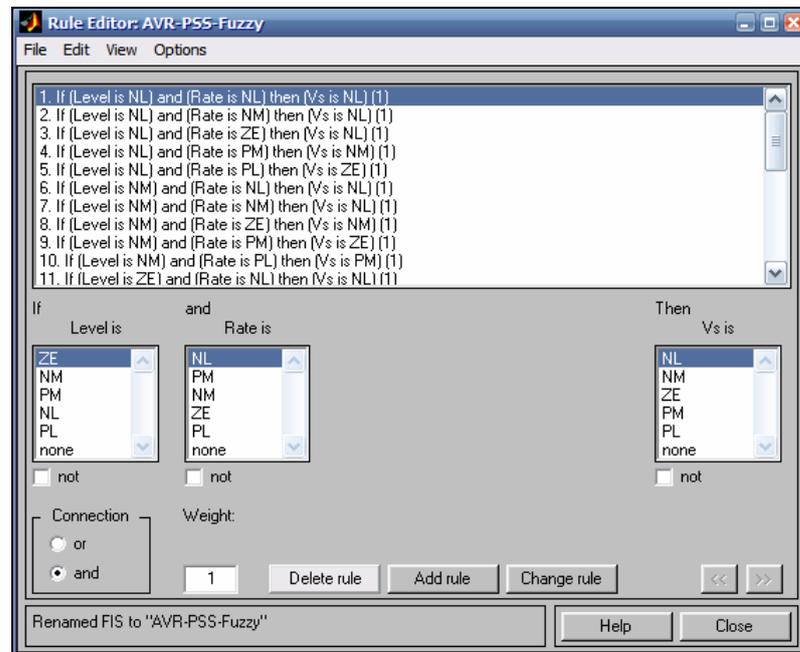
Os itens seguintes ilustram como utilizar o ambiente *Simulink* do *MatLab* para realizar a simulação proposta. O bloco *Fuzzy Logic Controller* pode ser programado conforme mostrado na Figura 5.8, onde os erros “e1” e “e2” são chamados de “*Level*” e “*Rate*” no editor do simulador, assim como o valor “u” é chamado de *Vs*. Nesse trabalho foram adotados, tanto para a etiqueta denominada “*level*” quanto para a designada como “*rate*”, cinco conjuntos nebulosos na construção das regras. Foram denominados os seguintes termos lingüísticos:

- negativo largo;
- negativo médio;
- zero;
- positivo médio;
- positivo largo.



**Fig.5.8:** Programação do *Fuzzy Logic Controller*.

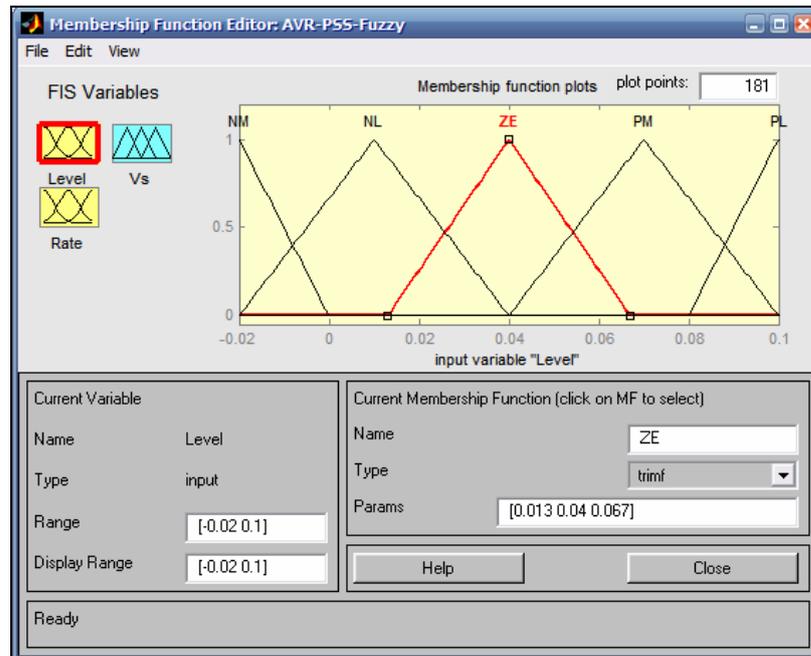
Para criar as regras no programa de simulação é necessário acessar o campo “Mamdani” que é mostrado na figura 5.8. Ao acessar esse campo o programa abre uma nova janela semelhante a apresentada na figura 5.9, onde é possível digitar as regras adotadas para o controlador nebuloso em questão. O conjunto de regras nebulosas adotadas pode ser visto na tabela 5.2. Os universos de discurso das funções de pertinências foram estimados por simulação com valores extremos das saídas dos filtros dos compensadores AVR's convencionais. Os valores encontrados estavam na faixa de 0,1 a -0,02 e de 0,09 a -0,01. As figuras 5.10 e 5.11 mostram as janelas de edição das funções de pertinência.



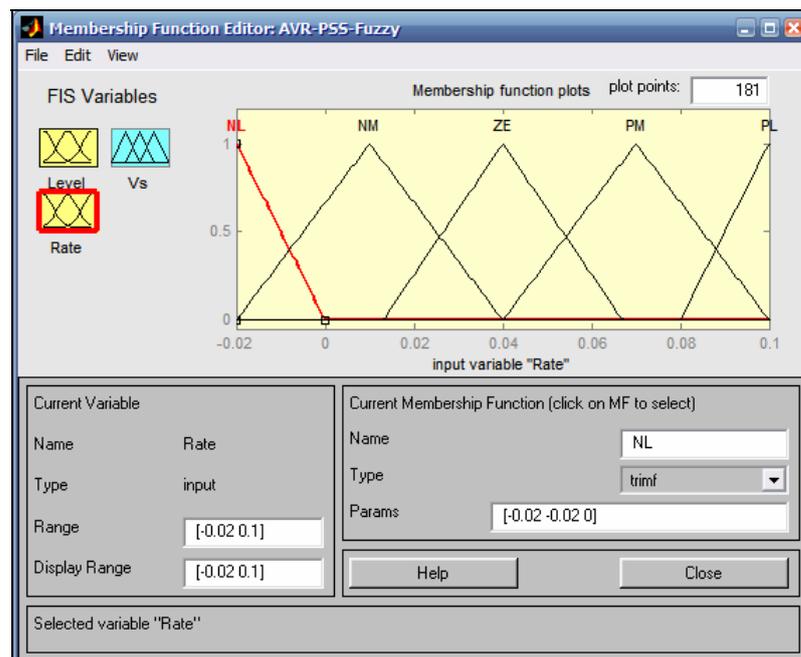
**Fig.5.9:** Introdução das Regras no *Fuzzy Logic Controller*.

**Tab.5.2:** Conjunto de Regras Nebulosas.

Índice	Regras Difusas
1	If (level is NL) and (rate is NL) then (Vs is NL)
2	If (level is NL) and (rate is NM) then (Vs is NL)
3	If (level is NL) and (rate is ZE) then (Vs is NL)
4	If (level is NL) and (rate is PM) then (Vs is NM)
5	If (level is NL) and (rate is PL) then (Vs is ZE)
6	If (level is NM) and (rate is NL) then (Vs is NL)
7	If (level is NM) and (rate is NM) then (Vs is NL)
8	If (level is NM) and (rate is ZE) then (Vs is NM)
9	If (level is NM) and (rate is PM) then (Vs is ZE)
10	If (level is NM) and (rate is PL) then (Vs is PM)
11	If (level is ZE) and (rate is NL) then (Vs is NL)
12	If (level is ZE) and (rate is NM) then (Vs is NM)
13	If (level is ZE) and (rate is ZE) then (Vs is ZE)
14	If (level is ZE) and (rate is PM) then (Vs is PM)
15	If (level is ZE) and (rate is PL) then (Vs is PL)
16	If (level is PM) and (rate is NL) then (Vs is NM)
17	If (level is PM) and (rate is NM) then (Vs is ZE)
18	If (level is PM) and (rate is ZE) then (Vs is PM)
19	If (level is PM) and (rate is PM) then (Vs is PL)
20	If (level is PM) and (rate is PL) then (Vs is PL)
21	If (level is PL) and (rate is NL) then (Vs is ZE)
22	If (level is PL) and (rate is NM) then (Vs is PM)
23	If (level is PL) and (rate is ZE) then (Vs is PL)
24	If (level is PL) and (rate is PM) then (Vs is PL)
25	If (level is PL) and (rate is PL) then (Vs is PL)



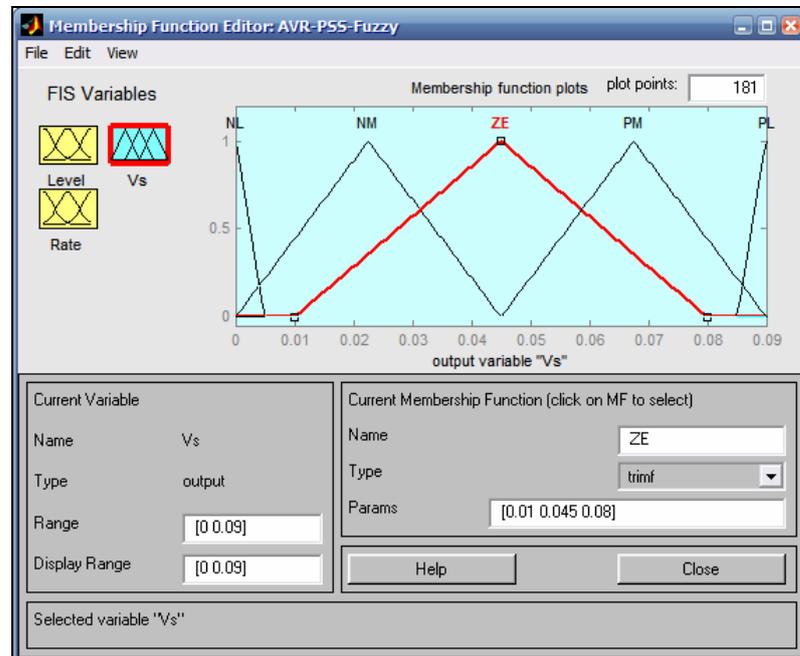
**Fig.5.10:** Funções de Pertinência da Entrada “Level”.



**Fig.5.11:** Funções de Pertinência da Entrada “Rate”.

As funções de pertinência da saída do compensador *AVR Fuzzy* foram ajustadas de forma que a resposta da malha de controle a entrada degrau tivesse uma resposta satisfatória. Nesse caso o conjunto zero (ZE) teve sua base ampliada, e os

conjuntos negativo largo (NL) e positivo largo (PL) tiveram suas bases reduzidas. A Figura 5.12 mostra a função de pertinência adotada para a saída  $V_s$ .



**Fig.5.12:** Funções de Pertinência Saída “ $V_s$ ”.

Para realizar a simulação do *AVR Fuzzy* com *PSS Fuzzy*, o bloco de controle convencional foi retirado e em seu lugar foi acrescentado o compensador nebuloso ilustrado na Figura 5.13. A Figura 5.14 ilustra a simulação do sistema. É fácil notar que o desempenho melhorou em relação a resposta da Figura 5.5 (com *AVR* convencional e *PSS Fuzzy*). Observando a nova resposta, percebe-se que o máximo pico está torno de 12% (menor que o do compensador tradicional em 17%). O tempo de acomodação também foi ligeiramente menor por volta de cinco segundos, ao invés de 4 [s].

Conclui-se que a proposta de um *AVR Fuzzy* junto com um *PSS Fuzzy* apresentou um desempenho melhor que um *ARV* convencional e um *PSS Fuzzy*



## **CAPÍTULO 6 - IMPLEMENTAÇÃO PRÁTICA DO AVR FUZZY.**

### **6.1. Introdução**

Como os resultados simulados apresentaram adequados, foi desenvolvida uma bancada de ensaios para verificar resultados práticos com a estrutura proposta.

### **6.2. Bancadas de Ensaio**

Para implementar a malha de controle com o *AVR Fuzzy* e o *PSS Fuzzy*, foi montada a estrutura ilustrada na foto da figura 6.1. Foram utilizadas quatro máquinas elétricas, sendo duas de corrente contínua e dois geradores síncronos.

Foram usados reostatos de potência e lâmpadas elétricas para simular variações de carga, pontes tiristorizadas e módulos de gatilhamento como sistema de excitação. As cargas foram comutadas através de um comutador manual.

Os instrumentos de medição empregados foram voltímetros, amperímetros, freqüencímetros e um osciloscópio digital.

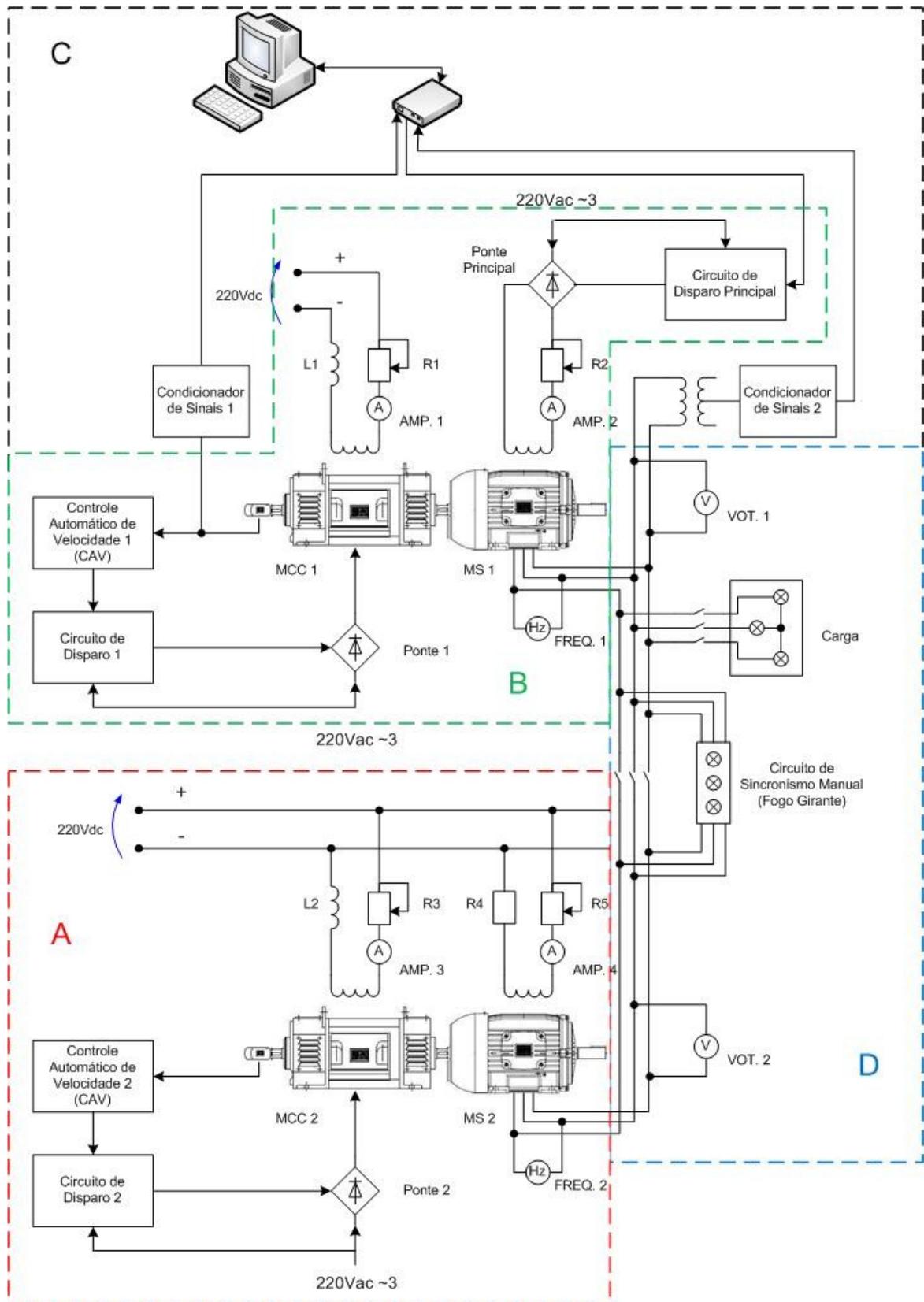
---



**Fig.6.1:** Bancada de Ensaio.

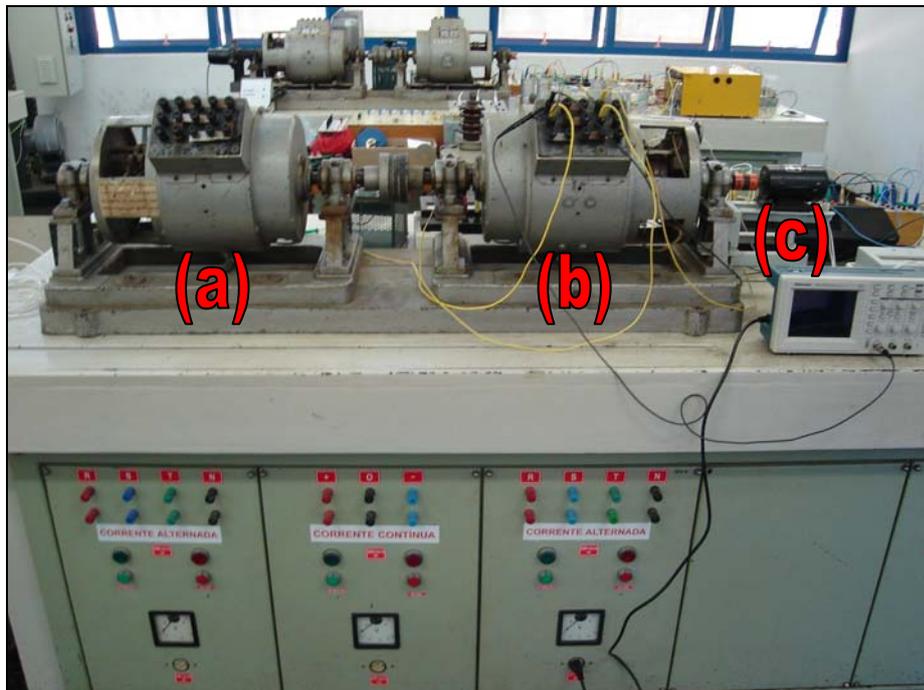
O diagrama funcional da montagem da bancada de testes está representado na Figura 6.2. O mesmo pode ser dividido em quatro partes:

- Conjunto de geração secundário: Um gerador foi utilizado para executar o paralelismo com o conjunto de geração principal. O objetivo é mostrar o efeito da atuação do *PSS Fuzzy*. No desenho o esse conjunto está no campo “A” da figura.
  - O campo “B” representa o conjunto de geração principal, onde os testes do compensador *AVR Fuzzy* foram realizados.
  - No campo “C” está o sistema de aquisição de dados, os condicionadores de sinais e o computador que implementa os algoritmos de controle.
  - As cargas e sistema de paralelismo formam o campo “D”.
-



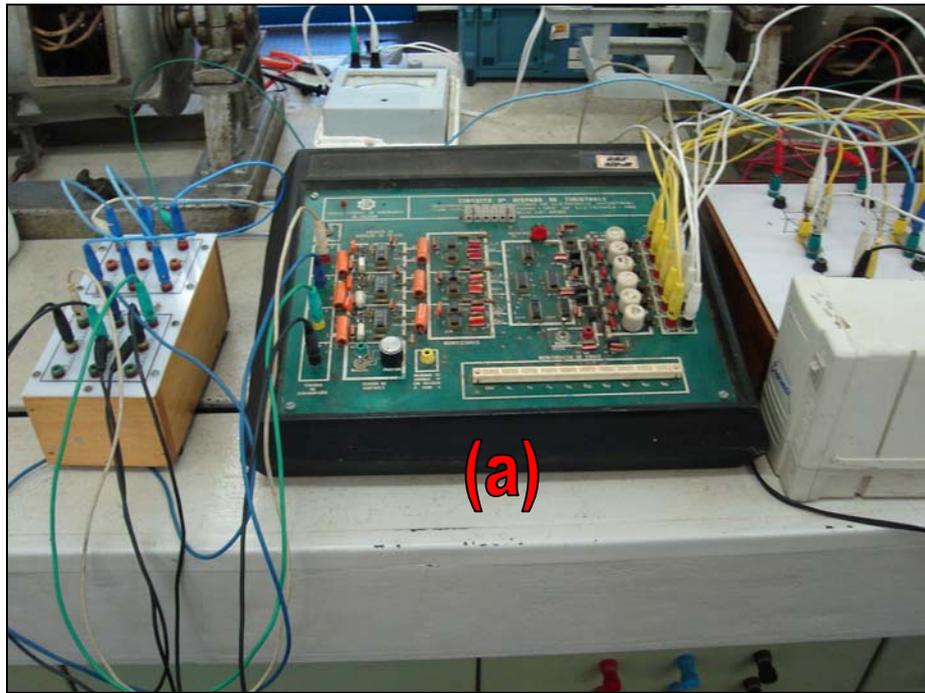
**Fig.6.2:** Desenho Esquemático de Montagem.

Os sistemas de geração são constituídos de dois geradores, sendo que as máquinas de corrente contínua representam as máquinas primárias (turbinas) conectadas nos eixos de cada um dos geradores síncronos, como mostrado na foto da Figura 6.3. Os acionamentos das máquinas de corrente contínua são semelhantes para ambos os conjuntos de geração. O motor de corrente contínua tem sua bobina de campo alimentada por uma fonte de 220Vdc. Na figura é possível observar o tacômetro (“c”), que realimenta a informação da velocidade para a regulação de rotação da máquina primária de um dos arranjos.



**Fig.6.3:** Foto Ilustrativa do Sistema de Ensaio.

A excitação das máquinas de corrente é realizada por intermédio de pontes tiristorizadas alimentada pela tensão da rede elétrica da bancada (230Vca / 60Hz). O controle do ângulo de disparo é realizado por circuitos de disparo, como exemplificado na foto da Figura 6.4.

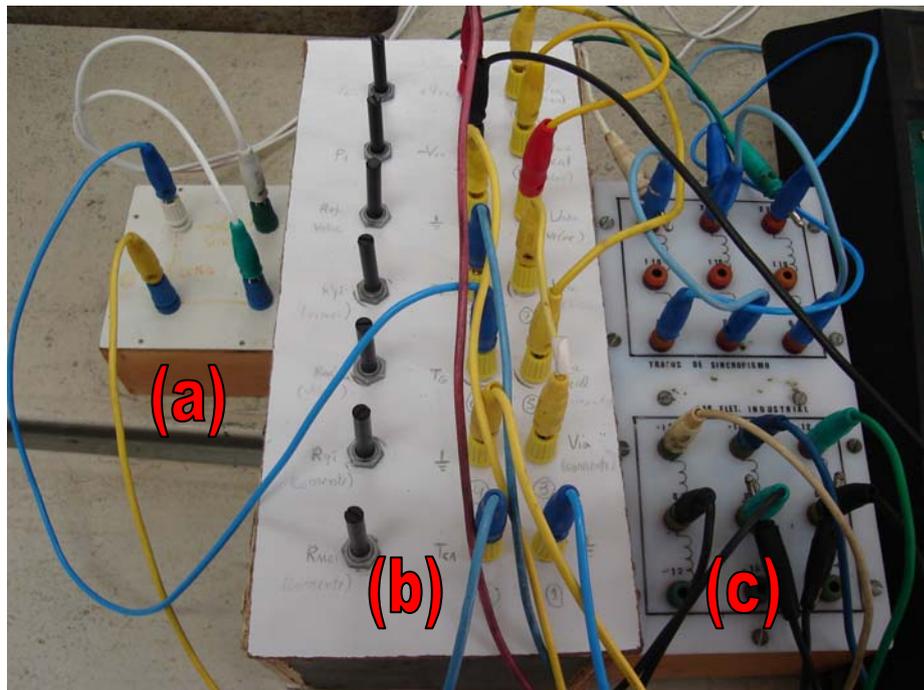


**Fig.6.4:** Foto (“a”) de Circuito de Disparo de Bancada.

O circuito de disparo possui uma entrada disponível para comando em sinal de tensão externa. Um circuito deste tipo foi empregado no acionamento do controlador automático de velocidade, que tem por finalidade manter a rotação das máquinas constante, independente das variações que ocorrerem na carga.

A figura 6.5 ilustra o controlador automático de velocidade (“b”) e seu transformador de acoplamento (“c”), que foi necessário para ajustar as tensões de saída desse módulo com os valores permitidos na entrada do circuito de disparo. Esse controlador estabiliza a velocidade de rotação das máquinas em uma frequência de 60Hz. Para isso ele utiliza a realimentação da velocidade das máquinas, medida através do tacômetro (Figura 6.2 “c”) acoplado ao eixo do motor de corrente contínua (Figura 6.3 “b”).

A saída de tensão do tacômetro usado no conjunto de geração principal (campo “B” da Figura ) está em uma faixa de 0 a 40 V, sendo que uma velocidade de 60 Hz representa aproximadamente 30V. Como o circuito de controle trabalha com uma alimentação máxima de 10V, foi necessário acrescentar um divisor resistivo como indicado na foto da Figura 6.5 (“a”).



**Fig.6.5:** Sistema para Controle de Velocidade.

O circuito de controle de velocidade é constituído basicamente por amplificadores operacionais formando um compensador PID. O ajuste dos parâmetros do compensador é realizado por intermédio de potenciômetros da tampa superior do módulo, como é possível observar na foto da figura 6.5 “b”.

O esquema elétrico, o desenho do circuito impresso e um breve descritivo de funcionamento do controlador automático de velocidade podem ser encontrados no Anexo 1 desse trabalho.

Para as máquinas síncronas existem algumas diferenças fundamentais na montagem dos dois conjuntos de geração, devido ao fato que todos os testes do sistema de controle difuso ser realizado no conjunto principal da bancada.

No o conjunto de geração secundário (campo “A” da Figura 6.2) a máquina síncrona foi excitada com uma corrente fixa no seu campo. Foi utilizada uma fonte de corrente contínua de 220Vdc e um conjunto de resistores e reostatos de potência para garantir que a corrente de campo não ultrapassasse o valor máximo permitido para a máquina. O conjunto foi ajustado para que a tensão na saída dos terminais da armadura da máquina fosse de 220Vac com uma freqüência de 60Hz.

Para o conjunto de geração principal, onde são implementados os algoritmos de controle, a excitação do campo da máquina síncrona é similar ao sistema de controle de velocidade das máquinas de corrente contínua, ou seja, existe um circuito de disparo que controla o ângulo de condução da ponte conversora que atua na excitação do gerador em questão.

Depois de testar individualmente cada um dos conjuntos, foi necessário preparar a estrutura para fazer o paralelismo entre os geradores da bancada de ensaio. Nesse trabalho foi definido que seria adotado o método do fogo girante para realizar o sincronismo das máquinas síncronas.

No Anexo 2 está descrita com maiores detalhes a técnica do fogo girante e como é feita a ligação elétrica entre os geradores.

O paralelismo entre as máquinas é extremamente importante para esse projeto, pois tem como objetivo verificar o funcionamento do *PSS Fuzzy*, que deve reduzir a amplitude das oscilações de baixa frequência que podem ocorrer em um gerador no momento que ele é interligado a um outro sistema elétrico de geração. Também será verificado o tempo de acomodação do sistema após efetuar o paralelismo entre os geradores para comprovar a robustez do *PSS* nebuloso.

O circuito de sincronismo é composto por uma seccionadora tipo faca tripolar, e um sistema de lâmpadas ligadas conforme mostrado no Anexo 2 que são responsáveis pela indicação do momento adequado para realizar o paralelismo entre os geradores da bancada. Para garantir o paralelismo também foi utilizado um fasímetro para determinar a seqüência correta das fases. Os níveis das tensões geradas foram monitorados por dois voltímetros.

Como cargas elétricas foram utilizadas lâmpadas de filamento com potências variando entre 200 e 100W ligadas em delta. Essas cargas foram comutadas através de uma chave seccionadora trifásica. A quantidade de cargas ligadas ao sistema varia de acordo com a simulação executada, e será detalhada na parte dos ensaios realizados.

No computador da bancada foram implementados os algoritmos de controle e registradas as grandezas analógicas e / ou digitais para controle em tempo real do processo. No computador foi instalada a placa de aquisição de dados interface de informações do sistema de controle. A placa possui conversores analógicos-digitais (A/D) e digitais-analógicos (D/A) responsáveis pela coleta de informações

---

da tensão controlada, da velocidade do processo e pelo comando de disparo da ponte tiristorizada do sistema de excitação do conjunto de geração principal.

A placa de aquisição utilizada é o modelo PCL-711 da Advantech que dispõe de oito entradas analógicas e uma saída analógica. Os conversores A/D e D/A têm faixas de 0 a  $\pm 5$  e 0 a 10 Vdc respectivamente, resolução de 12 bits e taxa de amostragem de até 10 KHz. As características da placa estão descritas no Anexo 3 desse trabalho.

O sistema de controle conta ainda com dois circuitos condicionadores, um ligado na saída do tacômetro e outro na saída do TP que está conectado aos terminais da armadura da máquina síncrona do sistema de geração principal. Esse circuito é formado por um filtro composto por um resistor de  $10K\Omega$  e um capacitor de  $0.22\mu F$  e é utilizado para filtrar eventuais ruídos de alta frequência presentes na instrumentação.

Os programas de controle em tempo real foram implementados em Linguagem C, e os dados coletados gravados em arquivos foram acessados através de programas em MATLAB visando a elaboração dos gráficos dos ensaios realizados.

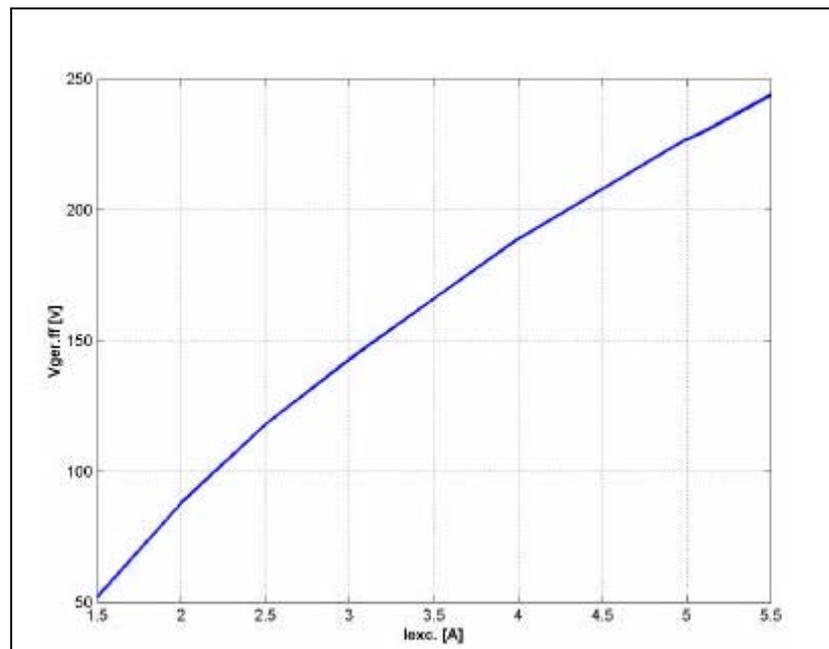
### **6.3. Modelagem da Bancada de Ensaio**

Como mostrado em Féo (2004) foi realizada a modelagem da bancada de ensaio. Foram realizadas algumas medidas na tensão gerada no sistema ao variar-se a corrente de excitação do processo (Tabela 6.1), mantendo a velocidade síncrona constante. O gráfico da Figura 6.6 mostra a curva de magnetização do gerador indicando uma não linearidade inerente ao sistema.

---

**Tab.6.1:** Dados da Curva de Magnetização do Sistema.

<b>I<sub>exc.</sub> [A]</b>	<b>V<sub>ger. ff</sub> [v]</b>
1.5	52.0
2.0	88.0
2.5	118.0
3.0	143.0
3.5	166.0
4.0	189.0
4.5	208.0
5.0	227.0
5.1	230.0
5.5	244.0

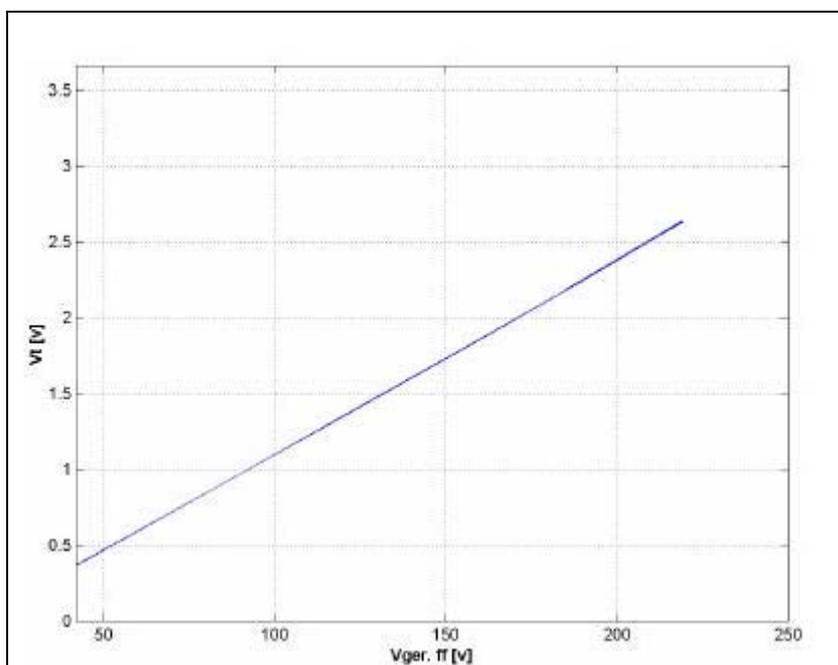
**Fig.6.6:** Curva de Magnetização do Sistema.

Foi obtida a resposta (Tabela 6.2 e Figura 6.7) do transdutor de tensão e seu condicionador de sinais para verificar a linearidade do mesmo.

---

**Tab.6.2:** Dados da Curva de Magnetização do Gerador.

Vger. ff [V]	Vt [V]
42	0.366
90	0.97
115	1.289
156	1.807
185	2.18
211	2.525
219	2.634

**Fig.6.7:** Gráfico do Transdutor de Tensão.

As outras instrumentações empregadas, como transdutores de velocidade, apresentam respostas lineares para a faixa de operação utilizada. Sendo assim, não foi necessário nenhum tipo de linearização nos sinais dos transdutores.

---

#### 6.4. Linearização do Conjunto Circuito de Disparo e Ponte Tiristorizada

Deseja-se uma relação linear entre o sinal de controle ( $u$ ) proveniente do canal D/A que comanda o circuito de disparo, e a tensão retificada de excitação ( $V_{exc}$ ) proveniente da ponte conversora que alimenta a excitação do gerador, ou seja:

$$V_{exc} = 1,35 * V_{ff} * \left( \frac{u}{5} \right) \quad (6.1)$$

Para cada intensidade de “ $u$ ” tem-se uma tensão  $V_{exc}$  associada:

$$\begin{array}{l} u = 0.0 \rightarrow V_{exc} = 0 \\ u = 2.5 \rightarrow V_{exc} = Max / 2 \\ u = 5.0 \rightarrow V_{exc} = Max \end{array} \quad (6.2)$$

Como é conhecido, o retificador pode ser expresso pela equação 6.3.

$$V_{exc} = 1,35 * V_{ff} * \cos \alpha \quad (6.3)$$

O circuito de disparo pode ser modelado pela equação 6.4, onde  $V_c$  é o sinal de comando e “ $\alpha$ ” é o ângulo de disparo da ponte retificadora.

$$\alpha = 180^\circ \cdot \left( 1 - \frac{V_c}{10} \right) \quad (6.4)$$

Substituindo as equações (6.1) em (6.3) temos:

---

$$\frac{u}{5} = \cos \alpha \quad (6.5)$$

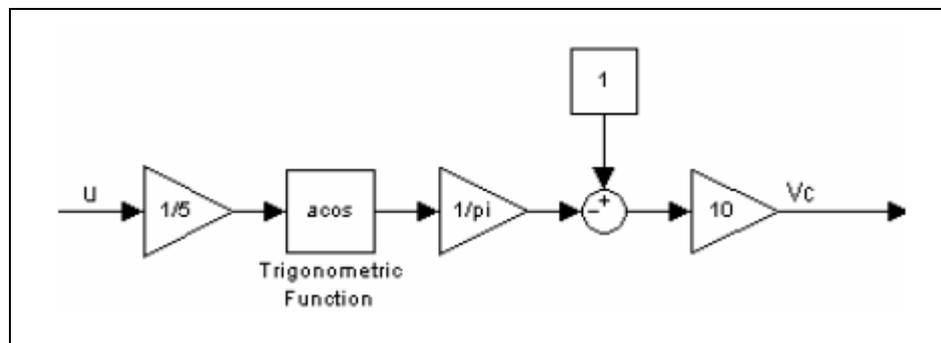
Com as equações (6.4) e (6.5) temos:

$$\frac{u}{5} = \cos \left( 180^\circ \cdot \left( 1 - \frac{V_c}{10} \right) \right) \quad (6.6)$$

Trabalhando a equação anterior, pode-se verificar que a tensão de controle é dada por:

$$V_c = 10 \cdot \left( 1 - \left( a \cos \left( \frac{u}{5} \right) \right) / 180^\circ \right) \quad (6.7)$$

Assim, a relação resultante entre as variáveis “u” e “V<sub>exc</sub>” pode ser linear. Essa relação pode ser representada pelo diagrama de blocos da Figura 6.8.

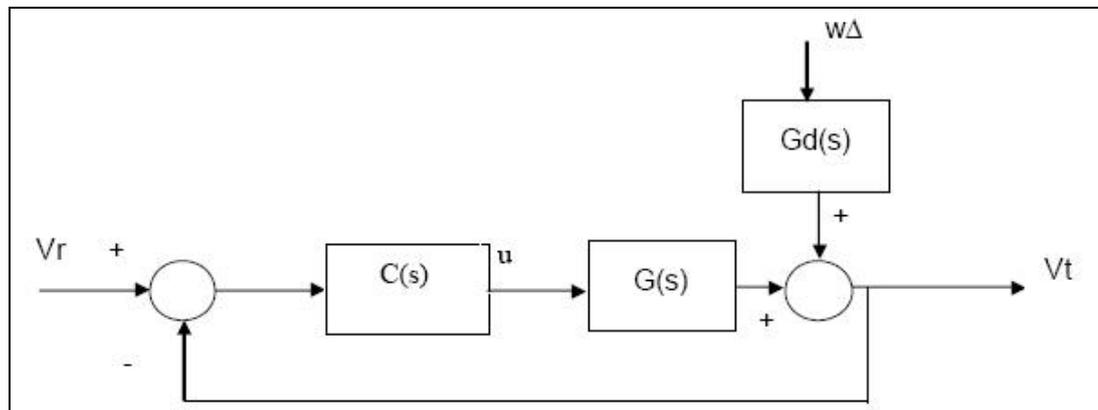


**Fig.6.8:** Processamento de Linerarização do Sinal de Comando.

## 6.5. Identificação do Modelo do Sistema

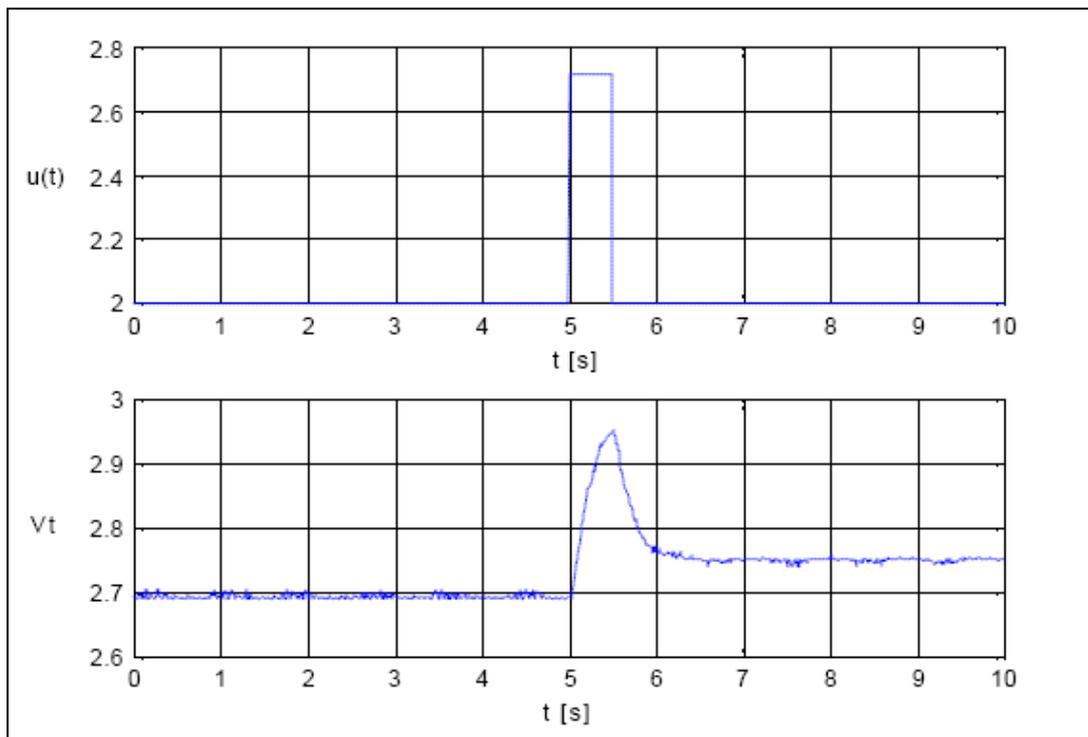
A finalidade de identificação do modelo do sistema consiste em determinar a função de transferência do processo, para em seguida calcular os parâmetros do compensador *Lead/Lag* tradicional e depois elaborar a compensação *Fuzzy*.

A estrutura simplificada do sistema de regulação de tensão do gerador é mostrada na Figura 6.9. A função  $G(s)$  representa o sistema de excitação, gerador e cargas. A função  $G_d(s)$  representa perturbações devido a variações de carga, que ocasionam modificações de frequência no gerador. O bloco  $C(s)$  simboliza o compensador da malha de controle de tensão do gerador (o *AVR*).



**Fig.6.9:** Modelo do Sistema de Regulação de Tensão.

Para identificar as funções  $G(s)$  e  $G_d(s)$  que modelam o sistema, foram empregadas técnicas de identificação usando um *toolbox* do *MATLAB*. Inicialmente foi obtida a função  $G(s)$  do sistema. Mantendo a rotação, tensão e carga do gerador em valores nominais, altera-se o sinal de comando ( $u$ ) de excitação do sistema em torno do seu ponto de operação e registra-se (Figura 6.10) os valores resultantes da tensão terminal ( $V_t$ ) do gerador, em um tempo de amostragem definido.



**Fig.6.10:** Medidas para identificação de  $G(s)$ .

O *Matlab* utiliza o comando “*arx*” para determinar a função de transferência do sistema usando o método dos mínimos quadrados. A seguinte seqüência de comandos foi empregada:

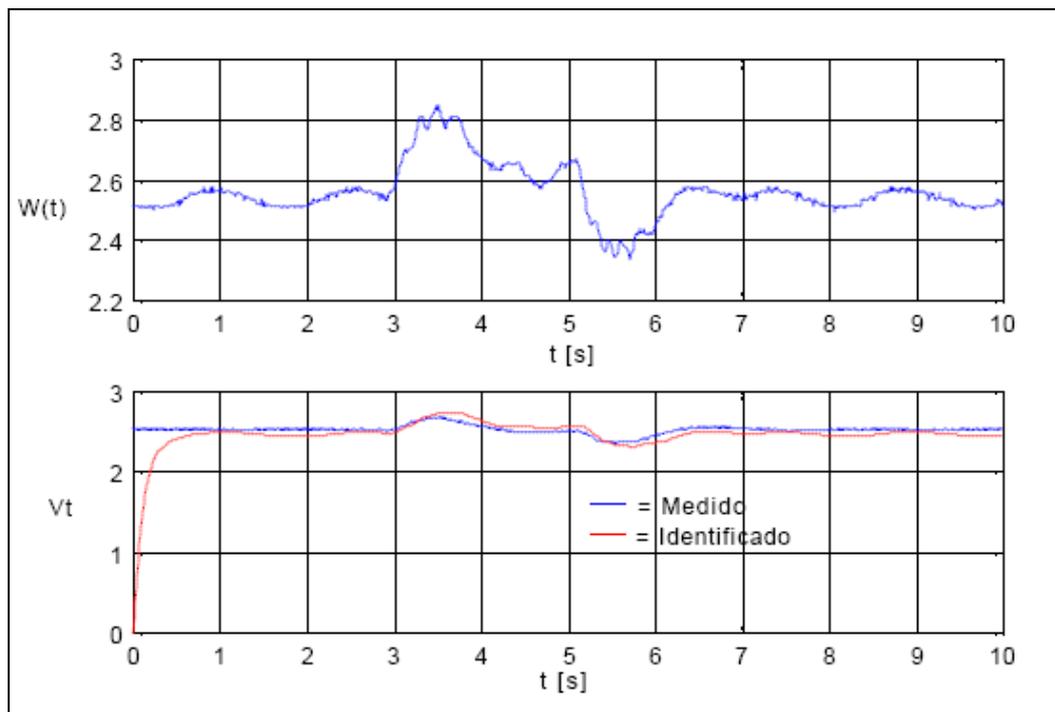
```
v [Vt(500:900) u(500:900)]; <Enter>
modelo arx(v[ 1]); <Enter>
[Nz,Dz] th2tf (modelo) <Enter>
[Ns,Ds] d2cm(Nz,Dz,0.001,'zoh') <Enter>.
```

Como o sinal da entrada de comando foi ajustado para excitar o gerador em torno do seu valor nominal de operação, o modelo linearizado obtido é representado por uma função de transferência de segunda ordem dada pela expressão (6.8), incorporando a função do excitador e do gerador.

$$G(s) = \frac{2,2526}{(0,0006s^2 + 1,7985s + 0,8724)} \quad (6.8)$$

Tanto a não linearidade da curva de magnetização do gerador como os limites de operação da ponte retificadora que alimenta o campo de excitação, são representados por uma função de saturação com limites em (+/- 5).

Da mesma forma é obtida a função  $G_d(s)$ . Mantendo o sinal de comando ( $u$ ) de excitação do sistema em torno do seu valor nominal, provocou-se uma variação de carga no sistema registrando os valores resultantes da tensão terminal ( $V_t$ ) do gerador e da informação da variação de rotação ( $W\Delta$ ) correspondente. A Figura 6.11 mostra o resultado das medidas para identificação da função  $G_d(s)$ .



**Fig.6.11:** Medidas para Identificação de  $G_d(s)$ .

Com os dados obtidos calcula-se a função de transferência correspondente, expressa pela equação (6.9). A curva em vermelho mostra a resposta simulada da função identificada, e em cor azul estão os dados medidos anteriormente.

$$G_d(s) = \frac{9,81}{(s + 8,76)} \quad (6.9)$$

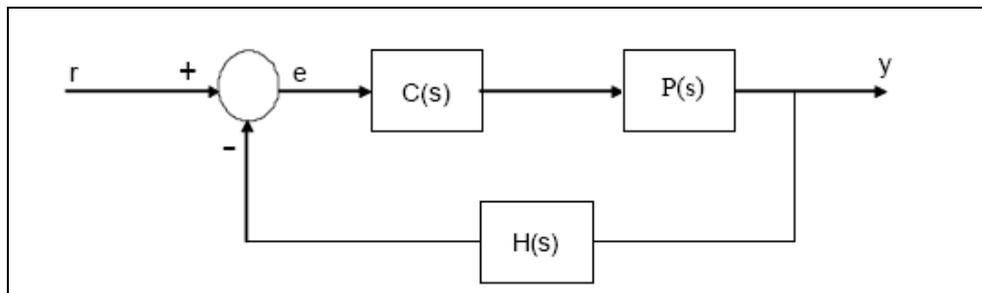
## 6.6. Projeto de um Compensador tipo Lead/Lag

Para comprovar que um compensador nebuloso pode apresentar resultados melhores no controle de sistemas não-lineares que um compensador clássico, serão projetados controladores *Lead/Lag* clássicos e difusos.

Nesse item serão mostrados os cálculos dos parâmetros para o compensador de avanço e atraso de fase convencional. As especificações desejadas são:

- Margem de fase (MF) de aproximadamente  $84^\circ$ ;
- Freqüência de cruzamento de ganho ( $\omega_{cg}$ ) em torno de 53 [rad/s];
- Erro em regime permanente de 0,005 (0,5%) para entrada degrau.

Como forma de simplificação será adotada a estrutura da Figura 6.12, onde  $C(s)$  é a função do compensador,  $P(s)$  é a função básica  $G(s)$  do processo, desconsiderando o efeito da perturbação modelada por  $G_d(s)$ , e  $H(s)$  é a função do transdutor de tensão. A variável “y” é a grandeza controlada, “r” é a entrada de referência e a variável “e” é o erro da malha de controle. A equação (6.10) modela a função do compensador da malha.



**Fig.6.12:** Diagrama de Blocos da Malha de Controle Adotada.

$$C(s) = \frac{a_1s + a_0}{b_1s + 1} \quad (6.10)$$

Para calcular os parâmetros do compensador serão usadas expressões conhecidas da teoria de controle (Phillips e Harbor, 1996). A expressão que relaciona o

erro em regime permanente é dada pela equação (6.11), onde “n” é o número de pólos na origem da função P(s) .

$$S^n * K * P(s)_{s=0} = \frac{1}{e_{ss}} \quad (6.11)$$

Fazendo s=0 na equação 6.11, temos:

$$P(s)_{s=0} = \frac{2,2526}{(0,0006 * (0)^2 + 1,7985 * (0) + 0,8724)} = \frac{2,2526}{0,8724} = 2,5821 \quad (6.12)$$

Considerando o erro em regime permanente  $e_{ss} = 0.005$  e substituindo a equação (6.12) na equação (6.11), tem-se:

$$S^0 * K * 2,5821 = \frac{1}{0,005} \rightarrow K = \frac{(1/0,005)}{2,5821} = 77,46 \quad (6.13)$$

O ângulo  $\Theta$  de compensação é dado por:

$$\theta = -180 + MF - \angle P(jWcg) * H(jWcg) \quad (6.14)$$

Onde:

$$H(s) = 1pu \rightarrow H(jWcg) = 1 \quad (6.15)$$

Substituindo  $jWcg = j53\text{rad/s}$  na equação (6.12) teremos:

$$P(j53) = \frac{2,2526}{(0,0006 * (j53)^2 + 1,7985 * (j53) + 0,8724)} = 0,0237 \angle -90,5^\circ \quad (6.16)$$

Assim, substituindo os valores calculados na equação (6.14), temos:

$$\theta = -5,5^\circ \quad (6.17)$$

As expressões dos parâmetros “a0”, “a1” e “b1” são dadas a seguir:

---

$$a_0 = K \quad (6.18)$$

$$a_1 = \frac{1 - a_0 |P(jWcg).H(Wcg)| \cdot \cos \theta}{|P(jWcg).H(Wcg)| \cdot \text{sen} \theta \cdot Wcg} = \quad (6.19)$$

$$b_1 = \frac{\cos \theta - a_0 \cdot |P(jWcg).H(Wcg)|}{\text{sen} \theta \cdot Wcg} = \quad (6.20)$$

Substituindo os valores desejados nas expressões (6.18), (6.19) e (6.20) obtêm-se os parâmetros do compensador:

$$a_0 = 77,46 \quad (6.21)$$

$$a_1 = 6,7864 \quad (6.22)$$

$$b_1 = 0,1612 \quad (6.23)$$

A função de transferência do compensador será:

$$C(s)_{\text{Lead / Lag}} = \frac{a_1 s + a_0}{b_1 s + 1} = \frac{6,7864s + 77,46}{0,1612s + 1} \quad (6.24)$$

Para verificar se as especificações do sistema de controle foram alcançadas com a função C(s) foi obtida, gera-se a resposta em frequência do sistema. A mesma é dada por:

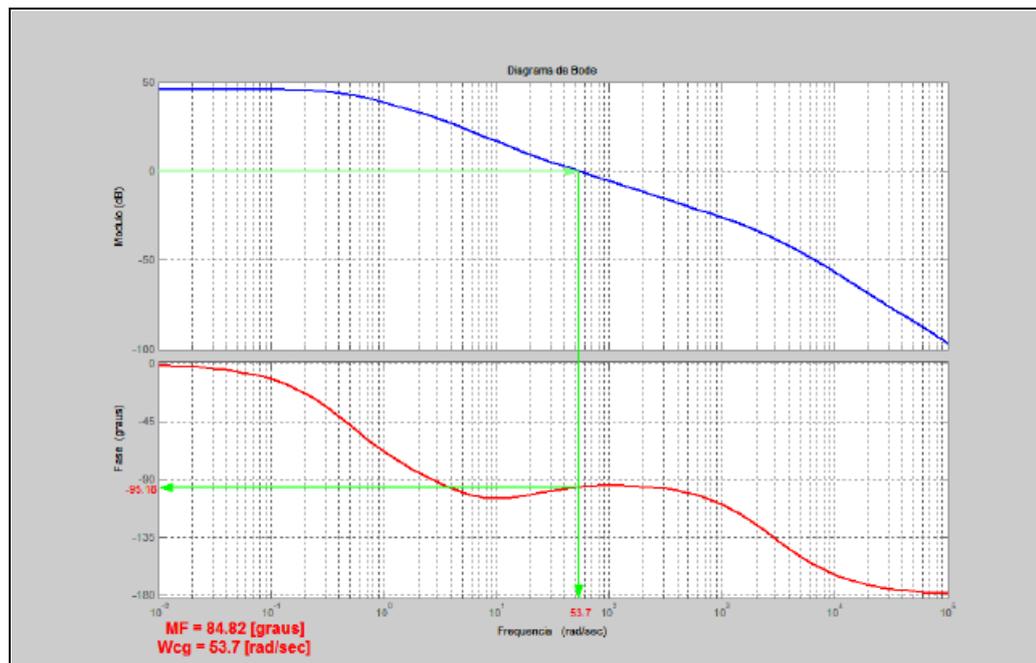
$$FT_{C_{\text{Lead / Lag}}} = \frac{6,7864s + 77,46}{0,1612s + 1} * \frac{2,2526}{(0,0006s^2 + 1,7985s + 0,8724)} = \quad (6.25)$$

$$FT_{C_{\text{Lead / Lag}}} = \frac{15,29s + 174,49}{0,0001s^3 + 0,2907s^2 + 1,9392s + 0,8724}$$

Utilizando os comandos a seguir gera-se o diagrama de Bode do sistema:

Num = [15.287 174.4864]; <Enter>  
 Den= [0.0001 0.2907 1.9392 0.8724]; <Enter>  
 Bode(num,den) <Enter>.

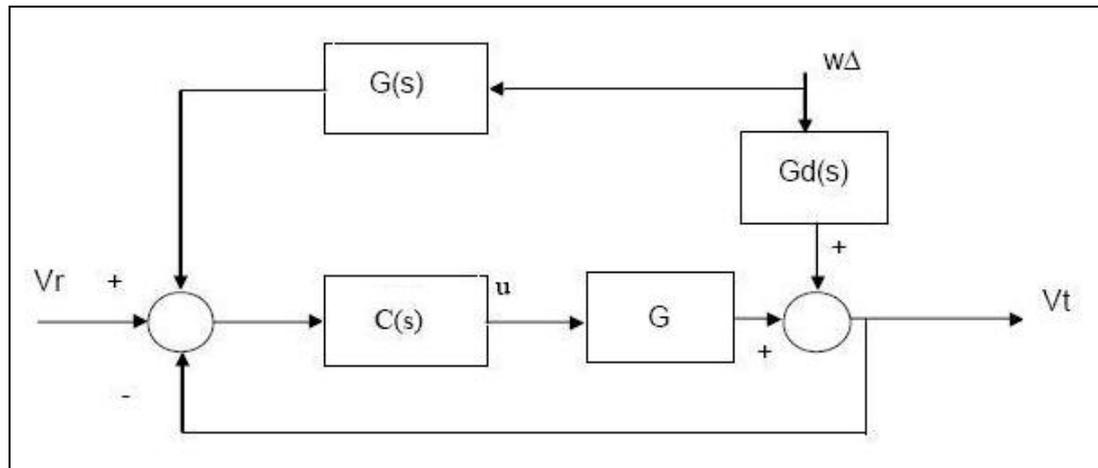
Na Figura 6.13 pode-se verificar os resultados:  $MF = 180^\circ - 95,18^\circ = 84,82^\circ$ ;  
 $W_{cg}=53,7$  rad/s.



**Fig.6.13:** Resposta em Frequência do Sistema Compensado.

## 6.7. Projeto de um PSS Convencional

A Figura 6.14 ilustra um PSS convencional, que é basicamente o mesmo modelo mostrado na Figura 6.9 incluindo-se o bloco “G(s)”, que constitui a função compensadora do PSS. Conhecendo-se as funções “G” e “Gd” já identificadas, e a função “C” devidamente calculada, é fácil obter Gs conforme procedimento bem conhecido da teoria de controle.



**Fig.6.14:** Diagrama de Blocos de um PSS Convencional.

Desconsiderando o efeito da referência de tensão, é fácil obter a expressão (6.26) que relaciona a tensão terminal ( $V_t$ ) e a velocidade do gerador ( $w\Delta$ ), ou a função de transferência (6.27). A condição ideal para rejeitar perturbações de carga (modelagem realizada pela variação da frequência ou da rotação do gerador  $w\Delta$ ), é que a parcela  $G_d + C.G.G_s$  seja nula, resultando na expressão (6.28), onde o módulo desta expressão é a função de compensação do PSS.

$$V_t = G_d.W\Delta + C.G.G_s.W\Delta - C.G.V_t \quad (6.26)$$

$$V_t/W\Delta = (G_d + C.G.G_s)/(1 + C.G) \quad (6.27)$$

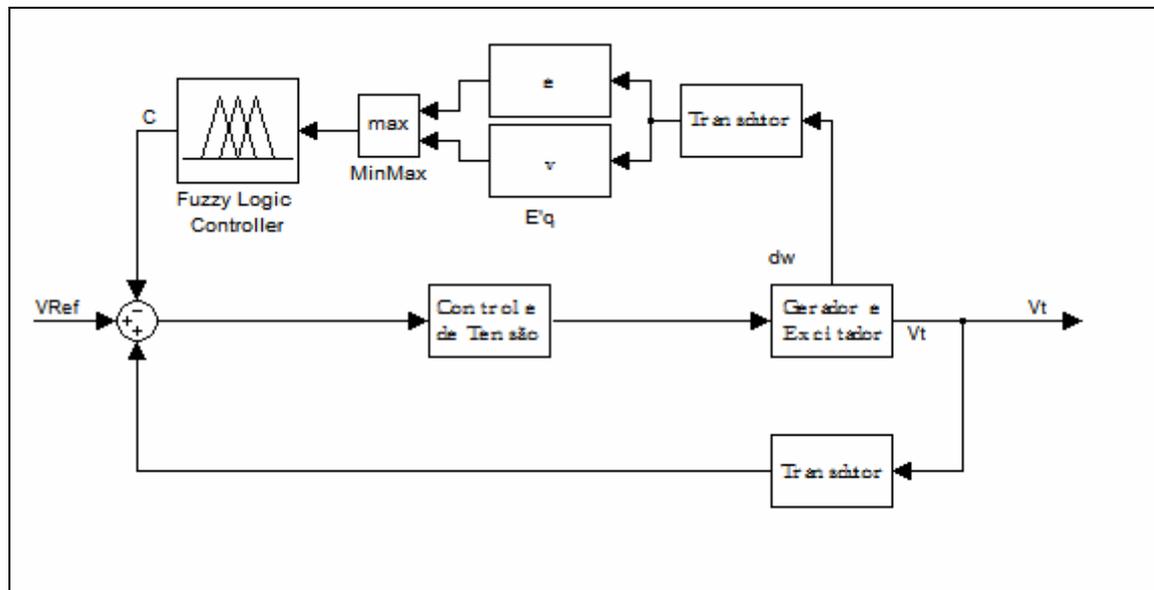
$$G_s = -G_d/(C.G) \quad (6.28)$$

Substituindo os valores numéricos de  $G$ ,  $C$ ,  $G_d$  e aproximando para uma função de primeira ordem com um par de zeros e pólo dominantes, tem-se a função (6.29).

$$G_s(s) = \frac{a_{1w}s + a_{0w}}{b_{1w}s + 1} = \frac{1,035s + 11,82}{2,07s + 1} \quad (6.29)$$

## 6.8. Projeto de um *PSS Fuzzy*

O *PSS Fuzzy* será utilizado tanto para o *AVR* clássico quanto para o *AVR Fuzzy*, e será baseado no modelo apresentado na Figura 6.15.

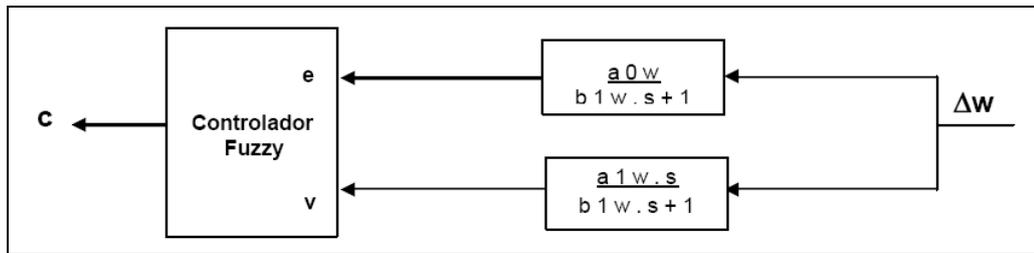


**Fig.6.15:** Diagrama Esquemático do *PSS Fuzzy*.

O sinal de compensação do *PSS* é derivado da frequência do gerador ou da velocidade da turbina, sendo processado em compensadores convencionais tipo avanço ou atraso de fase. Nesse item será empregado um sistema nebuloso para sintetizar o sinal auxiliar do *PSS* a partir da variação da velocidade do gerador.

O bloco de controle de tensão será detalhado nos próximos capítulos sobre duas formas distintas. Uma através da modelagem clássica usando um compensador *Lead/Lag* e outra utilizando os conceitos da Lógica *Fuzzy*.

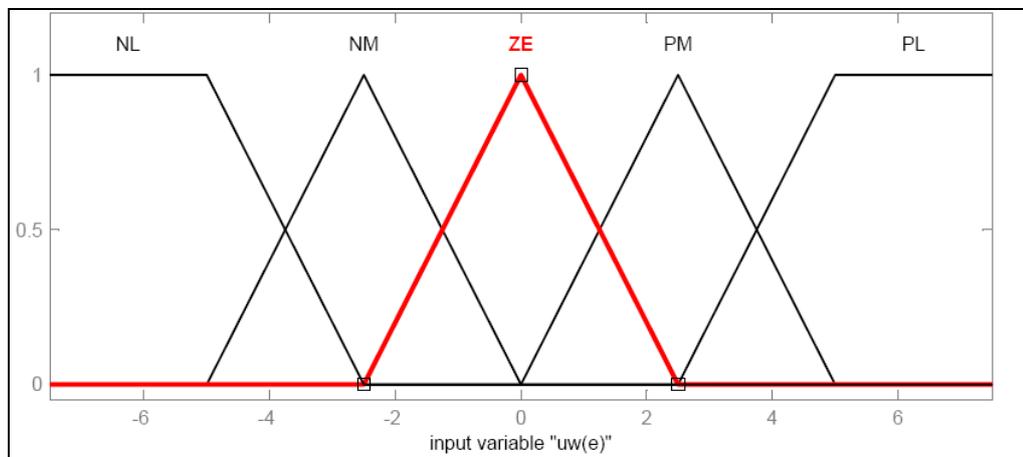
O bloco compensador *Fuzzy* tem a finalidade de substituir um compensador convencional *Lead/Lag* no *PSS* (Figura 6.15), e será usado para minimizar as não linearidades do processo. As funções passa-baixas (bloco *e*) e passa-altas (bloco *v*) filtram a informação do sinal de velocidade do gerador, fornecendo o comportamento de avanço ou atraso de fase das entradas do compensador nebuloso. A Figura 6.16 mostra as funções de cada filtro.



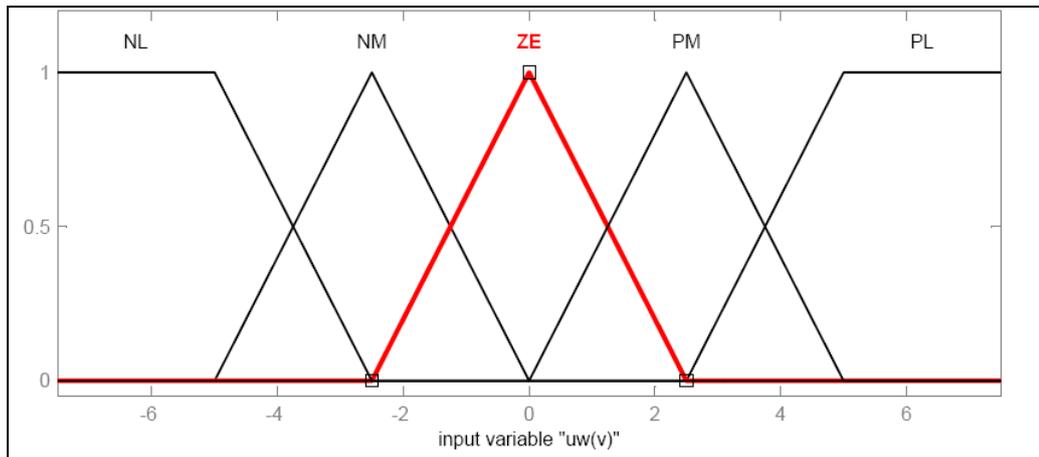
**Fig.6.16:** Detalhe dos Filtros Passa Alta e Passa Baixa do *PSS Fuzzy*.

Os parâmetros dos filtros são os mesmos utilizados no *PSS* convencional (equação (6.29)). As regras do controlador difuso são as mesmas apresentadas na Tabela 5.2. Serão utilizados operadores tipo max-produto nas inferências nebulosas e técnica de defuzificação similar ao centro de área. As funções de pertinência empregadas estão ilustradas nas Figuras 6.17, 6.18 e 6.19 sendo que os universos de discurso das mesmas foram escolhidos em função da excursão dos sinais da instrumentação de tensão e da rotação do gerador.

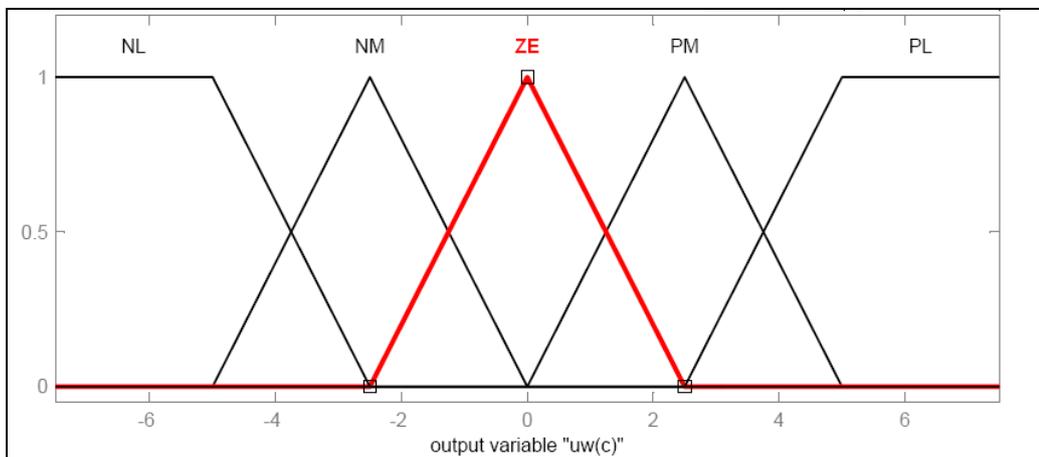
O programa do *PSS Fuzzy* foi implementado em linguagem “C” para que se possa executar o ensaio em tempo real. Esse programa será usado tanto nos ensaios com o *AVR* tradicional quanto do *AVR Fuzzy*.



**Fig.6.17:** Sinal de entrada (e) do *PSS Fuzzy*.



**Fig.6.18:** Sinal de entrada (v) do PSS Fuzzy.



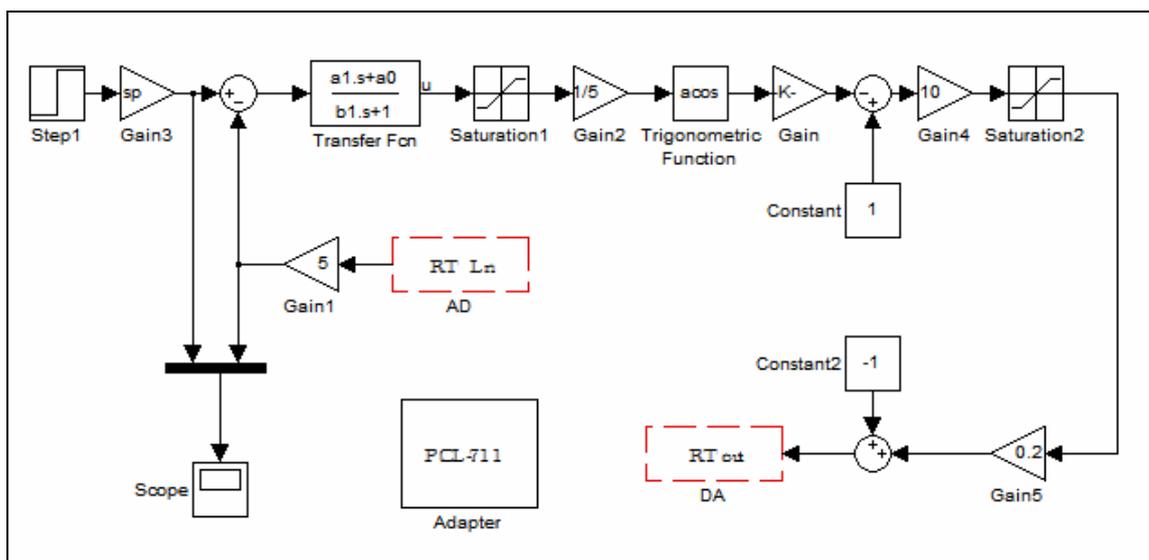
**Fig.6.19:** Sinal de saída (c) do PSS Fuzzy.

## 6.9. Ensaios com o Compensador AVR Clássico

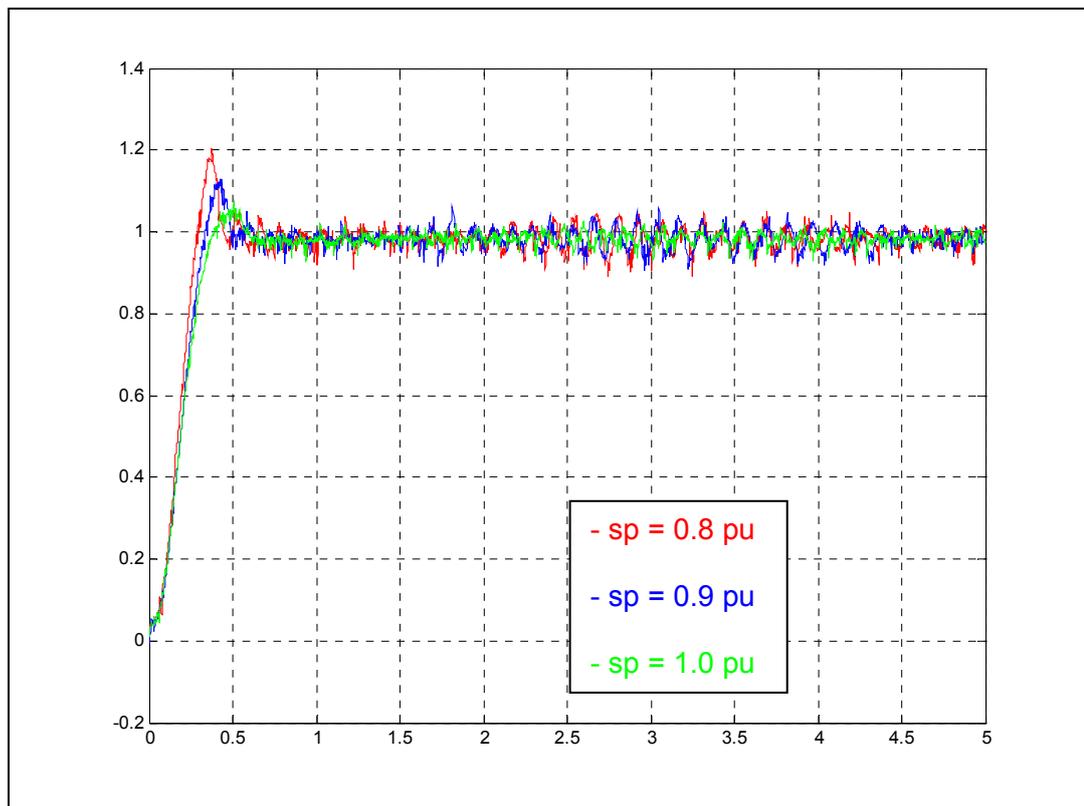
Os ensaios do AVR convencional no processo real foram divididos em duas partes, sendo que a primeira foi feita através do *Simulink/MatLab* utilizando um *toolbox* de tempo real. A outra utilizará um programa implementado em linguagem C.

Para o ensaio feito no *Simulink* foi utilizado o diagrama apresentado na Figura 6.20, onde os blocos *RT In* e *RT Out* simbolizam a entrada e a saída analógica da placa de aquisições modelo PCL-711 descrita no Anexo 3.

Nesse diagrama podemos verificar uma entrada tipo degrau e a variável *sp* (set point) que determina o nível de referência (0.8, 0.9 e 1.0 pu). Foi utilizado o compensador calculado pela equação (6.24) e a função de linearização indicada na Figura 6.8 (do retificador e ponte tiristorizada). Os demais blocos são utilizados para limitar os valores práticos da saída analógica da placa de aquisição de dados. Os ensaios foram realizados com o gerador operando de forma isolada, e com situações de carga distintas (plena carga, meia carga e a vazio). As curvas apresentadas nas Figuras 6.21, 6.22 e 6.23 mostram os resultados obtidos. Para essa etapa de testes não foi empregado o *PSS*, apenas o *AVR*.

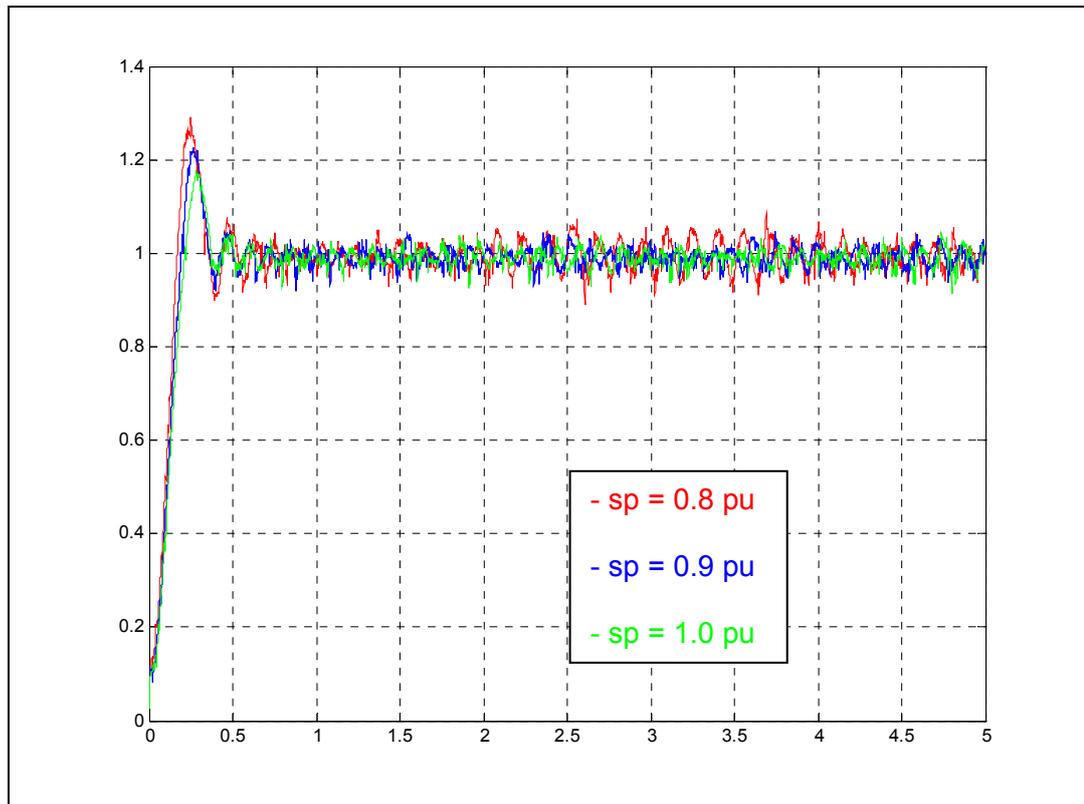


**Fig.6.20:** AVR Clássico Implementado no *Simulink*.



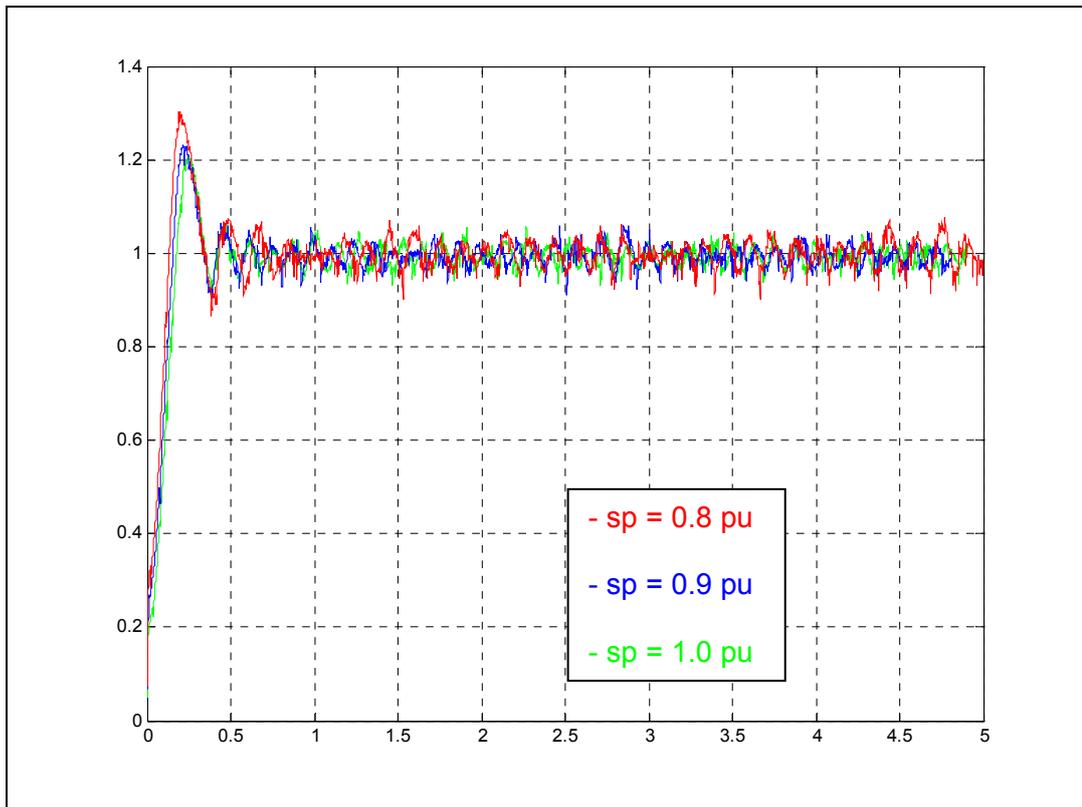
**Fig.6.21:** Resposta Reais do AVR Clássico com Carga Plena.

Os valores no eixo horizontal estão em segundos e no eixo vertical em p.u. Para o AVR em plena carga o máximo pico foi de aproximadamente 20% para um *set-point* de 0,8 p.u., e de 8% para um *set-point* de 1 p.u. O tempo de amortecimento teve pouca variação, sendo próximo de 0,6 segundos.



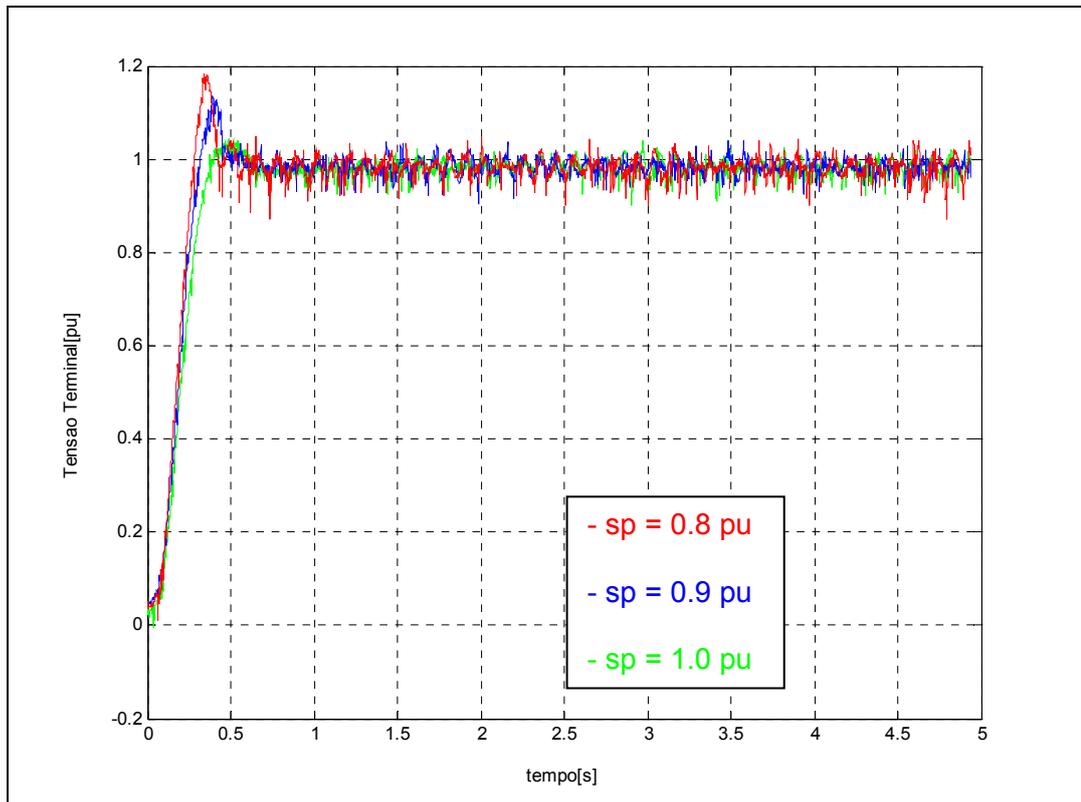
**Fig.6.22:** Resposta do AVR Convencional em Meia Carga.

Para o ensaio com o sistema de geração operando em meia carga, o máximo pico também ocorreu para o *set-point* de 0,8 p.u., sendo que o valor atingiu a marca máxima de 30%. Para o valor de 1 p.u. o valor obtido ficou em torno de 20%. O tempo de amortecimento ficou na ordem de 0,9 segundos. A Figura 6.23 mostra que com o gerador em vazio os resultados foram muito semelhantes aos encontrados o sistema a meia carga.

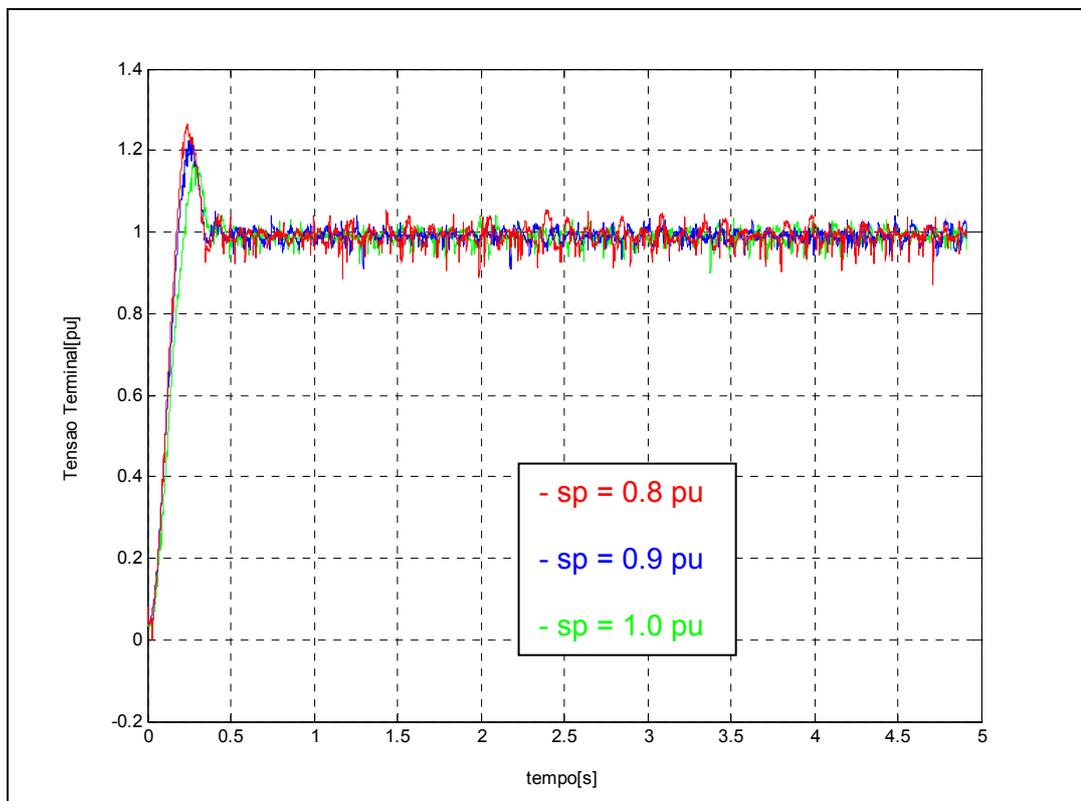


**Fig.6.23:** Resposta do AVR Clássico com Gerador a Vazio.

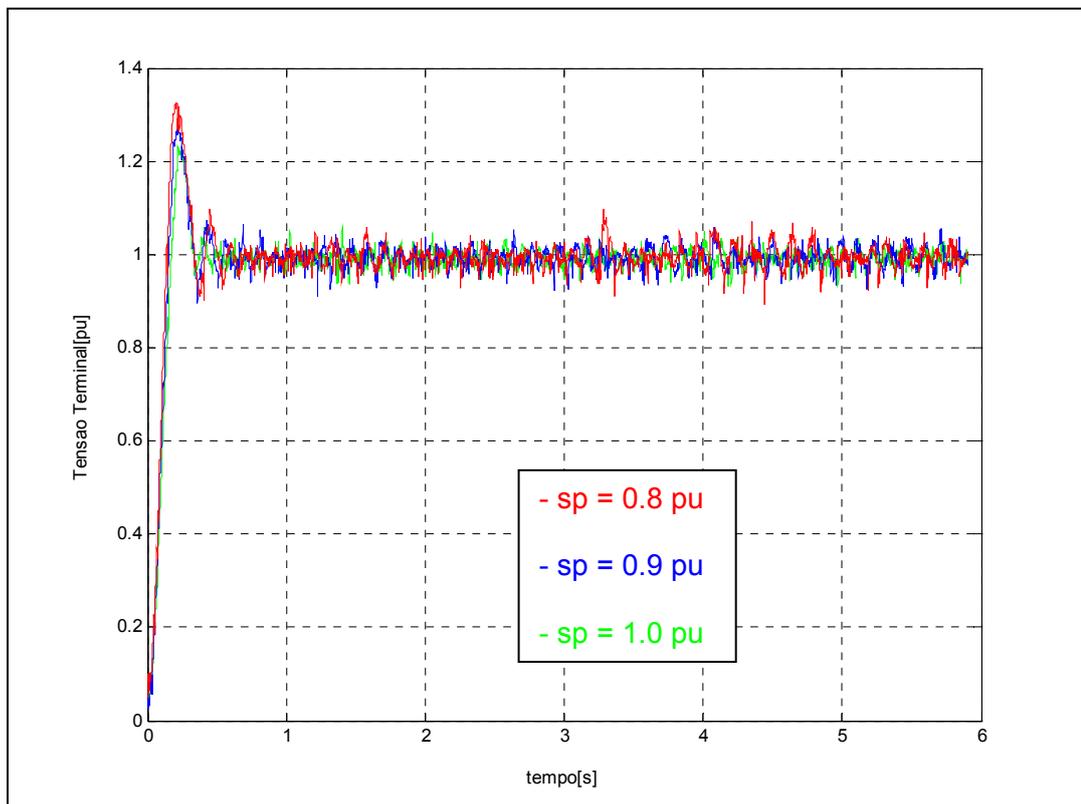
Foi desenvolvido um programa em linguagem C para realizar a função de um compensador *Lead/Lag* em tempo real acionando a placa de aquisição de dados. A descrição do código utilizado pode ser encontrada no Anexo 4. Os dados dos ensaios são gravados em arquivos e por meio dos programas “reader01” e “reader02” incluídos nos Anexos 5 e 6 é possível gerar os gráficos dos ensaios. Foram realizados os mesmo testes apresentados anteriormente e os resultados podem ser vistos nas Figuras 6.24, 6.25 e 6.26. Os resultados obtidos foram bem semelhantes aos anteriores com o programa de controle implementado no *Simulink*.



**Fig.6.24:** Resposta do AVR Clássico Implementado em “C” com Carga Plena.



**Fig.6.25:** Resposta do AVR Tradicional “C” em Meia Carga.

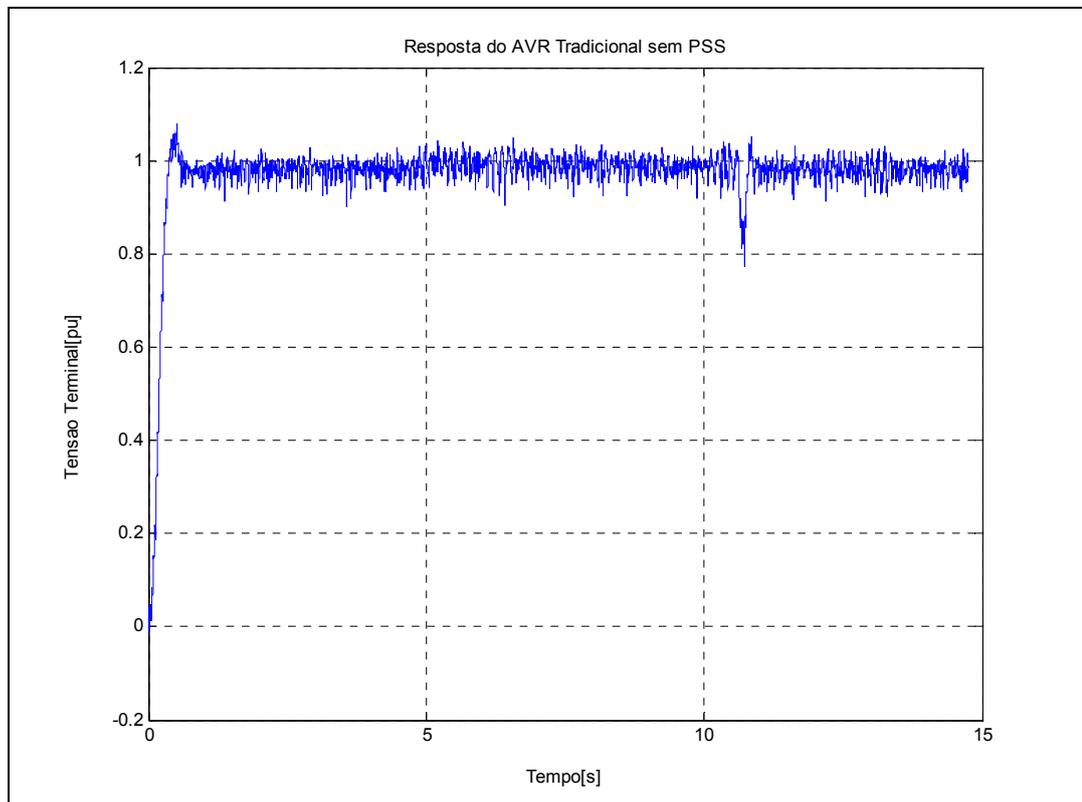


**Fig.6.26:** Resposta do AVR Convencional em “C” e Gerador a Vazio.

Para finalizar os testes com o regulador de tensão convencional foram realizados ensaios com variação brusca de carga. Nos ensaios foram estabelecidas as seguintes diretrizes:

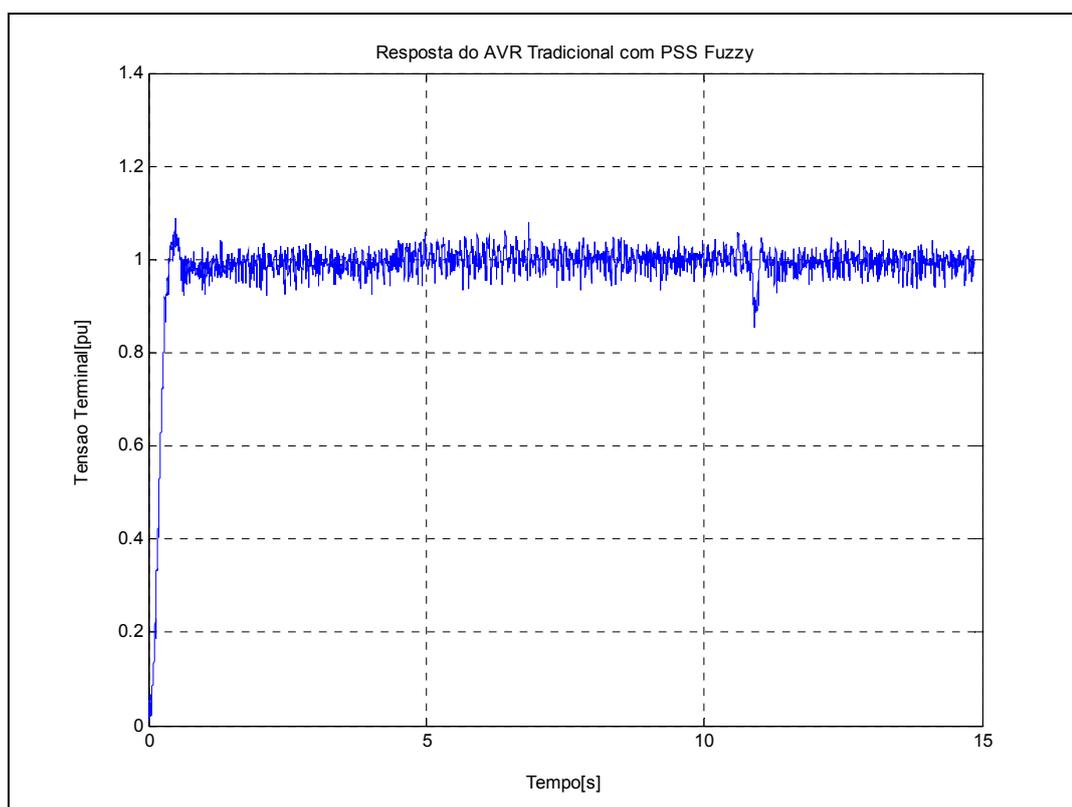
- O gerador estará operando de forma isolada;
- A partida será feita em plena carga;
- Será estabelecido um tempo para o sistema entrar em regime;
- Após a estabilização do sistema será retirada a carga do sistema;
- Será considerado um outro intervalo de tempo para o sistema estabilizar;
- Novamente será aplicada a carga nominal do sistema.

A Figura 6.27 traz o resultado do ensaio com variação de carga com referência de 1 p.u. A resposta mostra que o sistema apresenta modificação de tensão com a saída das cargas do sistema que são compensadas. Na retomada da carga o sistema sofre um afundamento de tensão superior a 20% do seu valor nominal e após uns 0,5 segundos a tensão volta ao seu valor nominal.



**Fig.6.27:** Resposta do AVR Conv. Implementado em “C” a Variação de Carga.

A figura 6.28 mostra agora a resposta do sistema de controle incluindo o *PSS Fuzzy* na malha de compensação. O sistema responde melhor as variações de carga, mesmo com o processo operando isoladamente. Na retomada da carga o sistema sofreu um afundamento em sua tensão terminal de aproximadamente 15%, apresentando um desempenho superior ao encontrado no mesmo sistema sem um *PSS*.



**Fig.6.28:** Resposta do AVR Clássico com *PSS Fuzzy*.

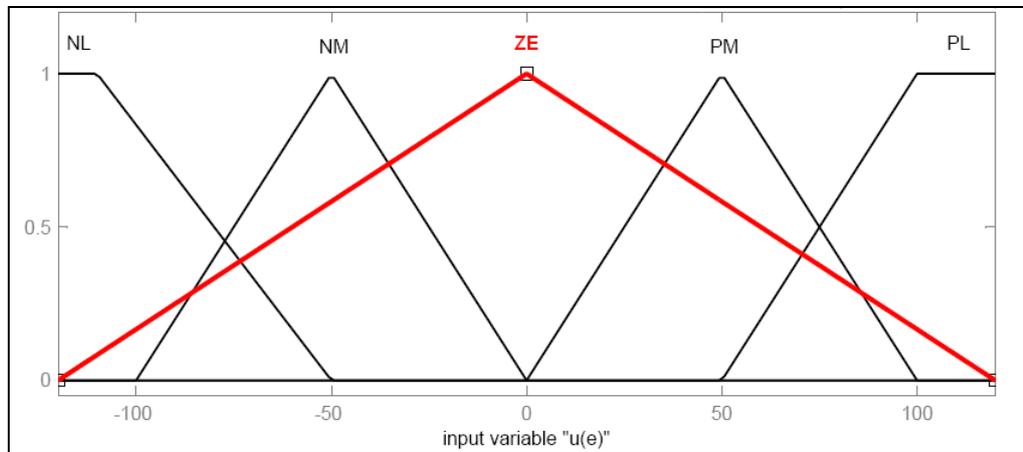
### 6.10. Projeto de um Compensador *AVR Fuzzy*

Será desenvolvido agora um *AVR Fuzzy*. Para esse projeto será utilizada a metodologia descrita no Capítulo 4.

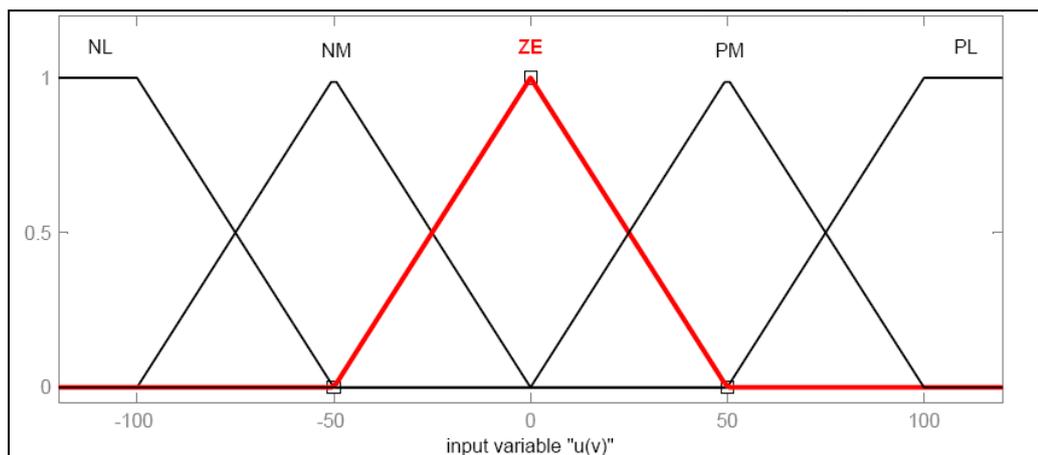
O *AVR Fuzzy* foi desenvolvido em linguagem “C”, pois não possível utilizar o *toolbox* de lógica difusa do *Simulink* em aplicações de tempo real. O código do programa está no Anexo 7.

A Figura 4.2, página 20, ilustra o diagrama utilizado para o projeto do compensador nebuloso. O sinal de controle é estabelecido em função da tensão nos terminais do gerador, sendo que existe uma parcela agregada pelo estabilizador de potencia (*PSS*). O *PSS Fuzzy* será o mesmo utilizado neste capítulo. O programa que inclui o *AVR Fuzzy* e o *PSS Fuzzy* encontra-se no Anexo 8. Os parâmetros  $a_0$ ,  $a_1$  e  $b_1$  são os mesmos utilizados no compensador tradicional mostrados na equação (6.24). As regras do controlador *Fuzzy* são as mesmas apresentadas na Tabela 4.1 mostrada no capítulo 4. As funções de pertinência empregadas estão

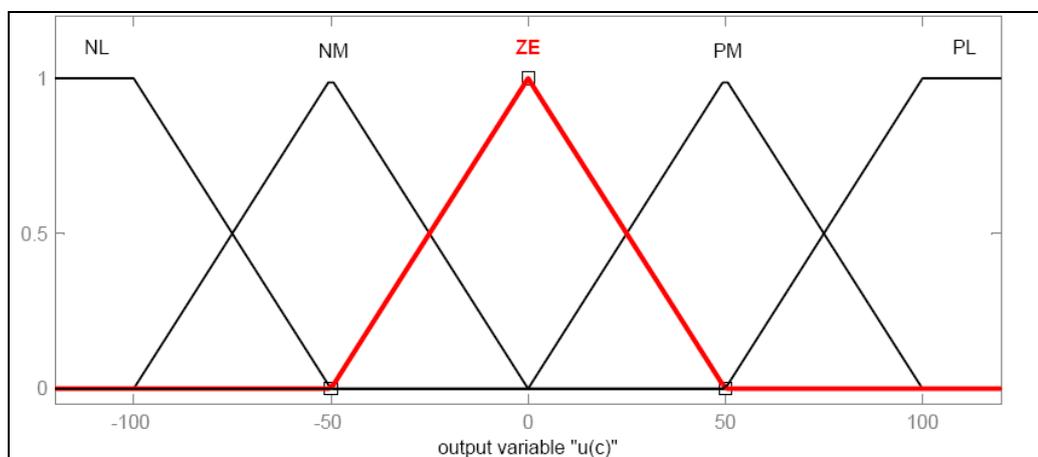
exemplificadas nas Figuras 6.29, 6.30 e 6.31. Os universos de discurso foram determinados através da medição dos sinais de entrada e saída do AVR convencional.



**Fig.6.29:** Sinal de entrada (e) do AVR Fuzzy.



**Fig.6.30:** Sinal de entrada (v) do AVR Fuzzy.

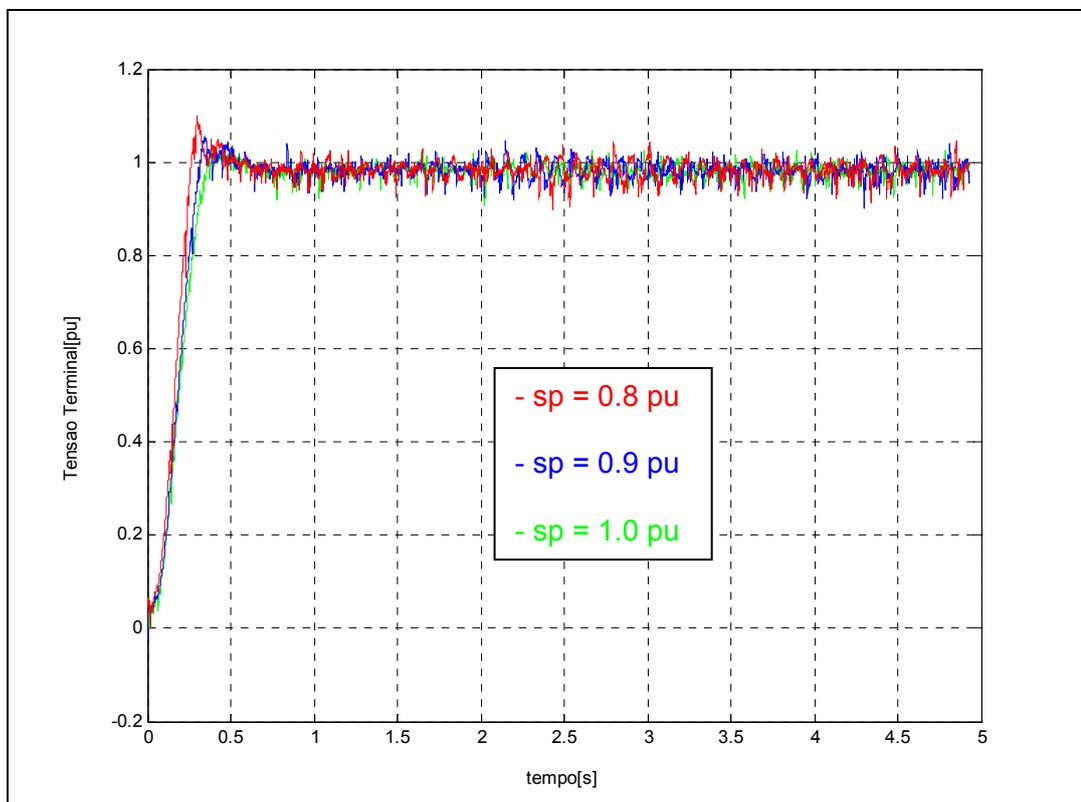


**Fig.6.31:** Sinal de Controle (u) do AVR Fuzzy.

Em Pinheiro (2000) são mostradas metodologias para escolha das funções de pertinência e ganhos de escalonamento para a sintonia de controladores difusos. As não-linearidades do processo controlado são compensadas pelas ações dos compensadores nebulosos.

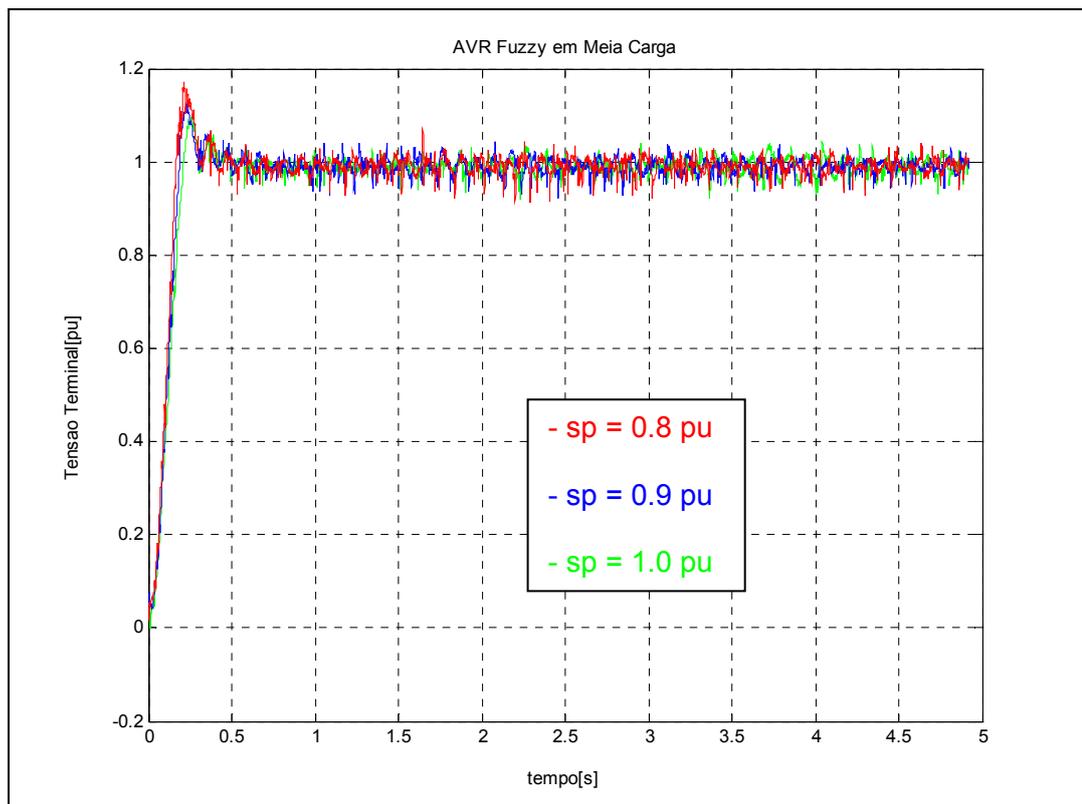
### 6.11. Ensaio com o Compensador *AVR Fuzzy*

Para o compensador nebuloso foram repeditos os testes de desempenho com diferentes condições de carga, de forma similar aos ensaios feitos com o *Lead/Lag* convencional. Inicialmente é incluído o *PSS Fuzzy*. A Figura 6.32 mostra o resultado encontrado com o *AVR Fuzzy* para o sistema em plena carga.

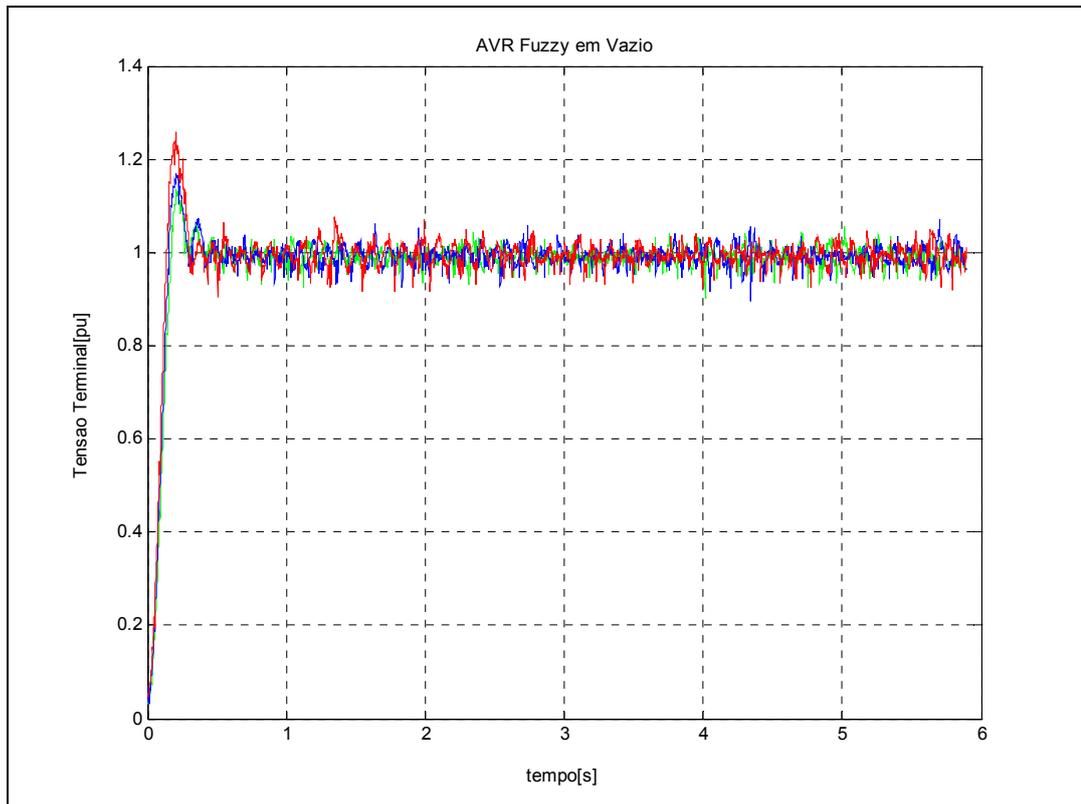


**Fig.6.32:** Resposta do *AVR Fuzzy* em Plena Carga.

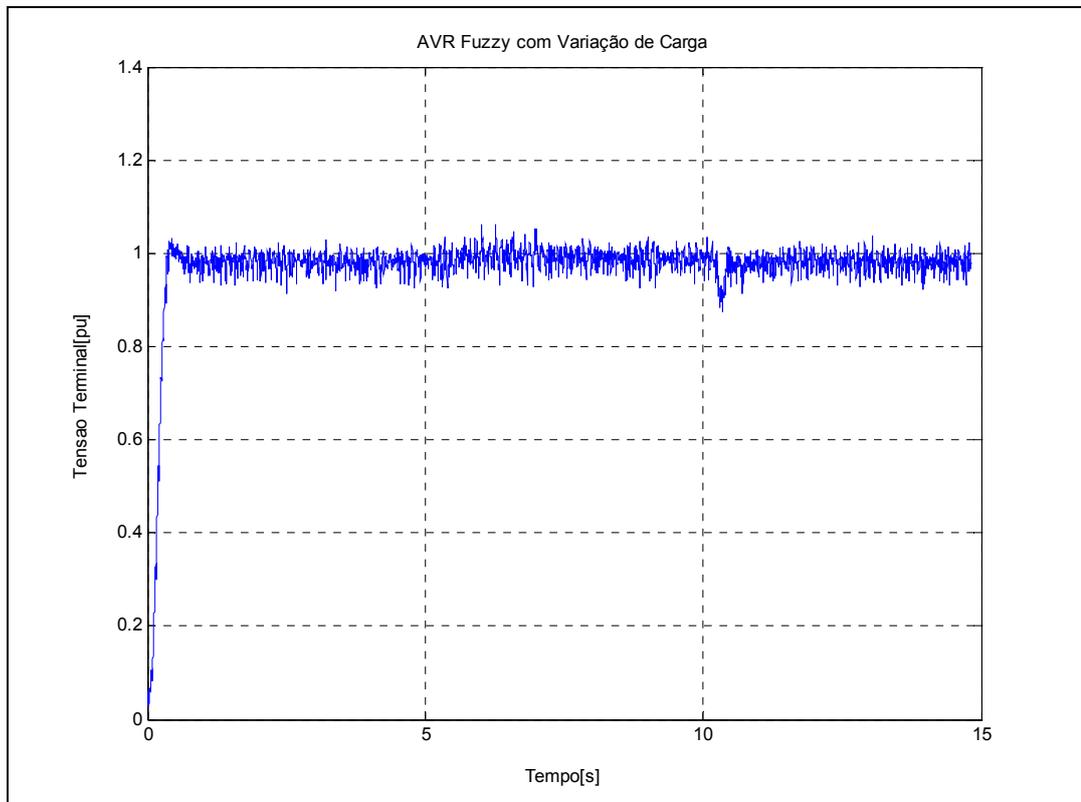
Comparando o desempenho do compensador nebuloso com o apresentado com o controlador clássico podemos perceber que o *AVR Fuzzy* apresenta valores de máximo pico menores que 10%. O mesmo ensaio foi repetido para o sistema em meia carga e a vazio (Figuras 6.33 e 6.34). A Figura 6.35 ilustra o ensaio com variações de carga no sistema. Os resultados obtidos foram melhores do que com o *AVR* clássico.



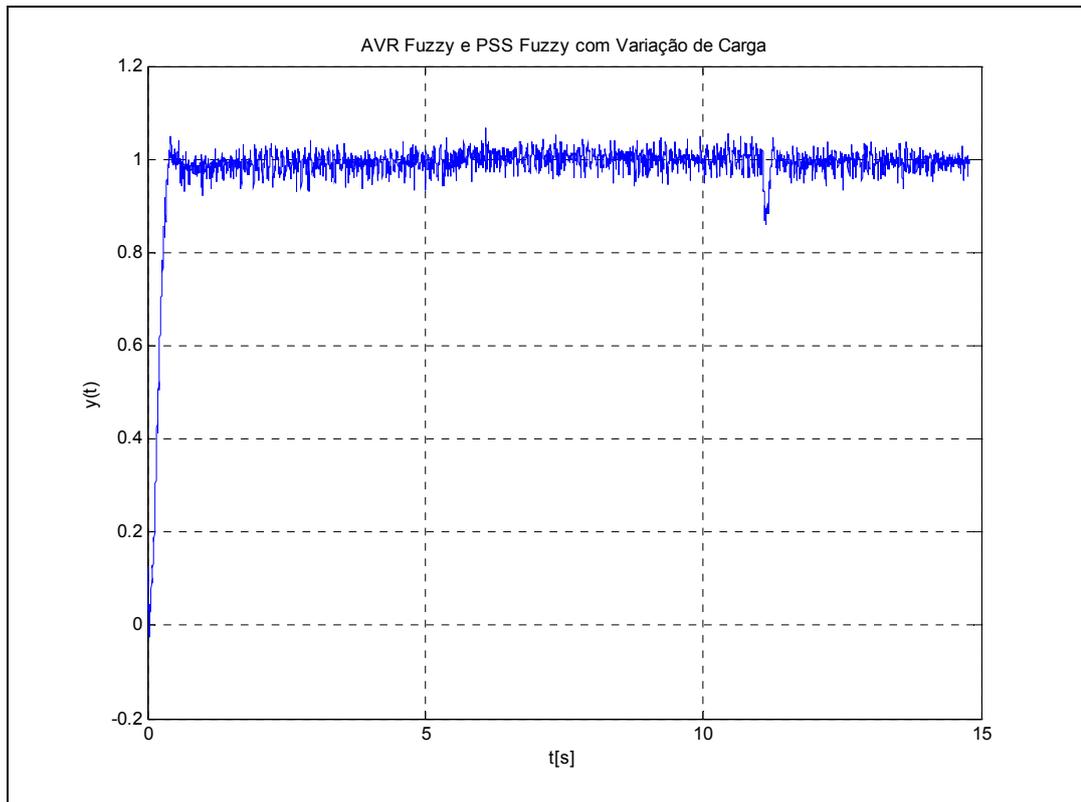
**Fig.6.33:** Resposta do *AVR Fuzzy* com Meia Carga.



**Fig.6.34:** Resposta do *AVR Fuzzy* com Gerador a Vazio.

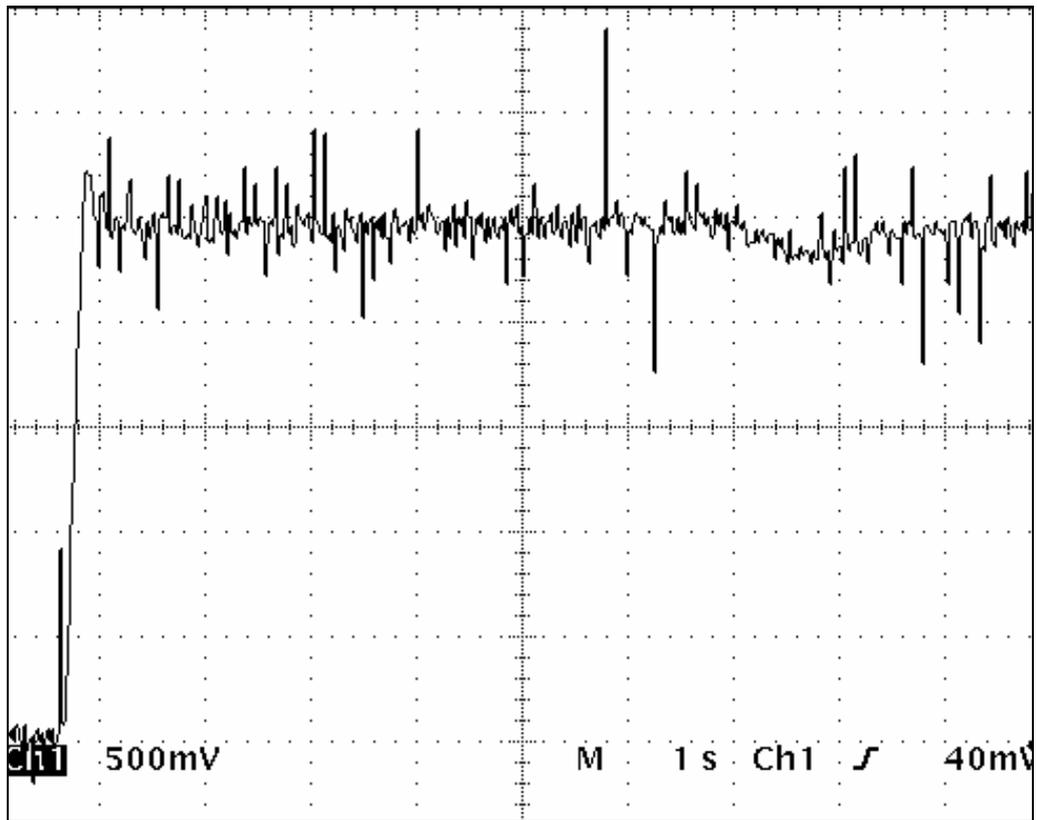


**Fig.6.35:** Resposta do *AVR Fuzzy* a Variação de Carga no Sistema.



**Fig.6.36:** Resposta do *AVR Fuzzy* com *PSS Fuzzy* a Variação de Carga.

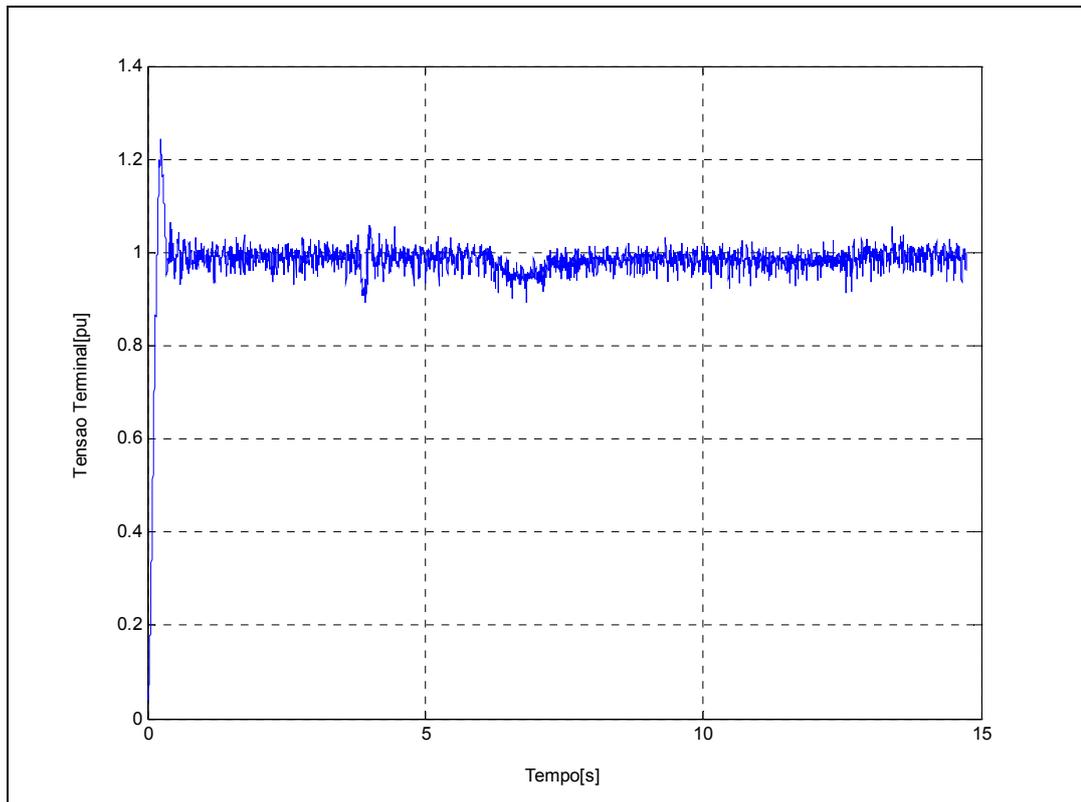
A Figura 6.37 mostra o registro da resposta do sistema utilizando um osciloscópio digital. O comportamento é semelhante ao obtido pelos dados reais gravados e processados pelo MATLAB. Verifica-se que o transdutor de tensão apresenta ruídos que são provenientes do chaveamento dos tiristores das pontes conversoras.



**Fig.6.37:** Oscilografia do AVR Fuzzy com PSS Fuzzy com Variação de Carga.

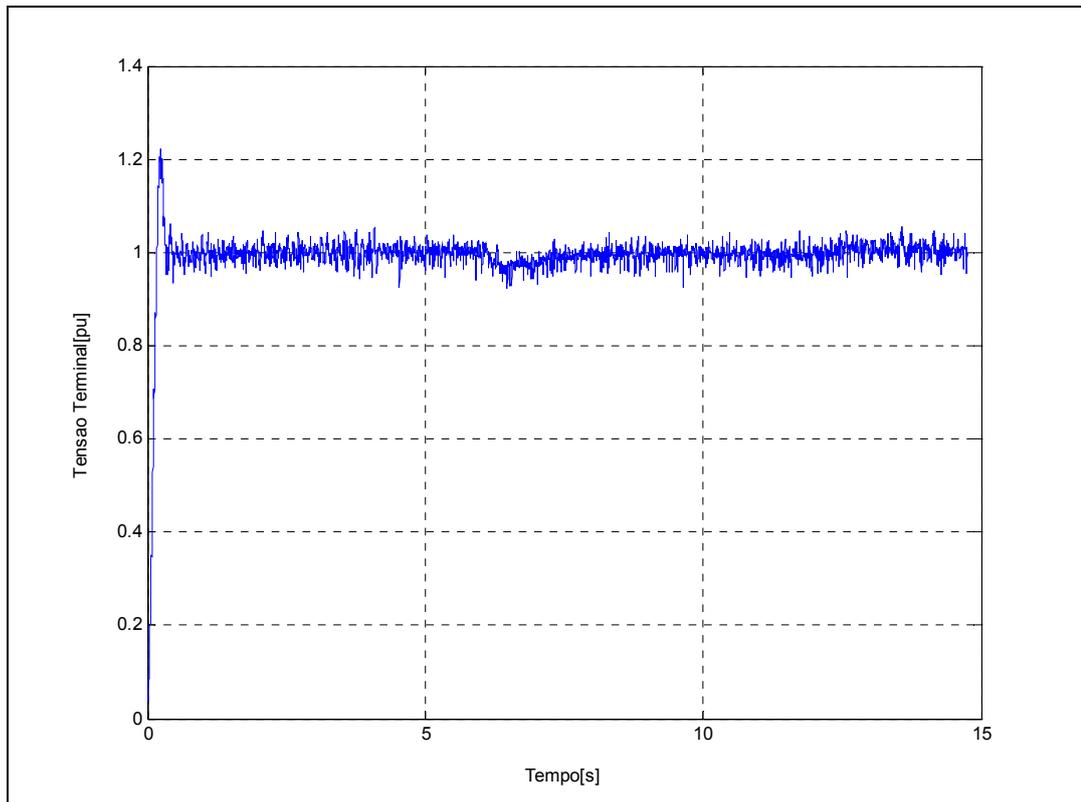
## 6.12. Ensaios de Paralelismo de Geradores com AVR Convencional

Para concluir os testes práticos do AVR e PSS Fuzzy foi elaborado um ensaio de paralelismo entre geradores. Foi utilizado primeiramente um compensador de avanço e atraso de fase tradicional. Será estabelecido um tempo de quatro segundos para que o primeiro gerador entre em regime permanente após sua partida. Após este tempo será realizado o paralelismo com o segundo gerador. Será aguardado um intervalo de aproximadamente dois segundos para o sistema entrar em regime. Depois será acrescentado carga total. E um tempo após será desfeito o paralelismo. A Figura 6.38 mostra o resultado do ensaio do AVR tradicional sem o uso do PSS estabelecendo-se o paralelismo.



**Fig.6.38:** Paralelismo com AVR Convencional e sem PSS.

É possível verificar que o sistema apresenta um tempo de resposta maior que com operação isolada, para o processo de regulação de tensão ser compensado adequadamente. A Figura 6.39 mostra a resposta do sistema com paralelismo e com AVR clássico junto com um *PSS Fuzzy*. Nota-se que em relação a figura anterior, a recuperação ao afundamento de tensão do sistema foi melhor.

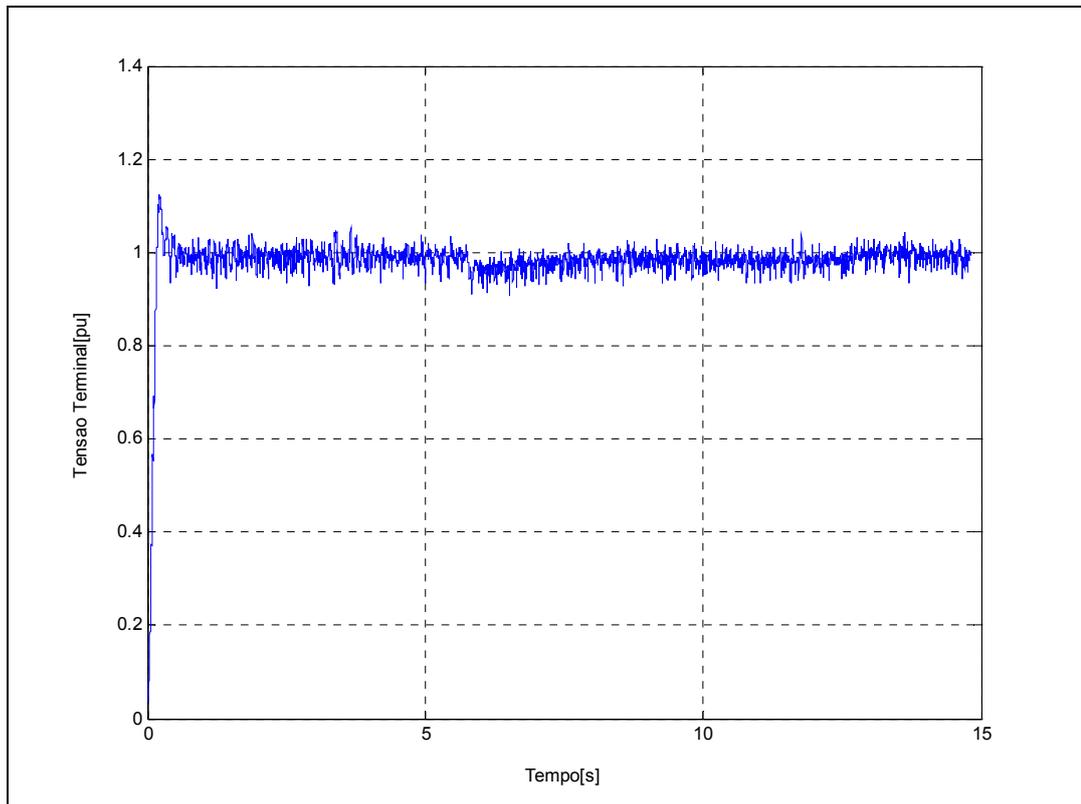


**Fig.6.39:** Paralelismo com *AVR Clássico* e com *PSS Fuzzy*.

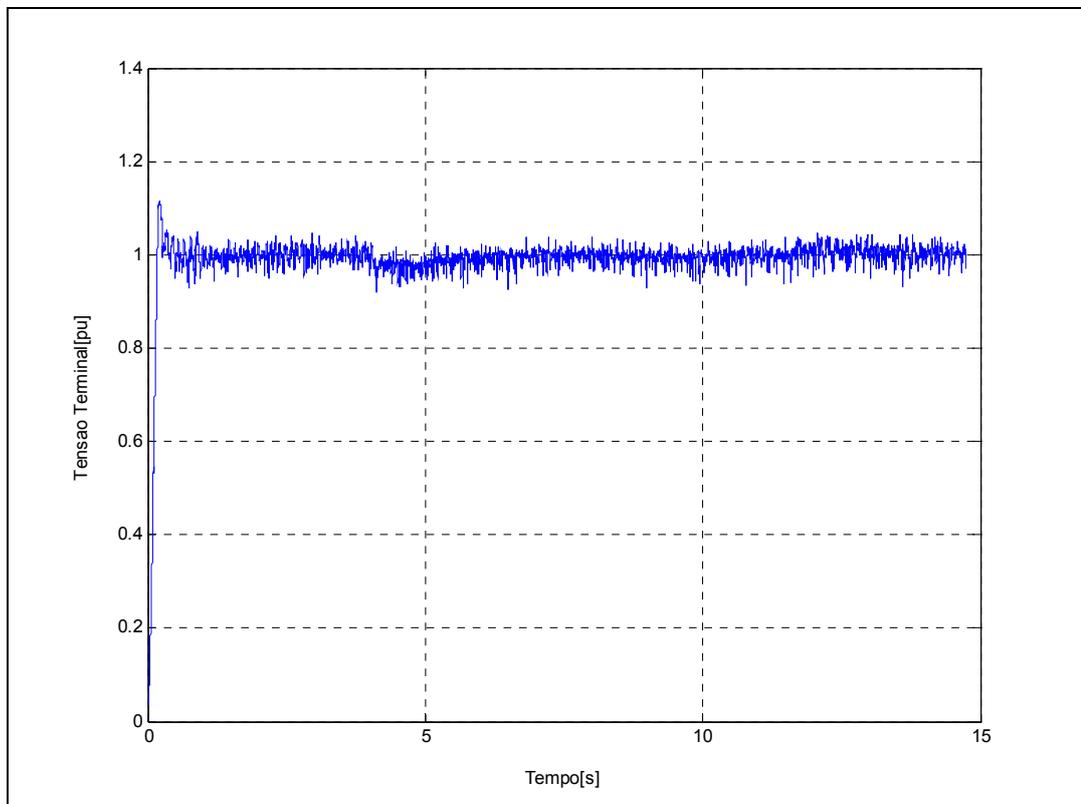
### 6.13. Ensaio de Paralelismo com *AVR Fuzzy*

Foram repetidos os testes apresentados no item 6.12. A Figura 6.40 mostra o resultado com o *AVR Fuzzy* e sem o *PSS*. O compensador difuso reduziu o pico inicial da tensão gerada e também reduziu os efeitos das variações de carga no paralelismo.

A Figura 6.41 mostra agora o ensaio com o *AVR Fuzzy* e o *PSS Fuzzy*. A utilização conjunta dos dois compensadores resultou em uma melhora na operação paralela dos geradores, o nível de afundamento de tensão foi menor.



**Fig.6.40:** Paralelismo com *AVR Fuzzy* e sem *PSS*.



**Fig.6.41:** Paralelismo com *AVR Fuzzy* e com *PSS Fuzzy*.

## CAPÍTULO 7 - CONCLUSÃO

### 7.1. Conclusão

O objetivo desse trabalho foi o de apresentar uma metodologia para o projeto de um compensador *AVR Fuzzy* e um *PSS Fuzzy* com características de avanço e atraso de fase para regulação e compensação de sistemas de geração de energia elétrica. O trabalho oferece uma alternativa a *AVRs* e *PSSs* convencionais, com a vantagem de uma maior robustez as variações e oscilações comuns nos sistemas elétricos de potência.

A metodologia desenvolvida mostrou-se eficiente e garantiu um desempenho melhor do que com compensadores convencionais, que geralmente não apresentam boas respostas a variações de condições operacionais e não-linearidades inerentes a sistemas reais.

Foi apresentada uma visão geral sobre compensadores convencionais de característica de avanço e atraso de fase e de compensadores nebulosos, ressaltando as principais características destas técnicas, bem como um método de projeto para cada uma delas.

Para comprovar a eficácia da implementação proposta, foi realizada a montagem de uma bancada experimental. Os resultados práticos obtidos comprovaram que a estrutura proposta é eficaz e pode melhorar a eficiência de sistemas de geração.

Assim é possível ressaltar que os objetivos deste trabalho de mestrado foram atingidos, e que seus resultados abrem perspectivas futuras para a continuidade do mesmo. Estudos mais aprofundados podem resultar na obtenção de controladores mais adequados para implementações reais em sistemas de grande porte.

Finalmente, pode-se dizer que os artigos que tratam da utilização de lógica nebulosa em reguladores de tensão e estabilizadores de sistema de potência, se constituem basicamente de simulações numéricas e apresentam poucos resultados experimentais. Esse estudo contribui principalmente para disponibilizar um trabalho com resultados práticos que enriquece os trabalhos dessa linha de pesquisa.

---

## 7.2. Trabalhos Futuros

Como proposta de continuidade deste trabalho, pode-se indicar os seguintes pontos que devem ser investigados adequadamente:

A utilização de microcontroladores ou processadores digitais de sinais (*DSP`s*) para implementar a lógica do *AVR* e do *PSS Fuzzy*, ao invés de utilizar um micro-computador pessoal e placas de aquisição de dados;

Estudo de casos (através de simulações numéricas) de modelos do sistema elétrico nacional com diversos geradores, linhas, pontos de carga, etc.

Estender o estudo de compensação difusa para os reguladores de velocidade.

---

**BIBLIOGRAFIA**

- 1) Almeida, A. T. L., “Máquinas Síncronas”, Grupo de Estudo em Manutenção Eletro-Eletrônica e Instalações, Universidade Federal de Itajubá, 2000.
  - 2) Anderson, P. M., Fouad, A. A. “Power System Control and Stability”, The Iowa State University Press, 1993.
  - 3) C Liu, M. Ishimaru, R. Yokoyama, K. Koyanagi, “Stabilizing Control for Long-term Low Damping Oscillation in Multi-Regional Power System”, Tokyo Metropolitan University, Minamiosawa, TEPCO Systems Corporation, Tokyo, Japan, 2001.
  - 4) Chian-Chuang Ding, King-Tan Lee, Chee-Ming Tsai, Tsong-Liang Huang, “Optimal Design for Power System Dynamic Stabilizer by Grey Prediction PID Control”, Department of Electrical Engineering, Tamkang University, Department of Electrical Engineering, Lee-Ming Institute of Technology, Taiwan, 2004.
  - 5) Costa, A. J. A. S., Silva, A. S., “Controle e Estabilidade de Sistemas Elétricos de Potência”, Universidade Federal de Florianópolis, 2000.
  - 6) Féo, A. E., “Implementação de um Estabilizador de Sistemas de Potência Utilizando Compensador Fuzzy com Características Lead/lag”, Universidade Federal de Itajubá, 2004.
  - 7) Ferreira, A. M. D., Barreiros, J. A. L., Brito-de-Souza, J. R., “Fuzzy Power System Stabilizer Including a Fuzzy PI Controller”, Federal University of Pará, 1997.
  - 8) Fouad Mrad, Sami Karaki, Bassel Copti, “A Novel Fuzzy Logic Synchronous Machine Stabilizer”, Department of Electrical and Computer Engineering American University of Beirut Beirut, Lebanon, 1997.
-

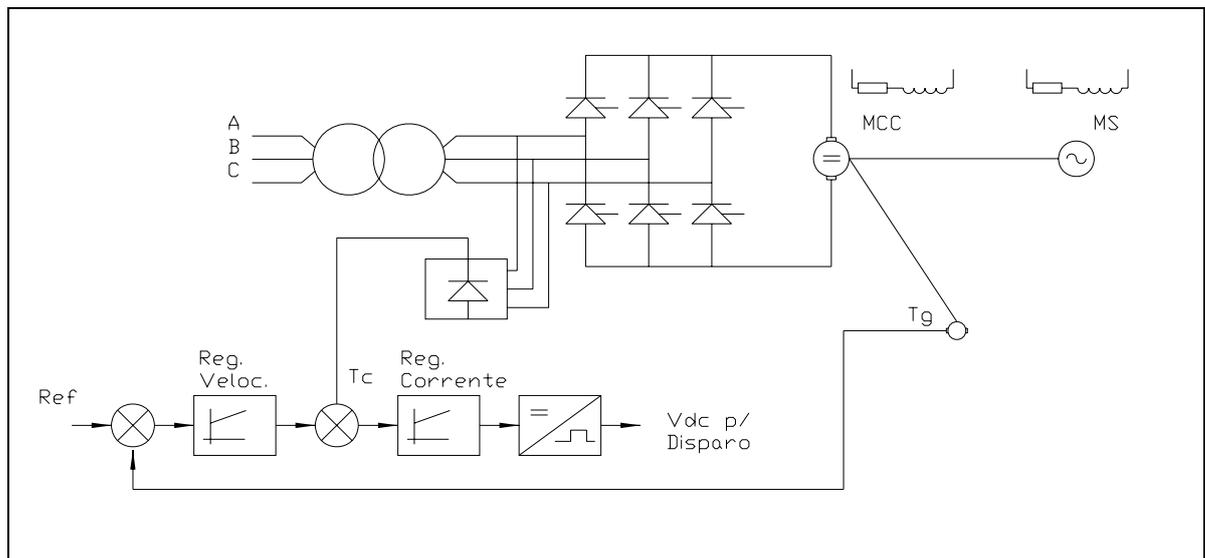
- 9) Harris, C. J., Billings. S. A. "Self-Turning and Adaptive Control: Theory and Applications", London: Peter Peregrinus Ltd, 1981, 362p.
  - 10)Hiroyuki Ukai, Yoshiaki Takagi, Jun-ichi Fujita, Hisashi Kando, Koichi Nakamura, Hideki Fujita, "Stabilizing Control of Multi-Machine Power Systems by using Gain Scheduling Control and Parameter Identification", Nagoya Institute of Technology and Chubu Electric Power Co. Inc., Japan, 2001.
  - 11)Hirsch, M. W., Smale, S., "Differential Equations Dynamical Systems, and Linear Algebra", University of California, Berkeley, 1974.
  - 12)Hiyama, T., Oniki, S., Nagashima, H., "Evaluation of Advanced Fuzzy Logic PSS on Analog Network Simulator and Actual Installation on Hydro Generators", Dept. of Electrical Engineering and Computer Science, Kumamoto University, and Research Laboratory, Kyushu Electric Power Co., Inc, Japan, 1996.
  - 13)Junior, W. B., Barreiros, J. A. L., Costa, J. C. T., Ferreira, A. M. D., "Controle Fuzzy Aplicado à Melhoria da Estabilidade Dinâmica em Sistemas Elétricos de Potência", Campos Universitário do Guamá, Belém, Pará, 2005.
  - 14)Lee, C. C., "Fuzzy Logia in Control Systems: Fuzzu Logic Controller – Part I and II, IEEE Transactions on Systems, Man and Cybernetics", vol. 20, n.2, mar/apr, 1990.
  - 15)Lee, Y., Lin, C., Hsu, K., Ghuang, "C. Design of Single-Input Fuzzy Logic Control Power System Stabilizer", Proceedings of IEEE TENCON'02, 2002.
  - 16)Mandani, E. H., Assilian S., "Applications of Fuzzy Algorithms for Control do Simple Dynamic Plant", Proc IEE, Pt. D, v.121, p. 1585-1588, 1974.
-

- 17) Natache S. D. A., Vilma A. O., Rodrigo A. R., Newton G. B., "O Uso de Sistemas de Controle Fuzzy para Estabilização de Sistemas de Potência Sujeitos a Variações Abruptas e Aleatórias em suas Condições de Operação", Depto Engenharia Elétrica, Universidade de São Paulo, 2004.
  - 18) Ogata K., "Engenharia de Controle Moderno", Universidade de Minnesota, Tradução 1982.
  - 19) Phillips, C. L., Harbor, R. "Feedback Control Systems", Prentice Hall, Englewood Cliffs, 1996.
  - 20) Pinheiro, C. A. M., "Análise e Projeto de Sistemas de Controle Fuzzy: Uma Abordagem no Domínio da Freqüência", Tese de Doutorado, Universidade Estadual de Campinas, 2000.
  - 21) Pinheiro, C. A. M., Gomide, F. A. C., "Análise e Projeto de Controladores Fuzzy por Resposta em Freqüência", XII Congresso Brasileiro de Automática, v.6, p2181-2186, 1998.
  - 22) Teixeira, M. C. M., Pietrobon, H. C., Assunção, E., "Novos Resultados Sobre a Estabilidade e Controle de Sistemas Não-Lineares Utilizando Modelos Fuzzy e LMI", Universidade Estadual Paulista e Instituto Tecnológico de Aeronáutica, São Paulo, 2000.
  - 23) Venayagamoorthy, G. K., Harley, R. G., Wunsch, D. C., "Implementation do Adaptive Critic Based Neuro Controllers for Turbo Generators in a Multimachine Power System", IEEE Transactions on Neural Networks, Vol. 14, No. 5, September 2003.
  - 24) Wenxin Liu, Ganesh K. Venayagamoorthy, Donald C. Wunsch II, "Adaptive Neural Network Based Power System Stabilizer Design", Department of Electrical and Computer Engineering, University of Missouri, Rolla, 2002.
  - 25) Zadeh, L. A., "Fuzzy sets. Inform. and Control", v.8, p.338-353, 1965.
-

## ANEXO 1: CONTROLE AUTOMÁTICO DE VELOCIDADE.

O módulo de controle automático de velocidade foi desenvolvido para garantir que a velocidade de gerador fique constante, independente das mudanças de característica da carga. Esse módulo ainda tem como função evitar variações bruscas e na corrente de alimentação da armadura do motor DC. A Figura 1.1 mostra o diagrama funcional desse regulador.

Para isso foram empregados dois controladores PID em cascata, sendo que o primeiro recebe uma referência de velocidade e é realimentado pela informação do taco-gerador. Esse sinal é ligado a um somador junto com uma referência da corrente de armadura do motor DC. O segundo controlador PID recebe o sinal de erro da saída do amplificador e gera uma rama de aceleração, limitada pelos valores de ganhos ajustados. A saída do módulo é ligada a um circuito de controle de disparo.



**Fig. Anexo 1.1:** Diagrama Esquemático do Regulador de Velocidade.

Nesse módulo de controle é possível fazer o ajuste da referência de velocidade através do Potenciômetro Ref. Veloc. da Figura 1.2, que para o estudo foi fixada na frequência nominal da rede em 60 Hz. O sinal do taco-gerador passa por um divisor resistivo para ser compatível com o valor de tensão esperado na entrada do controlador.



**Fig. Anexo 1.2:** Diagrama Frontal do Controlador Automático de Velocidade.

Para os parâmetros de velocidade também são permitidos os ajustes de tempo de resposta para as variações de velocidade (Potenciômetro Rampa Veloc.), amplitude da do sinal de saída (Potenciômetro Ganho Veloc.) e o peso que o parâmetro velocidade tem para o compensador (Potenciômetro P1).

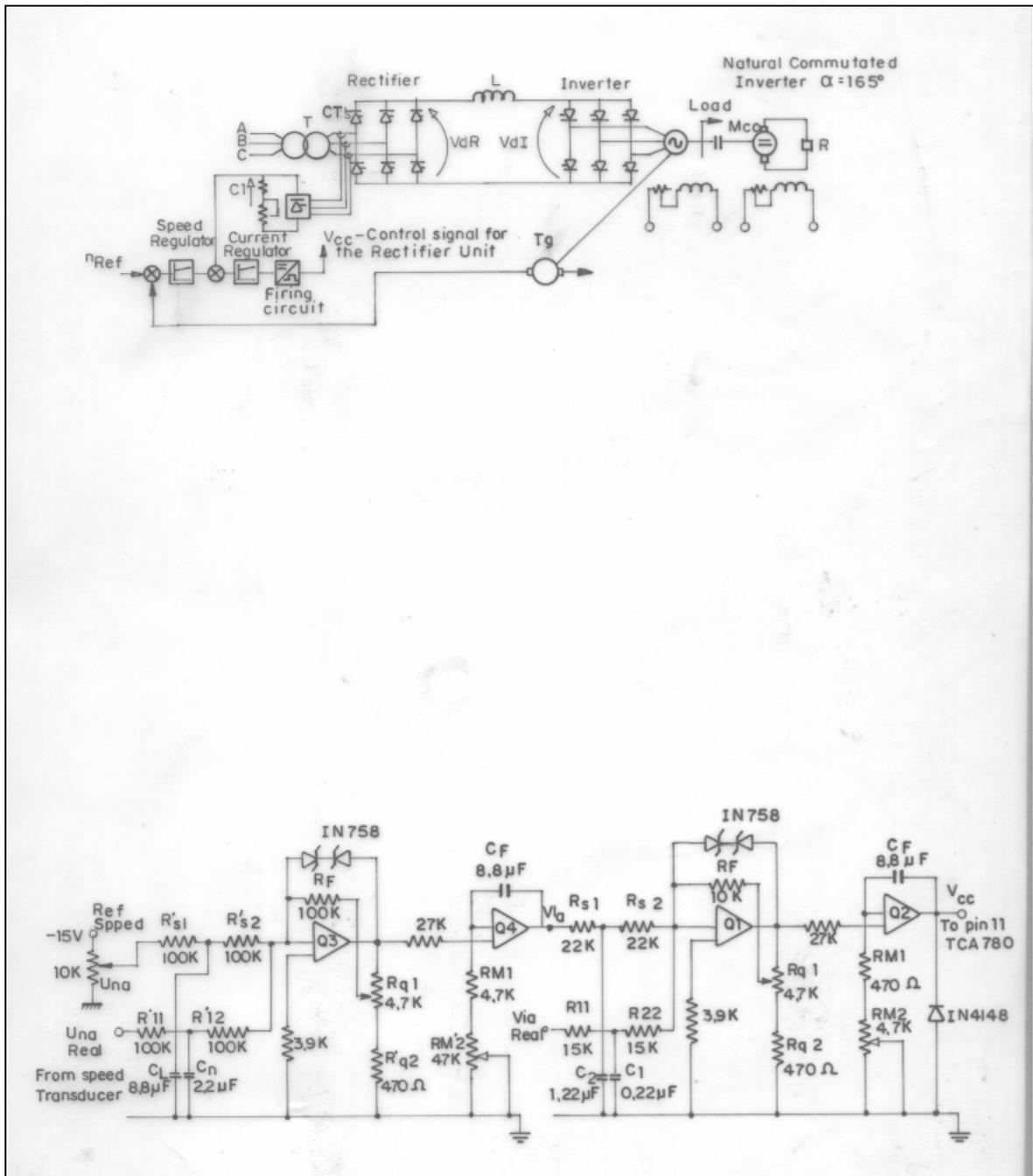
Para a compensação de corrente, são permitidos os mesmos ajustes com exceção da referência de corrente. Os ajustes são realizados pelos potenciômetros “Rampa Corrente”, “Ganho Corrente”, e “P2” mostrados na figura Anexo 1.2:

Os pontos de saídas e entradas são numerados de 1 a 14 como mostrado na figura Anexo 1.2 e são ligados conforme a tabela Anexo 1.1.

**Tab. Anexo 1.1:** Sinais do Regulador de Velocidade.

<b>Item</b>	<b>Tipo de Sinal</b>	<b>Descrição</b>	<b>Interligação</b>
1	Saída	Terra do Sinal de Controle	Disparo
2	Saída	Sinal de Controle	Disparo
3	Entrada	Referência do PID de Corrente	7
4	Entrada	Terra do Sinal de Velocidade	10
5	Entrada	Entrada do PID de Corrente	13
6	Entrada	Sinal de Velocidade do Taco Gerador	Ref. Veloc.
7	Saída	Saída do PID de Velocidade	3
8	Entrada	Sinal de Corrente do Motor	Ref. Corrente
9	Saída	Sinal de Velocidade para o PID	11
10	Entrada	Terra do Sinal de Corrente	Fonte
11	Entrada	Entrada do PID de Velocidade	9
12	Entrada	Alimentação de -12Vcc	Fonte
13	Saída	Circuito de Ganho de Corrente	5
14	Entrada	Alimentação de + 12Vcc	Fonte

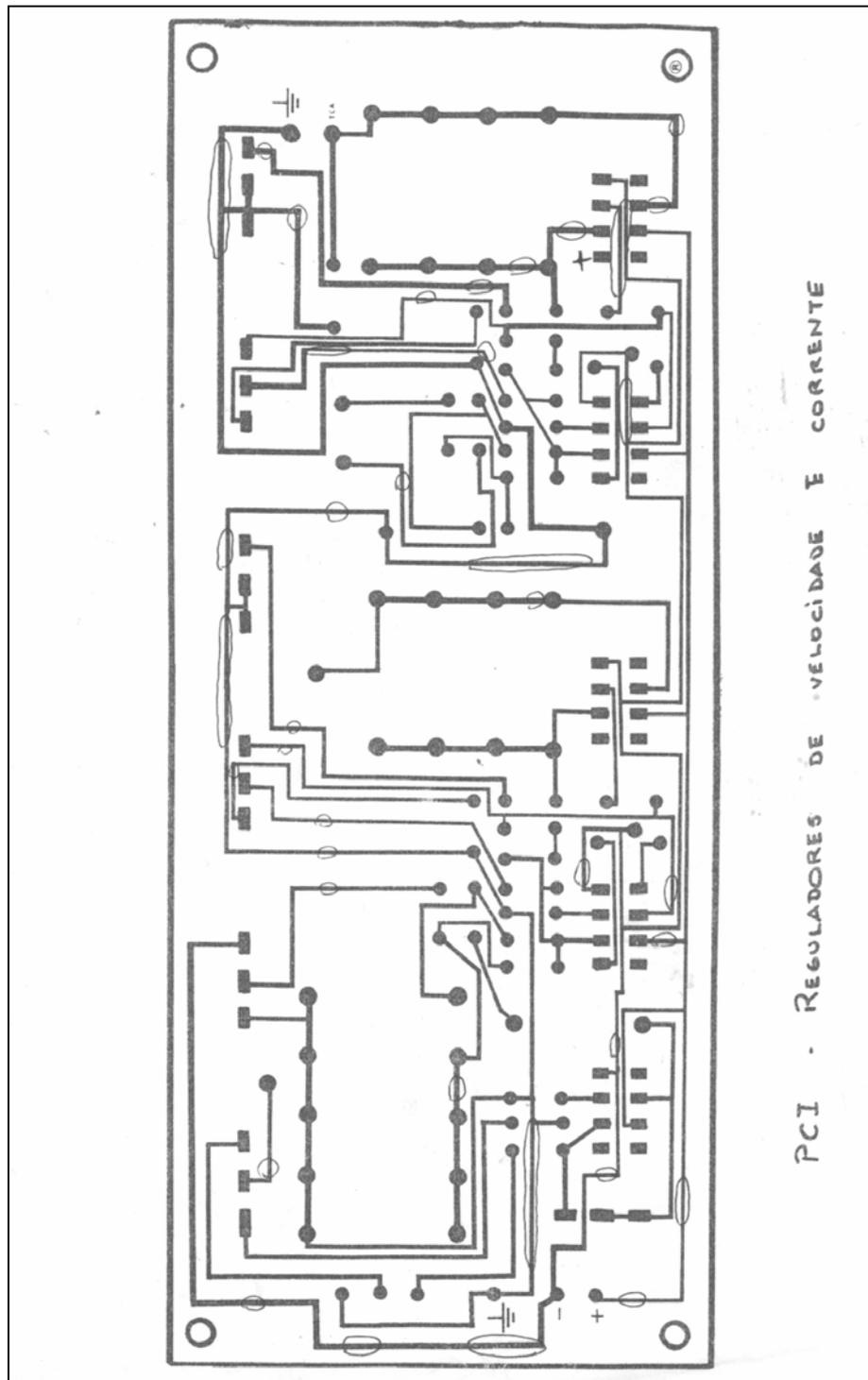
A figura Anexo 1.3 mostra o esquema elétrico do regulador automático de velocidade.



**Fig. Anexo 1.3:** Esquema Elétrico do Regulador de Velocidade.

A figura Anexo 1.4 mostra o lay out da placa de circuitos impresso do regulador de velocidades.





**Fig. Anexo 1.5:** Mascara do Circuito Impresso.

Na tabela Anexo 1.2 está a lista de componentes do regulador automático de velocidade.

Tab. Anexo 1.2: Lista de Materiais.

<b>Item</b>	<b>Componente</b>	<b>Descrição</b>	<b>Quantidade</b>
1	Resistor	Valor 100KΩ	05
2	Resistor	3,9KΩ	02
3	Resistor	470Ω	03
4	Resistor	27KΩ	02
5	Resistor	4,7KΩ	01
6	Resistor	22KΩ	02
7	Resistor	15KΩ	02
8	Resistor	10KΩ	01
9	Potenciômetro	10KΩ	01
10	Potenciômetro	4,7KΩ	03
11	Potenciômetro	47KΩ	02
12	Potenciômetro	470KΩ	01
13	Potenciômetro	10KΩ	02
14	Capacitor	2,2μF	14
15	Capacitor	1,22μF	01
16	Capacitor	0,22μF	01
17	CI	Amplificador Operacional	06

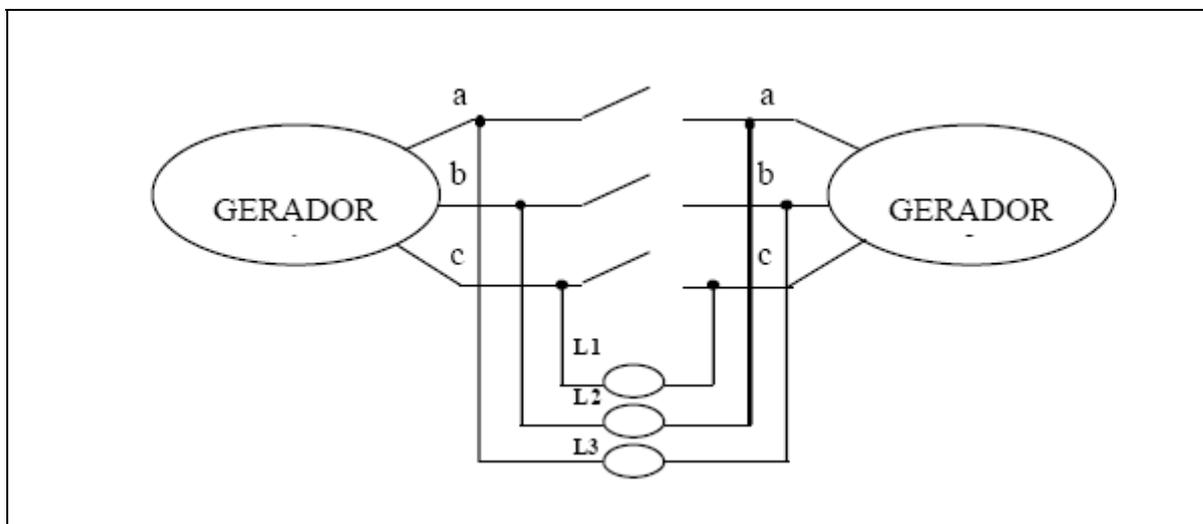
---

## ANEXO 2: TÉCNICA DO FOGO GIRANTE

A técnica do fogo girante é muito simples e foi muito empregada no sincronismo dos geradores. Com o advento da tecnologia essa técnica tem sido menos utilizada e em seu lugar são utilizados sincronoscópios automáticos.

Contudo essa técnica ainda é muito comum para pequenos geradores (geralmente de propriedades rurais) onde o custo de automação deixaria o projeto inviável.

Para realizar o paralelismo a técnica faz uso de três lâmpadas ligadas conforme a Figura 2.1. A ligação dessas lâmpadas é feita a uma chave seccionadora trifásica tripolar.



**Fig. Anexo 2.1:** Ligação do Circuito do Fogo Girante

O método consiste em verificar qual o momento em que as fases de ambos os lados não possuem defasagem. Esse método não permite com clareza determinar valores de tensão em cada uma das fases, por isso são utilizados voltímetros, um em cada lado da chave seccionadora.

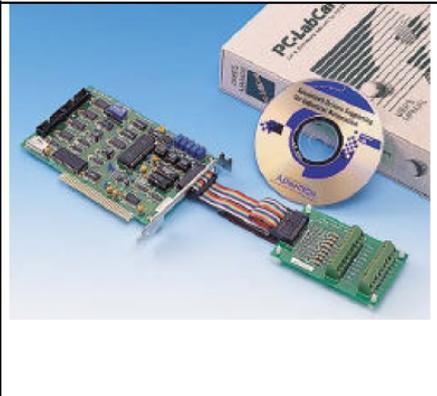
Caso as ligações das fases do sistema não sejam adequadas, não será possível encontrar o ponto de sincronismo, e lâmpadas ficarão cintilantes. Nesse caso é necessário inverter a seqüência de fases de um dos lados da chave trifásica.

Como os geradores têm a freqüência levemente diferente, os fasores das tensões estão levemente deslocados. No momento em que a lâmpada L1 apagar e L2 e L3 estiverem acessas com a metade da intensidade nominal, os fasores estarão em fase e é possível fechar a seccionadora e executar o paralelismo.

## ANEXO 3: INFORMAÇÕES BÁSICAS SOBRE A APLACA DE AQUISIÇÃO.

### PCL-711:

#### Cartão de Aquisição de Dados de Entrada e Saída: Analógico/ Digital

	<p style="text-align: center;"><b>Características</b></p> <ul style="list-style-type: none"> <li>. Oito entradas analógicas simples</li> <li>. Entradas A/D programáveis (várias escalas programáveis)</li> <li>. A/D, D/A separadas</li> <li>. Uma saída analógica</li> <li>. resolução do D/A e A/d com 12 bits</li> <li>. 16 entradas digitais</li> <li>. 16 saídas digitais</li> <li>. Incluindo os drivers de várias em linguagens de programação, as mais populares, as rotinas de calibração, demonstração e programas exemplos</li> <li>. Cartão de conexão e cabo inclusos.</li> </ul>
<p><b>Introdução</b></p> <p>O PCL-711 é um cartão completamente integrado que oferece quatro dos E/S's mais populares para PC/AT e sistemas compatíveis: Conversão de A/D, conversão de D/A, entradas digitais e saídas digitais.</p> <p>O PCL-711 é uma placa de baixo custo com as seguintes características: 8 entradas analógicas de 12 bits, uma saída analógica de 12 bits, 16 entradas digitais e 16 saídas digitais. Além disso, vem com um cartão de interface com cabos apropriados para conexões.</p> <p>O PCL-711 executa uma variedade de trabalhos de I/O com um conjunto de funções de software e uma seleção de outros cartões e acessórios disponíveis.</p>	
<p><b>Especificações</b></p>	
<p><b>Entradas analógicas</b></p> <ul style="list-style-type: none"> <li>. Número de Canais</li> <li>. Conversão de A/D</li> <li>. Faixa de Tensão de trabalho(V)</li> <li>. Modo de Disparo</li> <li>. Transferem de Dados</li> <li>. Precisão</li> <li>. Rejeição de Modo comum</li> <li>. Impedância de Entrada</li> <li>. Tensão Máxima de Entrada</li> </ul>	<ul style="list-style-type: none"> <li>8 canais em modo comum.</li> <li>12 bits, 25 <math>\mu</math>s de tempo de conversão</li> <li><math>\pm 5, \pm 2.5, \pm 1.25, \pm 0.625, \pm 0.3125</math></li> <li>Externo ou interno</li> <li>Controle por programa ou interrupções (IRQ's 2 - 7)</li> <li><math>\pm 2</math> LSB</li> <li>60 dB típico</li> <li><math>&gt; 10 M\Omega</math></li> <li><math>\pm 30 V_{DC}</math> max</li> </ul>

<p><b>Saídas Analógicas</b></p> <ul style="list-style-type: none"> <li>. Faixa do D/A</li> <li>. Tempo atualização</li> </ul> <p><b>Entrada Digital</b></p> <ul style="list-style-type: none"> <li>. Canal</li> </ul> <p><b>Saída digital</b></p> <ul style="list-style-type: none"> <li>. Canais</li> <li>. Nível Lógico 0</li> <li>. Nível Lógico 1</li> </ul> <p><b>Geral</b></p> <ul style="list-style-type: none"> <li>. Consumo de Energia</li> </ul>	<p>uma saída de 12 bits</p> <p>0 ~ 5 V ou 0 ~ 10 V</p> <p>30 <math>\mu</math>sec.</p> <p>16, nível TTL</p> <p>16</p> <p>0.5 V max @ 8 mA (consumo)</p> <p>2.4 V min @ 0.4 mA (fornece)</p> <p>+5V @ 500 mA típico, 1.0 A máx</p> <p>+12V @ 50 mA típico, 100 mA máx</p> <p>-12V @ 14 mA típico, 20 mA máx</p>
---	--

### SELEÇÃO DO ENDEREÇO DE BASE

Posição da chave de endereço de entrada/saída (E/S)						
Faixa (hex) **	1	2	3	4	5	6
	A9	A8	A7	A6	A5	A4
200-20F	1	0	0	0	0	0
210-21F	1	0	0	0	0	1
220-22F*	1	0	0	0	1	0
230-23F	1	0	0	1	0	0
300-30F	1	0	1	0	0	0
3F0-3FF	1	1	1	1	1	1

Nota: Fechado = 0 (ON), Aberto = 1 (OFF)  
A4...A9 corresponde às linhas de endereçamento do barramento do PC  
\* significa configuração de fábrica  
\*\* BASE

## Endereços de Acionamento dos Periféricos da Placa de Aquisição

Localização			Leitura	Escrita
BASE	+	0	Contador 0	Contador 0
BASE	+	1	Contador 0	Contador 0
BASE	+	2	Contador 0	Contador 0
BASE	+	3	N/U ***	Contador 0
BASE	+	4	Byte LSB do A/D	Byte LSB D/A
BASE	+	5	Byte MSB do A/D	Byte MSB D/A
BASE	+	6	Byte LSB do D/I	N/U
BASE	+	7	Byte MSB do D/I	N/U
BASE	+	8	N/U	Limpeza da Requisição de Interrupção
BASE	+	9	N/U	Controle de Ganho
BASE	+	10	N/U	Controle do Multiplexador
BASE	+	11	N/U	Modo de Controle
BASE	+	12	N/U	Disparo do A/D por Software
BASE	+	13	N/U	Byte LSB do D/O
BASE	+	14	N/U	Byte MSB do D/O
BASE	+	15	N/U	N/U
*** Não Usado				

**ANEXO 4: PROGRAMA DE IMPLEMENTAÇÃO DO AVR TRADICIONAL EM C**

```
// //
// PROGRAMA PARA IMPLEMENTAÇÃO DE AVR FUZZY EM TEMPO REAL //
// //
// Esse programa foi baseado no código elaborado por Carlos A. M. Pinheiro //
// e utilizado em Pinheiro (2000). //
// //
// As adaptações para implementação do PSS Fuzzy foram realizadas por //
// Alexandre E. Feo e utilizado em Alexandre (2004). //
// //
// A adequação do programa para executar as funções de regulador de tensão //
// baseada em um controlador tradicional com características de avanço e //
// atraso de fase foi executada por Carlos Eduardo dos Santos em 24/06/2006 //
// //
// //
// //
// Sumário: //
// //
// Programa Principal: Geração do Sinal de Controle Vc. //
// Passo 1: Descrição das Bibliotecas //
// Passo 2: Definição dos Vetores //
// Passo 3: Declaração das Variáveis Globais //
// Passo 4: Início do Programa Principal //
// Passo 5: Declaração das Variáveis Locais //
// Passo 6: Inicialização da Placa de Aquisição //
// Passo 7: Definição dos Parâmetros do AVR Tradicional //
// Passo 8: Definição do Tempo de Amostragem //
// Passo 9: Interface com o Usuário //
// Passo 10: Condições Iniciais //
// Passo 11: Preparação do Timer do Computador //
```

---

```
// Passo 12: Leitura do Canal "1" do A/D (Velocidade) //
// Passo 13: Inicialização do Contador da Placa //
// Passo 14: Leitura do Canal "0" do A/D (Tensão) //
// Passo 15: Sinal de Erro //
// Passo 16: Filtro Passa-Baixa //
// Passo 17: Filtro Passa-Alta //
// Passo 18: Limites do A/D //
// Passo 19: Linearização do Disparo //
// Passo 20: Tratamento dos Dados //
// Passo 21: Escrever no D/A //
// Passo 22: Armazenamento dos Dados //
// Passo 23: Ajuste do Timer do Computador //
// Passo 24: Rotina de Gravação //
// Passo 25: Mensagem Final ao Usuário //
// //
// //
// Sub-Rotinas: //
// Passo 26: Sub-Rotina para Gravação em Disco //
// Passo 27: Verificação do arquivo //
// Passo 28: Mensagem ao usuário. //
// Passo 29: Gravação dos dados. //
// //
// Passo 30: Sub-Rotina para Leitura do A/D //
// Passo 31: Habilita placa de conversões A/D. //
// Passo 32: Adquire dados. //
// Passo 33: Retorna valor em formato decimal. //
// //
// //////////////////////////////////////// //
// _____INÍCIO DO PROGRAMA PRINCIPAL_____ //
```

```
// Passo 1: Bibliotecas do compilador.
```

```
#include <STDLIB.H>
```

---

```

#include <STDIO.H>
#include <CONIO.H>
#include <MATH.H>
#include <dos.h>

// Passo 2: Definição do tamanho do vetor de dados.
#define DIMENSAO_VETOR 30000

// Passo 3: Declaração das variáveis globais.
float Vt; // Tensão terminal. //
unsigned char dadoLSB, dadoMSB; // Variáveis auxiliares. //
unsigned int dado, EndBase; // Variáveis auxiliares. //

// Declaração das sub-rotinas:
// 1) Gravação dos dados do ensaio.
void Grava_disco(char *nome_arq, int n_amost, int i_tempo, int *dados_int);

// 2) Leitura do A/D da placa de aquisição de dados.
float vin(int ch);

// Passo 4: Início da rotina principal.
void main()
{

// Passo 5: Declaração das variáveis locais.
char nome_arquivo[9]; //Caminho e nome do arquivo para gravação de dados. //
unsigned char x0;
unsigned int dadoAD;
int dados[DIMENSAO_VETOR], num_amstras, delta_tempo, i, j, k, ch, x3;

```

---

```

register int n;
float x1, x2, Ji;
float erro, Ref, Vc, a0, a0w, uw1, a1, b1, b1w, dt, u, uw, w2, uw2, deltaW,
        u1, u2, u2w, u2wi, u2i, u1p, u1wp, w1, uw1p, a1w, uvp;
float      Vp, Vi, Vt, Vt1, Vf, ge, gv, wc;
float e, ew, v, vw, T, Ta;

// Passa 6: Inicialização da placa de aquisição conforme dados do fabricante.
// Para maiores detalhes ver anexo (Anexo 3).
EndBase = 0x220;      // Endereço base da placa de aquisição de dados. //
outportb((EndBase+11),1); // Habilita conversão por software do A/D. //
outportb((EndBase+10),0); // Seleção do canal 0 - pino 1 do conector. //
outportb((EndBase+9),0); // Ganho unitário na entrada do A/D. //
dado = 4095.0 * 4.5/10.0;      // Valor inicial do D/A. //
dadoMSB = dado >> 8;      // Deslocar oito vezes para a direita. //
dadoMSB = dadoMSB & 15;    // Pegar apenas os primeiros 4 bits. //
dadoLSB = dado & 255;     // Pegar os 8 bits da esquerda. //
outportb((EndBase+4),dadoLSB); // Escrever nos registros do D/A. //
outportb((EndBase+5),dadoMSB); // Escrever nos registros do D/A. //

// Passo 7: Parâmetros do AVR Tradicional.
a0 = 77.46;
a1 = 6.7864;
b1 = 0.1612;

// Passo 8: Definição do tempo de amostragem / varredura.
Ta = 0.001;
dt = Ta;

// Passo 9: Interface com o usuário
clrscr();
printf("\n\n");

```

---

```

printf("Entre com o caminho e nome do arquivo para gravação dos dados:");
printf("\n\n");
gets(nome_arquivo); // Nome do arquivo onde serão gravados os dados. //
printf("\n");
printf("Número de amostras = ");
scanf("%d", &num_amostras);
if (num_amostras > 15000)
{
    printf("Valor maior que o permitido, ser adotado 15000 amostras.");
    num_amostras = 15000;
}
printf("\n");
printf("Referência [0 a 5] = "); // O valor 2.5 equivale a tensão nominal //
scanf("%f", &Ref); // Valor de referência do sistema de controle //
printf("\n");
printf("\n");
printf("\n");
printf("Ensaio em Andamento");
printf("\n");
printf("\n");

// Passo 10: Definição das condições iniciais.
wc = 2 * 3.14 * 40;
u1 = 0.0; // Parâmetro de saída do filtro passa-baixa do AVR Tradicional. //
u2i = 0.0; // Parâmetro de saída do filtro passa-alta do AVR Tradicional. //
n = 0; // Variável Auxiliar. //
j = 0; // Variável Auxiliar. //
for (n = 0; n < DIMENSAO_VETOR; n++)
    dados[n]=0; // Vetor de armazenamento de dados. //
n = 0;

// Passo 11: Preparação do timer do computador.
x0 = inportb(0x61); // Lê estado de habilitação do timer do computador. //
x0 = x0 | 1; // Ativa bit de habilitação do timer. //

```

---

```

outportb(0x61,x0); // Habilita timer do pc. //

// Passo 12: Leitura do canal 1 do A/D relativo ao valor da rotação do gerador.
ch = 0;
Vt=vin(ch);
ch = 1;

// Passo 13: Repetição da rotina de controle.
do {
    outportb(0x43,0xB4); // Modo do contador 2 do timer do computador. //
    outportb(0x42,0x00); // Valores iniciais do contador. //
    outportb(0x42,0x00);

// Passo 14: Leitura do canal 0 do A/D ro valor da tensão do gerador //
ch--;
Vt1=vin(ch);
uvp = (Vt1 - Vt) * wc;
Vt = Vt + uvp * dt; // Atualização da saída //

// Passo 15: Geração do sinal de erro.
erro = (Ref - Vt);// + uw); // Erro da malha do AVR. //

// Passo 16: Filtro passa-baixa da entrada do compensador.
u1p = (a0 * erro - u1)/b1;
u1 = u1 + u1p * dt;

// Passo 17: Filtro passa-alta da entrada do compensador.
u2 = (a1 * erro - u2i)/b1;
u2i = u2i + u2 * dt;
u = u1 + u2;

```

---

// Passo 18: Ajuste do sinal para os limites do conversor A/D.

```
if (u < 0.0) u = 0.0;           // Limite inferior = 0.0V. //
if (u > 5.0) u = 5.0;         // Limite superior = 5.0V. //
```

// Passo 19: Função de linearização do disparo.

```
Vc = 10.0 * (1.0 - (acos (u/5.0)/3.1416));
```

// Passo 20: Tratamento dos dados.

```
dado = 4095.0 * Vc/10.0;       // Valor de tensão em decimal inteiro. //
dadoMSB = dado >> 8;         // Deslocar oito vezes para a direita. //
dadoMSB = dadoMSB & 15;      // Pegar apenas os primeiros 4 bits. //
dadoLSB = dado & 255;        // Pegar os 8 bits da esquerda. //
```

// Passo 21: Escrever nos registros do D/A.

```
outportb((EndBase+4),dadoLSB);
outportb((EndBase+5),dadoMSB);
```

// Passo 22: Armazenar dados.

```
dadoAD = 4095.0 * (Vt + 5.0)/10.0; // Converte para formato inteiro. //
dados[n] = dadoAD;                // Armazena o valor da tensão terminal Vt. //
n = n + 1;
dados[n] = 4095.0 * (u/5.0);      // Converte para formato inteiro. //
n = n + 1;
j = j + 1;
```

// Rotina de tempo de amostragem / varredura.

```
do {
    outportb(0x43,0x80);         // Modo do contador 2 do time. //
    x1 = inportb(0x42);          // Leitura LSB do contador do time. //
    x2 = inportb(0x42);          // Leitura MSB do contador do time. //
    Ji = 256 * x2 + x1;          // Converte para 16 bits. //
    if (Ji < 0) T = 65536.0 + Ji;
    else T = Ji;                 // Verifica se positivo. //
    T = 65535.0 - T;
```

---

```

        T = (T/1.19318e6) + 20.0e-6;
        } while (T < Ta);          // Verifica tempo de amostragem. //
    ch++;                          // Vari vel auxiliar. //
    } while ((j < num_amostras) && (!kbhit())); // Rotina de controle. //
    dado = 4095.0 * 4.5/10.0;      // Valor inicial do D/A. //
    dadoMSB = dado >> 8;          // Deslocar oito vezes para a direita. //
    dadoMSB = dadoMSB & 15;       // Pegar apenas os primeiros 4 bits. //
    dadoLSB = dado & 255;        // Pegar os 8 bits da esquerda. //
    outportb((EndBase+4),dadoLSB); // Escrever nos registros do D/A. //
    outportb((EndBase+5),dadoMSB); // Escrever nos registros do D/A. //

// Passo 23: Ajuste do timer do computador.
delta_tempo = 256 * x2 + x1;     // Dado do contador do time. //
x0 = inportb(0x61);             // Estado de habilitação do timer do computador. //
x0 = x0 & 0xFE;                 // Desativa bit de habilitação do timer. //
outportb(0x61,x0);             // Inibição do timer do computador. //

// Passo 24: Rotina de gravação dos dados para um arquivo.
num_amostras = j;              // Dados do ensaio. //
Grava_disco(nome_arquivo, num_amostras, delta_tempo, dados);

// Passo 25: Mensagem final ao usuário.
printf("\n"); printf("\n");
printf("FINAL DO ENSAIO");
printf("\n"); printf("\n");
getch();

}

// _____ FIM DO PROGRAMA PRINCIPAL _____ //

```

---

```

// _____ INÍCIO DAS SUB-ROTINAS _____ //

// Passo 26: 1) Sub-rotina para gravação dos dados de ensaio em disco.
void Grava_disco(char *nome_arq, int n_amostr, int i_tempo, int *dados_int)
{
    // Declaração das variáveis.
    FILE *fp;
    register int i,j;
    int dados;

    // Passo 27: Verificação da abertura do arquivo.
    if ((fp = fopen (nome_arq,"w")) == NULL)
    {
        printf(" Não foi possível abrir o arquivo\n");
        return;
    }

    // Passo 28: Registro dos parâmetros e dados do ensaio.
    fprintf(fp,"%d\n",n_amostr);        // Gravação do número de amostras. //
    fprintf(fp,"%d\n",i_tempo);        // Gravação do tempo de amostragem. //
    i=0;
    j=0;
    do {                                // Passo 29: Gravação dos dados do ensaio. //
        dados = dados_int[i];
        fprintf(fp,"%d\n",dados);
        i = i + 1;
        dados = dados_int[i];
        fprintf(fp,"%d\n",dados);
        i = i + 1;
        dados = dados_int[i];
        fprintf(fp,"%d\n",dados);
        i = i + 1;
    }
}

```

---

```

    dados = dados_int[i];
    fprintf(fp,"%d\n",dados);
    i = i + 1;
    j = j + 1;
} while (j < n_amost);

fclose(fp);          // Encerra a gravação do arquivo de dados. //
}

// Passo 30: 2) Sub-rotina para leitura do A/D da placa de dados.
float vin(int ch)
{
    // Passo 31: Habilitação do conversor A/D.
    outportb((EndBase+12),ch);

    // Passo 32: Aquisição dos dados.
    do {
        dadoMSB = inportb(EndBase+5); // Leitura do MSB do A/D. //
        dado = dadoMSB & 0x10;        // Mascara do bit de conversão. //
    } while (dado != 0);             // Teste de final de conversão. //
    dadoLSB = inportb(EndBase+4);    // Leitura do LSB do A/D. //
    dado = (dadoMSB << 8) + dadoLSB; // Dado em formato binário. //
    Vt = 10.0 * dado/4095.0 - 5.0;   // Dado em formato decimal. //

    // Passo 33: Retorna o valor dos dados em decimal.
    return(Vt);
}

}

// _____FIM DAS SUB-ROTINAS_____ //

```

---

## ANEXO 5: FUNÇÃO “READER01”

Para restaurar os arquivo gerados pelo programa desenvolvido em “C” foi utilizada a rotina do MatLab descrita abaixo:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   Programa para Recuperação das Curvas de Resposta do Controlador   %
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

arq=input('Nome do Arquivo de Dados [entre aspas] = ');

```

```

fp = fopen(arq,'r'); % Abrir arquivo de dados %
v = fscanf(fp,'%f'); % Carrega o arquivo para a variável %
fclose(fp);        % Fecha o arquivo %

```

```

n=v(1);

```

```

it=v(2);

```

```

i=3;

```

```

j=1;

```

```

while i<2*n+2, % rotina de organização dos dados %
    Ad(j)=v(i);
    Da(j)=v(i+1);

```

---

```

    j=j+1;
    i=i+2;
end

ref=input('ref=');           % Rotina par trabalhar os dados %

ct1=10.0/4095;
ct2=5.0;
ct3=5.0/4095;

Ti=it;
if it<0
    Ti=65536+it;
end

dt=(65535-Ti)/1.1918e6;
l1f=0;
tt=0;
for i=1:n,
    t(i)=tt;
    tt=tt+dt;
    Vg=(ct1*Ad(i)-ct2);
    TN(i)=Vg;
    Vc=(ct1*Da(i)-ct2);
    Du(i)=Vc;
End

figure;           % Rotina para impressão na tela %
plot(t,(TN/ref));
xlabel('t[s]');ylabel('y(t)');grid;
figure;
plot(t,(Du/ref),'r');
xlabel('t[s]');ylabel('y(t)');grid;

```

---

```
figure;  
plot(t,(TN));  
grid;  
hold;  
plot(t,(Du),'r');  
xlabel('t[s]');ylabel('y(t)');grid;
```

---

## ANEXO 6: FUNÇÃO “READER02”

Durante o trabalho foi necessário comparar algumas curvas de forma simultânea, e assim foi gerado o código abaixo em MatLab para facilitar essa operação.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   Programa para Comparação das Curvas de Resposta do Controlador   %
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all; % Prepara a tela %

arq=input('Nome do Arquivo de Dados [entre aspas] = ');

fp = fopen(arq,'r'); % Abrir o arquivo %
v = fscanf(fp,'%f'); % Carregar o conteúdo dos arquivos para a variável %
fclose(fp); % Fechar o arquivo %

n=v(1); % Definição %
it=v(2);

i=3;
j=1;
wc=2*pi*0.0001;

ref=input('ref='); % Rotina par trabalhar os dados %

ct1=10.0/4095;

```

---

```
ct2=5.0;
ct3=5.0/4095;

Ti=it;
if it<0
    Ti=65536+it;
end

dt=(65535-Ti)/1.1918e6;
l1f=0;
tt=0;
v(2*n+3)= v(2*n+1);
while i<2*n+2,
    Ad(j)=v(i)+((v(i+2)-v(i))*wc*dt);
    j=j+1;
    i=i+2;
end

for i=1:n,
    t(i)=tt;
    tt=tt+dt;
    TN(i)=(ct1*Ad(i)-ct2);
    % Vc=ct3*Da(i);
    % Du(i)=Vc;
end

figure % Rotina para impressão na tela %

plot(t,(TN/ref));

xlabel('Tempo[s]');ylabel('Tensão Terminal[pu]');grid;
```

---

```
figure;
```

```
plot(t,TN);
```

```
xlabel('Tempo[s]');ylabel('Tensão Terminal[pu]');grid;
```

**ANEXO 7: PROGRAMA DE IMPLEMENTAÇÃO DO AVR E PSS FUZZY.**

```
// //
// PROGRAMA PARA IMPLEMENTAÇÃO DE AVR FUZZY EM TEMPO REAL //
// //
// Esse programa foi baseado no código elaborado por Carlos A. M. Pinheiro //
// e utilizado em Pinheiro (2000). //
// //
// As adaptações para implementação do PSS Fuzzy foram realizadas por //
// Alexandre E. Feo e utilizado em Alexandre (2004). //
// //
// A adequação do programa para executar as funções de regulador de tensão //
// baseada em um controlador Fuzzy com características de avanço e atraso //
// de fase foram executadas por Carlos Eduardo dos Santos em 24/06/2006. //
// //
// //
// //
// //
// Sumário: //
// //
// Programa Principal: Geração do Sinal de Controle Vc. //
// Passo 1: Descrição das Bibliotecas //
// Passo 2: Definição dos Vetores //
// Passo 3: Declaração das Variáveis Globais //
// Passo 4: Início do Programa Principal //
// Passo 5: Declaração das Variáveis Locais //
// Passo 6: Inicialização da Placa de Aquisição //
// Passo 7: Definição dos Parâmetros do PSS Fuzzy //
// Passo 8: Definição dos Parâmetros do AVR Fuzzy //
// //
// Passo 9: Definição do Tempo de Amostragem //
// Passo 10: Interface com o Usuário //
```

---

// Passo 11: Condições Iniciais //

// Passo 12: Preparação do Timer do Computador //

// Passo 13: Leitura do Canal "1" do A/D (Velocidade) //

// Passo 14: Inicialização do Contador da Placa //

// Passo 15: Leitura do Canal "0" do A/D (Tensão) //

// Passo 16: Implementação do *PSS Fuzzy* //

// a) Filtro Passa-Baixa do *PSS Fuzzy* //

// b) Filtro Passa-Alta do *PSS Fuzzy* //

// c) Sinal de Entrada do Fuzificador //

// d) Parâmetros da Função de Pertinência //

// e) Sinal Estabilizante //

// Passo 17: Sinal de Erro //

// Passo 18: Implementação do *AVR Fuzzy* //

// a) Filtro Passa-Baixa do *AVR Fuzzy* //

// b) Filtro Passa-Alta do *AVR Fuzzy* //

// c) Sinal de Entrada do Fuzificador //

// d) Parâmetros da Função de Pertinência //

// e) Sinal de Controle //

// Passo 19: Limites do A/D //

// Passo 20: Linearização do Disparo //

// Passo 21: Tratamento dos Dados //

// Passo 22: Escrever no D/A //

// Passo 23: Armazenamento e Tratamento dos Dados //

// Passo 24: Rotina de Amostragem / Varredura //

// Passo 25: Ajuste do Timer do Computador //

// Passo 26: Rotina de Gravação //

// Passo 27: Mensagem Final ao Usuário //

// //

// Sub-Rotinas: //

// Passo 28: 1) Sub-Rotina para Gravação em Disco //

// a) Declaração das Variáveis //

// b) Verificação da Abertura do Arquivo //

// c) Parâmetros do Ensaio //

// //

---

```

// Passo 29: Sub-Rotina para Leitura do A/D //
//   a) Habilitação do Conversor A/D //
//   b) Aquisição de Dados //
// //
// Passo 30: Compensador Fuzzy //
//   a) Declaração das Variáveis //
//   b) Fuzificador //
//   c) Implicações Lógicas //
//   d) Defuzificador //
//   e) Retorno do Sinal de Controle //
// //
// //////////////////////////////////////////////////////////////////// //

```

```

// _____INÍCIO DO PROGRAMA PRINCIPAL_____ //

```

```

// Passo 1: Bibliotecas do compilador.

```

```

#include <STDLIB.H>

```

```

#include <STDIO.H>

```

```

#include <CONIO.H>

```

```

#include <MATH.H>

```

```

#include <dos.h>

```

```

// Passo 2: Definição do tamanho do vetor de dados.

```

```

#define DIMENSAO_VETOR 30000

```

```

// Passo 3: Declaração das variáveis globais.

```

```

float Vt; // Tensão terminal. //

```

```

unsigned char dadoLSB, dadoMSB; // Variáveis auxiliares. //

```

```

unsigned int dado, EndBase; // Variáveis auxiliares. //

```

---

```
// Declaração das sub-rotinas:
// 1) Gravação dos dados do ensaio.
void Grava_disco(char *nome_arq, int n_amost, int i_tempo, int *dados_int);

// 2) Leitura do A/D da placa de aquisição de dados.
float vin(int ch);

// 3) Aplicação do compensador Fuzzy.
float ver(float e, float v, float ceNL, float ceNM,
          float ceZE, float cePM, float cePL, float beNL, float beNM,
          float beZE, float bePM, float bePL, float cvNL, float cvNM,
          float cvZE, float cvPM, float cvPL, float bvNL, float bvNM,
          float bvZE, float bvPM, float bvPL, float caNL, float caNM,
          float caZE, float caPM, float caPL, float ueNL, float ueNM,
          float ueZE, float uePM, float uePL, float uvNL, float uvNM,
          float uvZE, float uvPM, float uvPL);

// _____ INÍCIO DO PROGRAMA PRINCIPAL _____ //

// Passo 4: Início da rotina principal.
void main()
{

// Passo 5: Declaração das variáveis locais.
char nome_arquivo[9]; //Caminho e nome do arquivo para gravação de dados. //
```

---

```

unsigned char x0;
unsigned int dadoAD;
int dados[DIMENSAO_VETOR], num_amostras, delta_tempo, i, j, k, ch, x3;
register int n;
float x1, x2, Ji;
float erro, Ref, Vc, a0, a0w, uw1, a1, b1, b1w, dt, u, uw, w2, uw2, deltaW,
        u1, u2, u2w, u2wi, u2i, u1p, u1wp, w1, uw1p, a1w, uvp, uwp, deltaW1;
float      Vp, Vi, Vt, Vt1, Vf, ge, gv, wc, wc1;
float e, ew, v, vw, T, Ta;
float ceNL, ceNM, ceZE, cePM, cePL,
        beNL, beNM, beZE, bePM, bePL,
        cvNL, cvNM, cvZE, cvPM, cvPL,
        bvNL, bvNM, bvZE, bvPM, bvPL,
        caNL, caNM, caZE, caPM, caPL,
        ueNL, ueNM, ueZE, uePM, uePL,
        uvNL, uvNM, uvZE, uvPM, uvPL;

```

*// Passo 6: Inicialização da placa de aquisição conforme dados do fabricante.*

*// Para maiores detalhes ver Anexo 3.*

```

EndBase = 0x220;      // Endereço base da placa de aquisição de dados. //
outportb((EndBase+11),1); // Habilita conversão por software do A/D. //
outportb((EndBase+10),0); // Seleção do canal 0 - pino 1 do conector. //
outportb((EndBase+9),0); // Ganho unit rio na entrada do A/D. //
dado = 4095.0 * 4.5/10.0; // Valor inicial do D/A (representa 0 Volts) //
dadoMSB = dado >> 8; // Deslocar oito vezes para a direita. //
dadoMSB = dadoMSB & 15; // Pegar apenas os primeiros 4 bits. //
dadoLSB = dado & 255; // Pegar os 8 bits da esquerda. //
outportb((EndBase+4),dadoLSB); // Escrever nos registros do D/A. //
outportb((EndBase+5),dadoMSB); // Escrever nos registros do D/A. //

```

*// Passo 7: Parâmetros do PSS Fuzzy.*

---

```
a0w = 11.82;  
a1w = 1.035;  
b1w = 2.07;
```

```
// Passo 8: Parâmetros do AVR Fuzzy.
```

```
a0 = 77.46;  
a1 = 6.7864;  
b1 = 0.1612;
```

```
// Passo 9: Definição do tempo de amostragem / varredura.
```

```
Ta = 0.001;  
dt = Ta;
```

```
// Passo 10: Interface com o usuário.
```

```
clrscr();  
printf("\n\n");  
printf("Entre com o caminho e nome do arquivo para gravação dos dados:");  
printf("\n\n");  
gets(nome_arquivo); // Nome do arquivo onde serão gravados os dados. //  
printf("\n");  
printf("Número de amostras = ");  
scanf("%d", &num_amostras);  
if (num_amostras > 15000)  
{  
    printf("Valor maior que o permitido, ser adotado 15000 amostras.");  
    num_amostras = 15000;  
}  
printf("\n");  
printf("Referência [0 a 5] = "); // O valor 2.5 equivale a tensão nominal //  
scanf("%f", &Ref); // Valor de referência do sistema de controle //  
printf("\n");
```

---

```

printf("\n");
printf("\n");
printf("Ensaio em Andamento");
printf("\n");
printf("\n");

// Passo 11: Definição das condições iniciais.
wc = 2 * 3.14 * 40;
wc1 = 2 * 3.14 * 0.2;
deltaW1 = 0.0;
u1 = 0.0;    // Parâmetro de saída do filtro passa-baixa do AVR Fuzzy. //
u2i = 0.0;   // Parâmetro de saída do filtro passa-alta do AVR Fuzzy. //
uw1 = 0.0;   // Parâmetro de saída do filtro passa-baixa do PSS Fuzzy. //
u2wi = 0.0;  // Parâmetro de saída do filtro passa-alta do PSS Fuzzy. //
n = 0;       // Variável Auxiliar. //
j = 0;       // Variável Auxiliar. //
for (n = 0; n < DIMENSAO_VETOR; n++)
    dados[n]=0; // Vetor de armazenamento de dados. //
n = 0;

// Passo 12: Preparação do timer do computador.
x0 = inportb(0x61); // Lê estado de habilitação do timer do computador. //
x0 = x0 | 1;       // Ativa bit de habilitação do timer. //
outportb(0x61,x0); // Habilita timer do pc. //

// Passo 13: Leitura do canal 1 do A/D relativo ao valor da rotação do gerador.
ch = 0;
Vt=vin(ch);
ch = 1;
w1=vin(ch);

// Passo 14: Repetição da rotina de controle.
do {

```

---

```

outportb(0x43,0xB4); // Modo do contador 2 do timer do computador. //
outportb(0x42,0x00); // Valores iniciais do contador. //
outportb(0x42,0x00);

// Leitura do canal 1 do A/D relativo a rotação do gerador.
w2=vin(ch); // Segunda leitura do canal A/D1. //
deltaW1 = (w2 - w1); // Calculo da variação de rotação no gerador. //
uwp = (deltaW1 - deltaW) * wc1;
deltaW = uwp * dt;
w1 = w2; // Atualiza w1 para a próxima leitura. //
// Passo 15: Leitura do canal 0 do A/D relativo a tensão do gerador.
ch--;
Vt1=vin(ch);
uwp = (Vt1 - Vt) * wc;
Vt = Vt + uwp * dt;

// Passo 16: Implementação do PSS Fuzzy.
// a) Filtro passa-baixa da entrada do PSS Fuzzy.
uw1p = (a0w * deltaW - uw1)/b1w;
uw1 = uw1 + uw1p * dt;
// b) Filtro passa-alta da entrada do PSS Fuzzy.
u2w = (a1w * deltaW - u2wi)/b1w;
u2wi = u2wi + u2w * dt;
// c) Definição dos sinais de entrada do Fuzificador.
ew = uw1;
vw = u2w;
// d) Parâmetros da função de pertinência.
ceNL=-5.0; ceNM=-2.5; ceZE= 0.0; cePM= 2.5; cePL= 5.0;
beNL= 5.0; beNM= 5.0; beZE= 3.5; bePM= 5.0; bePL= 5.0;
cvNL=-5.0; cvNM=-2.5; cvZE= 0.0; cvPM= 2.5; cvPL= 5.0;
bvNL= 5.0; bvNM= 5.0; bvZE= 5.0; bvPM= 5.0; bvPL= 5.0;
caNL=-5.0; caNM=-2.5; caZE= 0.0; caPM= 2.5; caPL= 5.0;
// e) Definição do sinal estabilizante.
uw=ver(ew, vw, // Sinal estabilizante. //

```

---

```

ceNL, ceNM, ceZE, cePM, cePL,
beNL, beNM, beZE, bePM, bePL,
cvNL, cvNM, cvZE, cvPM, cvPL,
bvNL, bvNM, bvZE, bvPM, bvPL,
caNL, caNM, caZE, caPM, caPL,
ueNL, ueNM, ueZE, uePM, uePL,
uvNL, uvNM, uvZE, uvPM, uvPL);

```

*// Passo 17: Geração do sinal de erro.*

```

erro = (Ref - Vt + uw);           // Erro da malha do AVR. //

```

*// Passo 18: Implementação do AVR Fuzzy.*

*// a) Filtro passa-baixa da entrada do AVR Fuzzy.*

```

u1p = (a0 * erro - u1)/b1;

```

```

u1 = u1 + u1p * dt;

```

*// b) Filtro passa-alta da entrada do AVR Fuzzy.*

```

u2 = (a1 * erro - u2i)/b1;

```

```

u2i = u2i + u2 * dt;

```

*// c) Definição dos sinais de entrada do Fuzificador.*

```

e=u1;

```

```

v=u2;

```

*// d) Parâmetros da função de pertinência.*

```

ceNL=-100.0; ceNM= -50.0; ceZE= 0.0; cePM= 50.0; cePL= 100.0;

```

```

beNL= 50.0; beNM= 50.0; beZE= 120.0; bePM= 50.0; bePL= 50.0;

```

```

cvNL=-100.0; cvNM= -50.0; cvZE= 0.0; cvPM= 50.0; cvPL= 100.0;

```

```

bvNL= 50.0; bvNM= 50.0; bvZE= 50.0; bvPM= 50.0; bvPL= 50.0;

```

```

caNL=-100.0; caNM= -50.0; caZE= 0.0; caPM= 50.0; caPL= 100.0;

```

*// e) Definição do sinal estabilizante.*

```

u=ver(e, v,           // Sinal estabilizante. //

```

```

ceNL, ceNM, ceZE, cePM, cePL,

```

```

beNL, beNM, beZE, bePM, bePL,

```

```

cvNL, cvNM, cvZE, cvPM, cvPL,

```

```

bvNL, bvNM, bvZE, bvPM, bvPL,

```

```

caNL, caNM, caZE, caPM, caPL,

```

---

```

ueNL, ueNM, ueZE, uePM, uePL,
uvNL, uvNM, uvZE, uvPM, uvPL);

```

```

// Passo 19: Ajuste do sinal para os limites do conversor A/D.

```

```

if (u < 0.0) u = 0.0;           // Limite inferior = 0.0V. //
if (u > 5.0) u = 5.0;         // Limite superior = 5.0V. //

```

```

// Passo 20: Função de linearização do disparo.

```

```

Vc = 10.0 * (1.0 - (acos (u/5.0)/3.1416));

```

```

// Passo 21: Tratamento dos dados.

```

```

dado = 4095.0 * Vc/10.0; // Valor de tensão em decimal inteiro. //
dadoMSB = dado >> 8;    // Deslocar oito vezes para a direita. //
dadoMSB = dadoMSB & 15; // Pegar apenas os primeiros 4 bits. //
dadoLSB = dado & 255;   // Pegar os 8 bits da esquerda. //

```

```

// Passo 22: Escrever nos registros do D/A.

```

```

outportb((EndBase+4),dadoLSB);
outportb((EndBase+5),dadoMSB);

```

```

// Passo 23: Armazenamento e tratamento dos dados.

```

```

dadoAD = 4095.0 * (Vt + 5.0)/10.0; // Converte para formato inteiro. //
dados[n] = dadoAD; // Armazena o valor da tensão terminal Vt. //
n = n + 1;
dados[n] = 4095.0 * (u/5.0); // Converte para formato inteiro. //
n = n + 1;
j = j + 1;

```

---

```

// Passo 24: Rotina de tempo de amostragem / varredura.
do {
    outportb(0x43,0x80);    // Modo do contador 2 do time.    //
    x1 = inportb(0x42);    // Leitura LSB do contador do time. //
    x2 = inportb(0x42);    // Leitura MSB do contador do time. //
    Ji = 256 * x2 + x1;    // Converte para 16 bits.        //
    if (Ji < 0) T = 65536.0 + Ji;
    else T = Ji;          // Verifica se positivo.        //
    T = 65535.0 - T;
    T = (T/1.19318e6) + 20.0e-6;
    } while (T < Ta);    // Verifica tempo de amostragem. //
ch++;                    // Variável auxiliar. //
} while ((j < num_amostras) && (!kbhit())); // Rotina de controle. //

// Passo 25: Ajuste do timer do computador.
delta_tempo = 256 * x2 + x1;    // Dado do contador do time.    //
x0 = inportb(0x61);    // Estado de habilitação do timer do computador. //
x0 = x0 & 0xFE;    // Desativa bit de habilitação do timer. //
outportb(0x61,x0);    // Inibição do timer do computador. //

// Passo 26: Rotina de gravação dos dados para um arquivo.
num_amostras = j;    // Dados do ensaio. //
Grava_disco(nome_arquivo, num_amostras, delta_tempo, dados);
dado = 4095.0 * 4.5/10.0;    // Valor inicial do D/A. //
dadoMSB = dado >> 8;    // Deslocar oito vezes para a direita. //
dadoMSB = dadoMSB & 15;    // Pegar apenas os primeiros 4 bits. //
dadoLSB = dado & 255;    // Pegar os 8 bits da esquerda. //
outportb((EndBase+4),dadoLSB);    // Escrever nos registros do D/A. //
outportb((EndBase+5),dadoMSB);    // Escrever nos registros do D/A. //

// Passo 27: Mensagem final ao usuário.
printf("\n"); printf("\n");

```

---

```

printf("FINAL DO ENSAIO");
printf("\n"); printf("\n");
getch();
}
// _____ FIM DO PROGRAMA PRINCIPAL _____ //

// _____ INÍCIO DAS SUB-ROTINAS _____ //

// Passo 28: 1) Sub-rotina para gravação dos dados de ensaio em disco.
void Grava_disco(char *nome_arq, int n_amost, int i_tempo, int *dados_int)
{
    // a) Declaração das variáveis.
    FILE *fp;
    register int i,j;
    int dados;

    // b) Verificação da abertura do arquivo.
    if ((fp = fopen (nome_arq,"w")) == NULL)
        {
            printf(" Não foi possível abrir o arquivo\n");
            return;
        }

    // c) Registro dos parâmetros e dados do ensaio.
    fprintf(fp,"%d\n",n_amost);        // Gravação do número de amostras. //
    fprintf(fp,"%d\n",i_tempo);       // Gravação do tempo de amostragem. //
    i=0;
    j=0;

    do {                               // Gravação dos dados do ensaio. //
        dados = dados_int[i];
        fprintf(fp,"%d\n",dados);
        i = i + 1;
        dados = dados_int[i];
    }
}

```

---

```

    fprintf(fp,"%d\n",dados);
    i = i + 1;
    dados = dados_int[i];
    fprintf(fp,"%d\n",dados);
    i = i + 1;
    dados = dados_int[i];
    fprintf(fp,"%d\n",dados);
    i = i + 1;
    j = j + 1;
} while (j < n_amostr);
fclose(fp);           // Encerra a gravação do arquivo de dados. //
}

// Passo 29: 2) Sub-rotina para leitura do A/D da placa de dados.
float vin(int ch)
{
    // a) Habilitação do conversor A/D.
    outportb((EndBase+12),ch);
    // b) Aquisição dos dados.
    do {
        dadoMSB = inportb(EndBase+5); // Leitura do MSB do A/D. //
        dado = dadoMSB & 0x10; // Mascara do bit de conversão. //
    } while (dado != 0); // Teste de final de conversão. //
    dadoLSB = inportb(EndBase+4); // Leitura do LSB do A/D. //
    dado = (dadoMSB << 8) + dadoLSB; // Dado em formato bin rio. //
    Vt = 10.0 * dado/4095.0 - 5.0; // Dado em formato decimal. //

    // c) Retorna o valor dos dados em decimal.
    return(Vt);
}

// Passo 30: 3) Compensador Fuzzy.
float ver(float e, float v, float ceNL, float ceNM,

```

---

```

float ceZE, float cePM, float cePL, float beNL, float beNM,
float beZE, float bePM, float bePL, float cvNL, float cvNM,
float cvZE, float cvPM, float cvPL, float bvNL, float bvNM,
float bvZE, float bvPM, float bvPL, float caNL, float caNM,
float caZE, float caPM, float caPL, float ueNL, float ueNM,
float ueZE, float uePM, float uePL, float uvNL, float uvNM,
float uvZE, float uvPM, float uvPL)
{

```

```

// a) Declaração das variáveis.

```

```

float Vc, d1;

```

```

float r1, r2, r3, r4, r5, r6, r7, r8, r9, r10, r11, r12, r13, r14, r15,
r16, r17, r18, r19, r20, r21, r22, r23, r24, r25;

```

```

// Fuzificador.

```

```

// Pertinência de "e" na curva NL.

```

```

if (((ceNL - beNL) <= e) && (e <= (ceNL + beNL)))
    ueNL = 1.0 - (fabs (ceNL - e)) / beNL;
else ueNL = 0.0;
    if (e < ceNL)
        ueNL = 1.0;

```

```

// Pertinência de "e" na curva NM.

```

```

if (((ceNM - beNM) <= e) && (e <= (ceNM + beNM)))
    ueNM = 1.0 - (fabs (ceNM - e)) / beNM;
else ueNM = 0.0;

```

```

// Pertinência de "e" na curva ZE.

```

```

if (((ceZE - beZE) <= e) && (e <= (ceZE + beZE)))
    ueZE = 1.0 - (fabs (ceZE - e)) / beZE;
else ueZE = 0.0;

```

---

```
// Pertinência de "e" na curva PM.
if (((cePM - bePM) <= e) && (e <= (cePM + bePM)))
    uePM = 1.0 - (fabs (cePM - e)) / bePM;
else uePM = 0.0;
```

```
// Pertinência de "e" na curva PL.
if (((cePL - bePL) <= e) && (e <= (cePL + bePL)))
    uePL = 1.0 - ( fabs ( cePL - e ) ) / bePL;
else uePL = 0.0;
    if (e > cePL)
        uePL = 1.0;
```

```
// Pertinência de "v" na curva NL.
if (((cvNL - bvNL) <= v) && (v <= (cvNL + bvNL)))
    uvNL = 1.0 - (fabs (cvNL - v)) / bvNL;
else uvNL = 0.0;
    if (v < cvNL)
        uvNL = 1.0;
```

```
// Pertinência de "v" na curva NM.
if (((cvNM - bvNM) <= v) && (v <= (cvNM + bvNM)))
    uvNM = 1.0 - (fabs (cvNM - v)) / bvNM;
else uvNM = 0.0;
```

```
// Pertinência de "v" na curva ZE.
if (((cvZE - bvZE) <= v) && (v <= (cvZE + bvZE)))
    uvZE = 1.0 - (fabs (cvZE - v)) / bvZE;
else uvZE = 0.0;
```

```
// Pertinência de "v" na curva PM.
if (((cvPM - bvPM) <= v) && (v <= (cvPM + bvPM)))
```

---

```
uvPM = 1.0 - (fabs (cvPM - v)) / bvPM;  
else uvPM = 0.0;
```

```
// Pertinência de "v" na curva PL.
```

```
if (((cvPL - bvPL) <= v) && (v <= (cvPL + bvPL)))  
    uvPL = 1.0 - (fabs (cvPL - v)) / bvPL;  
else uvPL = 0.0;  
    if (v > cvPL)  
        uvPL = 1.0;
```

```
// c) Implicações lógicas.
```

```
r1 = ueNL * uvNL;  
r2 = ueNL * uvNM;  
r3 = ueNL * uvZE;  
r4 = ueNL * uvPM;  
r5 = ueNL * uvPL;  
r6 = ueNM * uvNL;  
r7 = ueNM * uvNM;  
r8 = ueNM * uvZE;  
r9 = ueNM * uvPM;  
r10 = ueNM * uvPL;  
r11 = ueZE * uvNL;  
r12 = ueZE * uvNM;  
r13 = ueZE * uvZE;  
r14 = ueZE * uvPM;  
r15 = ueZE * uvPL;  
r16 = uePM * uvNL;
```

---

```

r17 = uePM * uvNM;
r18 = uePM * uvZE;
r19 = uePM * uvPM;
r20 = uePM * uvPL;
r21 = uePL * uvNL;
r22 = uePL * uvNM;
r23 = uePL * uvZE;
r24 = uePL * uvPM;
r25 = uePL * uvPL;

```

```
// d) Defuzificador.
```

```

d1 = ((caNL * r1) + (caNL * r2) + (caNL * r3) + (caNM * r4) +
      (caZE * r5) + (caNL * r6) + (caNL * r7) + (caNM * r8) +
      (caZE * r9) + (caPM * r10) + (caNL * r11) + (caNM * r12) +
      (caZE * r13) + (caPM * r14) + (caPL * r15) + (caNM * r16) +
      (caZE * r17) + (caPM * r18) + (caPL * r19) + (caPL * r20) +
      (caZE * r21) + (caPM * r22) + (caPL * r23) + (caPL * r24) +
      (caPL * r25));
Vc = d1/(r1 + r2 + r3 + r4 + r5 + r6 + r7 + r8 + r9 + r10 +
        r11 + r12 + r13 + r14 + r15 + r16 + r17 + r18 + r19 + r20 +
        r21 + r22 + r23 + r24 + r25);

```

```
// e) Retorna o valor do sinal de controle Vc.
```

```

return(Vc);
}

```

```
// _____FIM DAS SUB-ROTINAS_____ //
```

---

## ANEXO 8: MODELAGEM DOS FILTROS PASSA- ALTA E BAIXA

A modelagem dos filtros passa alta e passa baixa que foi aplicada para a criação do programa em tempo real do controlador *Fuzzy* está descrita abaixo.

A função de transferência de um compensador de avanço e atraso de fase pode ser vista abaixo:

$$\frac{u(s)}{e(s)} = \frac{a_1s + a_0}{b_1s + 1} \quad (\text{Anexo8.1})$$

Trabalhando a equação podemos separa a função passa baixas e a função pasas altas desse filtro, conforme mostra a equação (Anexo8.2)

$$\frac{u(s)}{e(s)} = \frac{a_1s}{b_1s + 1} + \frac{a_0}{b_1s + 1} \quad (\text{Anexo8.2})$$

Para facilitar a análise foram feitas as seguintes definições:

$$\frac{u_2(s)}{e(s)} = \frac{a_1s}{b_1s + 1} \quad (\text{Anexo8.3})$$

e

$$\frac{u_1(s)}{e(s)} = \frac{a_0}{b_1s + 1} \quad (\text{Anexo8.4})$$

sendo que:

$$u(s) = u_1(s) + u_2(s) \quad (\text{Anexo8.5})$$

Trabalhando a equação (Anexo8.3) vem:

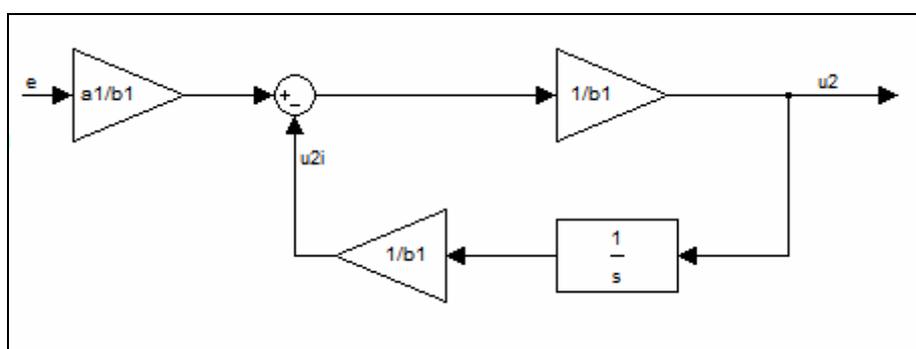
---

$$(b_1 s + 1).u_2(s) = a_1 s.e(s)$$

$$\left(b_1 + \frac{1}{s}\right).u_2(s) = a_1.e(s)$$

$$\left(1 + \frac{1}{b_1} \cdot \frac{1}{s}\right).u_2(s) = \frac{a_1}{b_1}.e(s) \quad (\text{Anexo8.6})$$

Com a equação (Anexo8.6) pode-se montar o diagrama de blocos correspondente à função  $u_2(s)$ , como mostra a Figura 8.1:



**Fig. Anexo 8.1:** Diagrama de Blocos do Filtro Passa Altas.

Do diagrama de blocos pode-se determinar os valores de  $u_2$  e  $u_{2i}$  pelas equações (Anexo8.7) e (Anexo8.8).

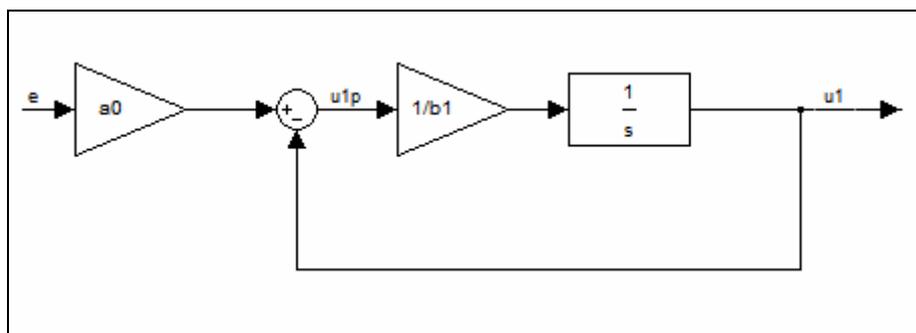
$$u_2(s) = [a_1 s.e(s) - u_{2i}(s)] / b_1 \quad (\text{Anexo8.7})$$

$$u_{2i}(s) = u_{2i}(s) + u_2(s).dt \quad (\text{Anexo8.8})$$

De forma análoga ao filtro passa-altas da equação (Anexo8.3), pode-se trabalhar a equação do filtro passa-baixas (Anexo8.4), resultando na Figura 8.2.

$$(b_1 s + 1) \cdot u_1(s) = a_0 \cdot e(s)$$

$$\left(1 + \frac{1}{b_1} \cdot \frac{1}{s}\right) \cdot u_1(s) = \frac{a_0}{b_1} \cdot \frac{1}{s} \cdot e(s) \quad (\text{Anexo8.9})$$



**Fig. Anexo 8.2:** Diagrama de Blocos do Filtro Passa Baixas.

Onde conclui-se que  $u_1$  e  $u_{1p}$  são dados respectivamente por:

$$u_{1p}(s) = [a_0 \cdot e(s) - u_1(s)] / b_1 \quad (\text{Anexo8.10})$$

$$u_1(s) = u_1 + u_{1p}(s) \cdot dt \quad (\text{Anexo8.11})$$

onde:

$a_0=1$  e  $b_1=1/w_c$  e  $W_c=2\text{Pi} \cdot 40$

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)