

UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI
Programa de Pós-Graduação em Automação e Sistemas Elétricos Industriais



*Unifei - Universidade Federal de
Itajubá*



*Grupo de Estudos em Manutenção
Eletro-Eletrônica e Instalações*

**“ANÁLISE COMPARATIVA DE
DESEMPENHO DE REGULADORES DIGITAIS
EM ACIONAMENTO DE MOTOR DE
CORRENTE CONTÍNUA EXCITAÇÃO SÉRIE”**

Dissertação submetida à
Universidade Federal de Itajubá
como parte dos requisitos necessários
para a obtenção do Título de Mestre
em Engenharia Elétrica

Tony Youssif Teixeira Darido

Orientador: Prof. Dr. Ângelo José Junqueira Rezek

UNIFEI - Universidade Federal de Itajubá

Itajubá, MG - Abril de 2004

Ficha Catalográfica elaborada pela biblioteca Mauá
Bibliotecária Cristiane N. C. Carpinteiro – CRB6 / 1702

D 218 a

Darido, Tony Youssif Teixeira.

Análise comparativa de desempenho de Reguladores Digitais em
acionamento de motor de corrente contínua excitação série / por Tony
Youssif Teixeira Darido. – Itajubá, MG : [s.n.], 2004.
76p. il.

Orientador: Prof. Dr. Ângelo José Junqueira Rezek
Dissertação (mestrado) – Universidade Federal de Itajubá.

1. Lógica fuzzy 2. Acionamento CC série 3. Aquisição de dados
4. Comparação de desempenho. I. Rezek, Ângelo José Junqueira, orient.
II. Universidade Federal de Itajubá. III. Título

CDU 621.313.222 : 62-55

DEDICATÓRIA

Dedico este trabalho a todos que me incentivaram e ajudaram de alguma forma, direta ou indiretamente.

Especialmente aos meus familiares e professores, onde gostaria de registrar o amigo e professor Antonio Tadeu Lyrio de Almeida.

AGRADECIMENTOS

O autor deste trabalho agradece

A Deus que permitiu a concretização deste grande passo.

Ao professor Ângelo J. J. Rezek pelo grande e importante auxílio no desenvolvimento desta dissertação, pois sua orientação foi imprescindível à conclusão da mesma.

A FAPEMIG-MG, pela aprovação do projeto de pesquisa “ Implementação e Avaliação de Reguladores Digitais em acionamentos Elétricos “, processo TEC2917-98, cuja concessão possibilitou a realização deste trabalho de pesquisa.

Ao professor Carlos A. M. Pinheiro pelo importante auxílio na implementação dos reguladores Fuzzy.

LISTA DE SIMBOLOS

U_{ca}	tensão de saída da ponte.
R_a	resistência total (armadura + alisamento).
I_a	corrente de armadura.
E_a	força contra eletromotriz.
n	velocidade.
C	conjugado ou torque.
U	tensão de alimentação.
L_a	indutância total (armadura + alisamento).
L_e	indutância excitação série.
C_c	conjugado resistente ou torque de carga.
J	momento de inércia (motor + carga).
w	velocidade angular.
I_n	corrente de armadura nominal.
T_a	constante de tempo da armadura.
U_n	tensão de alimentação nominal.
C_n	conjugado nominal do motor.
n_n	velocidade nominal.
T_H	constante de tempo de aceleração.
V_i	fator multiplicador da corrente nominal.
V_{ff}	tensão fase-fase.
V_{fn}	tensão fase-neutro.
P_n	potência nominal.

LISTA DE FIGURAS

Figura 2.1 – Diagrama Básico de Acionamento e Controle de um Motor.....	5
Figura 2.2 – Conversor Unidirecional Controlado (Simbologia).....	6
Figura 2.3 – Operação em Mono-quadrante.....	6
Figura 2.4 – Operação em Quatro Quadrantes.	7
Figura 3.1 - Representação da Parte Mecânica e do Circuito Elétrico da Armadura do Motor CC com Excitação Série.	8
Figura 3.2 – Diagrama de Blocos da Parte Mecânica do Motor CC.....	11
Figura 3.3 – Diagrama de Blocos do Circuito da Armadura da Máquina CC.....	12
Figura 3.4 – Diagrama de Blocos Completo da Máquina CC.	13
Figura 3.5 – Acionamento em Mono-quadrante de uma Máquina CC Através de uma Ponte Conversora Graetz (totalmente controlada).	13
Figura 3.6 – Diagrama de Blocos do Controle de Velocidade em Malha Fechada de uma Máquina CC.....	16
Figura 3.7 – Malha de Regulação de Corrente.....	18
Figura 3.8 – Malha de Regulação de Velocidade.	20
Figura 4.1 – Ponte Conversora Graetz (totalmente controlada).	24
Figura 4.2 – Acionamento em Mono-quadrante Através de uma Ponte Conversora Graetz.....	25
Figura 4.3 – Formas de Ondas Típicas do TCA 780.....	26
Figura 4.4 – Circuito de Disparo Tipo Rampa.	26
Figura 5.1 – Diagrama Esquemático de Controle Digital em Malha Fechada de uma Máquina CC.	27
Figura 5.2 – Fluxograma do Controle Digital.....	30
Figura 5.3 - Comparação entre um controlador fuzzy e um controlador convencional.	32
Figura 5.4 - Representação das funções de pertinência típicas para uma variável fuzzy.	33
Figura 5.5 – Representação das funções de pertinência adotadas para a 34	34
Figura 6.1 – Partida com Carga e Degrau Negativo e Positivo de Carga 38	38
Figura 6.2 – Degrau Negativo e Positivo na Referência 38	38
Figura 6.3 – Partida com Carga e Degrau Negativo e Positivo de Carga 39	39
Figura 6.4 – Degrau Negativo e Positivo na Referência 39	39
Figura 6.5 – Partida com Carga e Degrau Negativo e Positivo de Carga 40	40
Figura 6.6 – Degrau Negativo e Positivo na Referência 40	40
Figura 6.7 – Partida com Carga e Degrau Negativo e Positivo de Carga 41	41
Figura 6.8 – Degrau Negativo e Positivo na Referência 41	41
Figura 6.9 – Partida com Carga e Degrau Negativo e Positivo de Carga 43	43

Figura 6.10 – Degrau Negativo e Positivo na Referência	43
Figura 6.11 – Partida com Carga e Degrau Negativo e Positivo de Carga	44
Figura 6.12 – Degrau Negativo e Positivo na Referência	44

LISTA DE TABELAS

Tabela 3.1 – Dados do motor CC.	17
Tabela 3.2 – Escolha do tipo de controlador e método adequado de otimização para o sistema.	21
Tabela 3.3 – Ajuste dos parâmetros dos reguladores resultantes da aplicação dos métodos da otimização em função da linearidade, do módulo e da simetria.	22
Tabela 5.1 - Regras base para o controle fuzzy de corrente.	35
Tabela 5.2 – Relação entre os parâmetros dos reguladores.....	36

ÍNDICE

RESUMO.....	1
ABSTRACT	2
1. INTRODUÇÃO:	3
2. ACIONAMENTO E CONTROLE DE MÁQUINAS DE CORRENTE CONTÍNUA:	5
3. EQUACIONAMENTO COMPLETO DO ACIONAMENTO DE UM MÁQUINA DE CORRENTE CONTÍNUA:.....	8
3.1. EQUACIONAMENTO E DIAGRAMA DE BLOCOS DO MOTOR:.....	8
3.2. ESCOLHA E AJUSTE DOS REGULADORES :	16
4. IMPLEMENTAÇÃO E MONTAGEM DO SISTEMA DE ACIONAMENTO :	24
4.1. ACIONAMENTO EM MONO-QUADRANTE ATRAVÉS DE UMA PONTE CONVERSORA GRAETZ :	24
4.2. CIRCUITO DE DISPARO :	26
5. O CONTROLE DIGITAL DOS ACIONAMENTOS :	27
5.1. AQUISIÇÃO E CONVERSÃO DE DADOS :	27
5.2. ESCOLHA DOS REGULADORES :	28
5.2.1. REGULADOR PI :	28
5.2.2. REGULADOR FUZZY :	31
5.2.2.1. CONTROLE NEBULOSO (<i>FUZZY CONTROL</i>) :	31
5.2.2.2. PROJETO DO CONTROLADOR FUZZY :	33
6. RESULTADOS OBTIDOS :	37
6.1. RESULTADOS EXPERIMENTAIS :	37
6.2. OUTROS RESULTADOS OBTIDOS :	42
7. CONCLUSÕES :	45
REFERÊNCIAS BIBLIOGRÁFICAS :	46
ANEXO A – CIRCUITO ELETRÔNICO PARA O COMANDO DE DISPARO DE UMA PONTE TRIFÁSICA TIRISTORIZADA	48
ANEXO B :	61
PROGRAMA PARA O ACIONAMENTO CONTROLADO DE UMA MÁQUINA CC UTILIZANDO REGULADORES PI (PROPORCIONAL-INTEGRAL) :	61
PROGRAMA PARA O ACIONAMENTO CONTROLADO DE UMA MÁQUINA CC UTILIZANDO REGULADORES COM LÓGICA FUZZY (PI-FUZZY):	67

RESUMO

Os sistemas digitais de acionamentos elétricos são atualmente muito utilizados na indústria, substituindo aos poucos os sistemas analógicos existentes. Devido ao grande desenvolvimento da eletrônica e dos microprocessadores pode-se desenvolver novas técnicas de controle digital, como por exemplo, a adaptativa e a vetorial. Mais recentemente, com o aparecimento de técnicas de inteligência artificial, uma nova tendência vem se observando no projeto de controladores ainda mais otimizados.

Esta possibilidade de melhoria pode ser conseguida com o projeto e implementação de reguladores de corrente e velocidade empregando a lógica fuzzy.

Pretende-se nesse trabalho demonstrar os resultados de uma análise comparativa entre reguladores digitais do tipo proporcional e integral (PI) com reguladores empregando lógica fuzzy (PI-Fuzzy). Para isso, foi utilizado em laboratório um sistema de controle para o acionamento de um motor de corrente contínua em mono-quadrante, realizado através de uma ponte conversora Graetz totalmente controlada, sendo o sistema implementado, utilizando-se um microcomputador PC, uma placa de aquisição de dados, um circuito eletrônico de disparo e sensores de corrente e velocidade.

Nesse microcomputador, foram utilizados dois diferentes softwares, em linguagem C++ de acordo com a lógica analisada, sendo esse software responsável pela aquisição dos dados de entrada, controle das malhas de velocidade e corrente, e envio de um sinal de controle pelo qual pode-se controlar o circuito de disparo.

Sendo enfim apresentados os resultados experimentais obtidos em laboratório com relação ao comportamento e resposta do sistema diante de determinadas condições de operação para ambas as lógicas de controle.

ABSTRACT

Nowadays the digital system of electric drives has been largely used in the industry, taking place of the analogical systems. Due to the great development of electronics and microprocessors it is possible the development new digital control systems, e.g., the adaptive and vectorial. Recently, as artificial intelligence technics grow, a new trend starts to appear in the project of more optimized controllers.

This possibility of improvement can be reached with project and implementation of current and speed regulators using fuzzy logic.

The aim of this work is to demonstrate the results of a comparative analysis between proportional and integral digital regulators against other ones using fuzzy logic. For this, a control system for the DC-driving of a c.c motor was made in laboratory, with a fully controlled Graetz bridge, a PC, a data input board, a firing electronic circuit and speed and current sensors.

In this microcomputer, two different softwares were used, in the C++ language, according to the analyzed logic, being it responsible for the input data acquisition, control of speed and current loops and output of a control signal which controls the firing circuit.

Finally, the experimental results observed in laboratory, related with the response and behavior of the system due to specifically conditions of operation for both control logics, are presented and discussed.

1. INTRODUÇÃO:

Desde sua invenção, as máquinas elétricas (de corrente contínua, de indução, ou síncronas) têm sido muito utilizadas nos processos industriais, onde passaram a substituir o homem, principalmente nos serviços mais difíceis e pesados. Com o desenvolvimento desses processos de fabricação e o aumento do número de máquinas utilizadas tornou-se necessário a utilização de sistemas de controle e automação, à princípio, apenas eletro mecânicos e posteriormente eletrônicos analógico-digitais.

Apesar de mais caras, as máquinas de corrente contínua são muito utilizadas nos processos onde se exige um controle de velocidade apurado, como por exemplo, nos processos de bobinamento nas indústrias de papel e celulose, e na laminação nas indústrias siderúrgicas. São também muito empregadas no acionamento de veículos de tração elétrica, como trens e metrô, e em componentes automotivos, como vidros elétricos e limpadores de pára-brisa.

Assim como para outros tipos de máquinas, o acionamento e controle das máquinas de corrente contínua pode ser analógico ou digital. Devido a sua grande simplicidade e flexibilidade na implementação de sistemas de controle dinâmico, os dispositivos digitais têm sido muito utilizados na indústria, substituindo os circuitos de controle analógico, cujas configurações são rígidas e pré-definidas.

Essa flexibilidade associada às vantagens de menores custos e dimensões, permite uma elevação no número de aplicações utilizando microprocessadores e computadores digitais em sistemas de controle.

Uma das principais alternativas desse tipo de controle é a utilização de um microcomputador PC para aquisição de dados. Tal procedimento vem sendo muito utilizado, principalmente no controle de malha fechada, nas áreas de monitoração e supervisão industrial. Isso é possível devido ao aumento da capacidade de processamento, ao baixo custo dos equipamentos e grande flexibilidade oferecida pelos softwares de controle utilizados, possibilitando uma eficiente interface com o sistema através de placas de aquisição de dados ou até mesmo através de controladores lógicos programáveis (CLPs). Algumas das vantagens da utilização dessa tecnologia e desses tipos de interfaces são: o aumento do número de máquinas controladas ou monitoradas num sistema e a possibilidade de se conseguir o controle em tempo real, podendo-se dispensar a presença constante de um operador para a alteração de velocidade e

implementação de ajustes necessários diante de alterações na carga da máquina.

2. ACIONAMENTO E CONTROLE DE MÁQUINAS DE CORRENTE CONTÍNUA:

Um sistema de acionamento e controle de um motor tem como objetivo controlar um ou mais parâmetros dessa máquina, como por exemplo: a velocidade, a posição angular, a aceleração ou conjugado do eixo. Como as condições mecânicas de saída do motor dependem das condições elétricas de entrada é de vital importância para qualquer tipo de controle a análise da relação entre essas condições, ou seja, a análise das características peculiares de cada máquina e sistema de acionamento.

O diagrama abaixo ilustra o esquema básico de acionamento e controle de um motor, incluindo a carga e a fonte de alimentação.

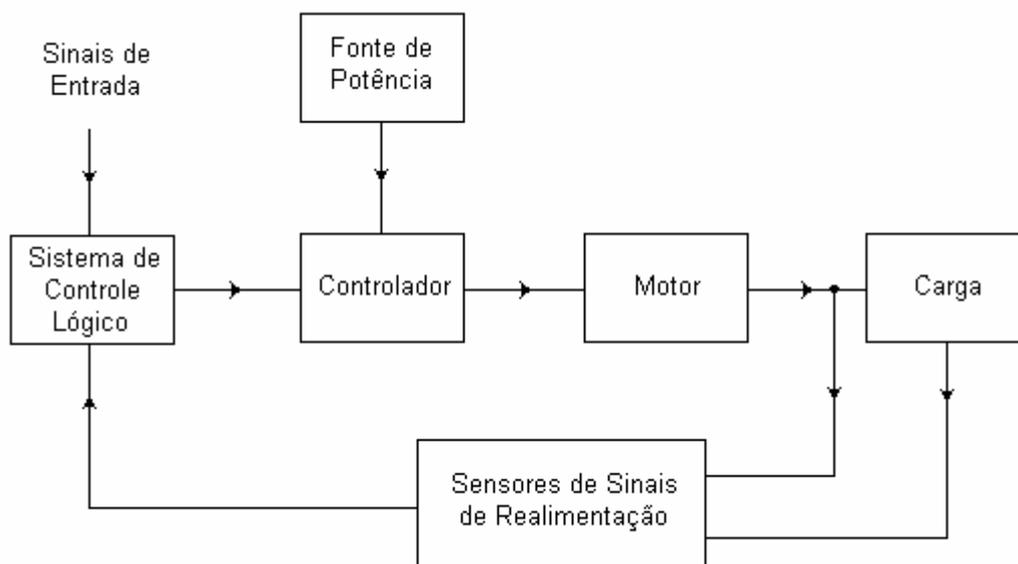


Figura 2.1 – Diagrama Básico de Acionamento e Controle de um Motor.

O acionamento e controle de uma máquina de corrente contínua é em geral muito simples, realizado através de diodos e tiristores (SCR's). São utilizados retificadores controlados pela fase (phase-controlled) para fontes de potência de corrente alternada, enquanto para fontes de corrente contínua são utilizados circuitos pulsadores (choppers).

Esses acionamentos podem ser unidirecionais (em mono-quadrante) ou bidirecionais (em multi-quadrantes).

- **Acionamento em Mono-quadrante:**

Nos acionamentos em 1º quadrante é comum a utilização de retificadores semi ou totalmente controlados. Nesses casos, a velocidade do motor é variada através da tensão aplicada à armadura do motor, sendo essa tensão variada através do ângulo de disparo (α) dos tiristores comandados pelos reguladores de corrente e velocidade, representados nas figuras 2.2 e 2.3 abaixo:

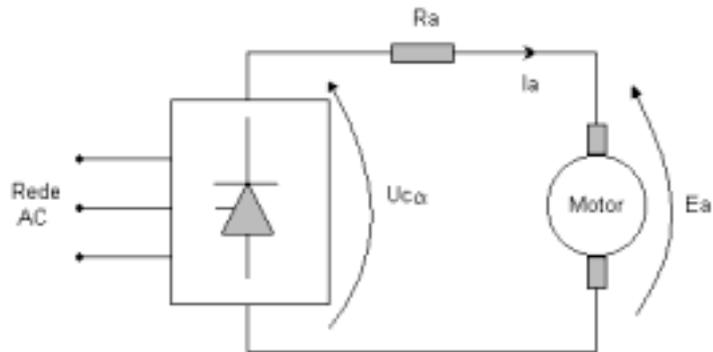


Figura 2.2 – Conversor Unidirecional Controlado (Simbologia).

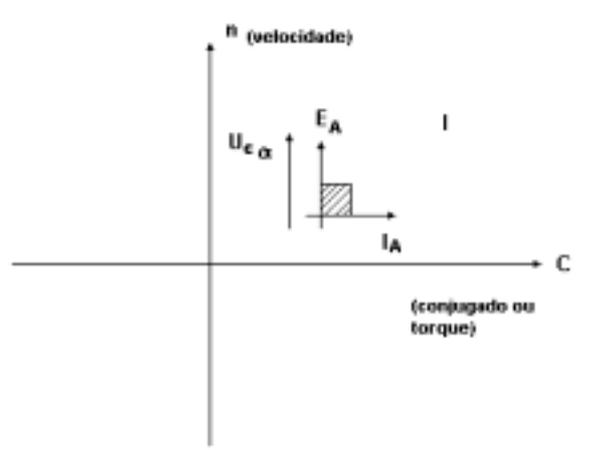


Figura 2.3 – Operação em Mono-quadrante.

- **Acionamento em Multi-quadrantes:**

Para os acionamentos em multi-quadrantes pode-se utilizar contadores (para a comutação mecânica do circuito da armadura ou da excitação); conversores anti-paralelos (para comutação eletrônica do circuito da armadura, com ou sem corrente de circulação); ou choppers (conversores CC-CC) para a operação em quatro quadrantes.

Nesse tipo de acionamento, temos as seguintes etapas :

No **1º quadrante**, a máquina de CC opera como um motor tracionando a carga.

No **2º quadrante**, ocorre uma inversão do sentido de atuação do conjugado, a máquina mantém o sentido de rotação, porém passa a operar como um gerador. Ocorrendo assim a frenagem regenerativa.

No **3º quadrante**, após a anular-se a velocidade, o conjugado mantém o sentido de rotação inverso ao do 1º quadrante.

No **4º quadrante**, ocorre novamente uma inversão do sentido do conjugado do motor, quando a máquina mantém o sentido de rotação e passa a operar como gerador. Ocorrendo mais uma vez a frenagem regenerativa, o que será mostrado na figura 2.4.

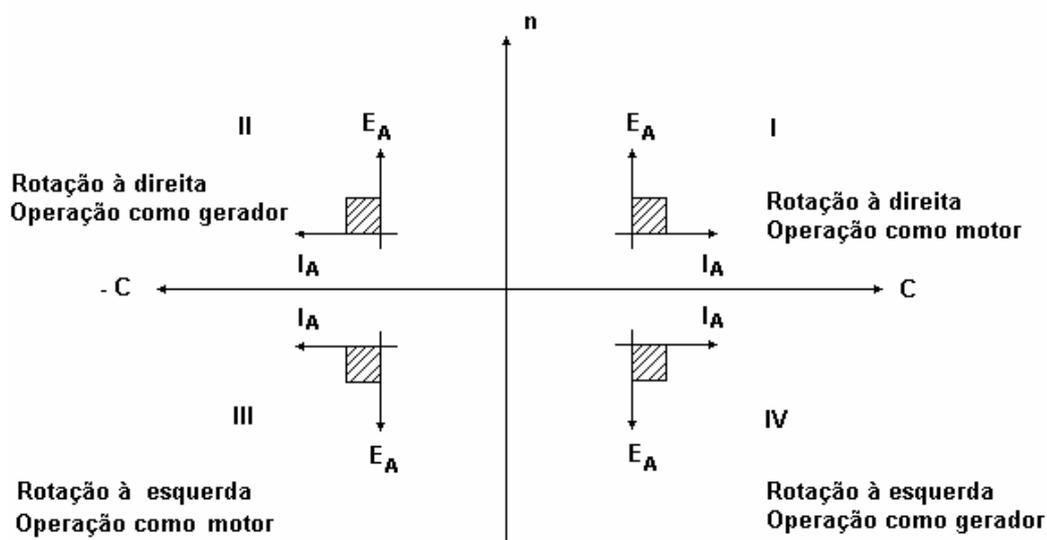


Figura 2.4 – Operação em Quatro Quadrantes.

- Para esse trabalho, foram analisados em laboratório apenas acionamentos em mono-quadrantes, no entanto, os mesmos são válidos para acionamentos em multi-quadrantes, onde os reguladores digitais utilizados são os mesmos.

3. EQUACIONAMENTO COMPLETO DO ACIONAMENTO DE UM MÁQUINA DE CORRENTE CONTÍNUA:

3.1. EQUACIONAMENTO E DIAGRAMA DE BLOCOS DO MOTOR:

A representação do circuito de armadura do motor de corrente contínua com excitação série, bem como da parte mecânica, está mostrada na figura 3.1.

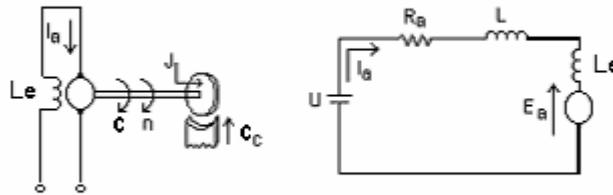


Figura 3.1 - Representação da Parte Mecânica e do Circuito Elétrico da Armadura do Motor CC com Excitação Série.

Na figura 3.1 tem-se que:

- Ea - Força contra eletromotriz.
- U - Tensão de alimentação.
- Ra - Resistência total (armadura e indutor de alisamento).
- La - Indutância total (armadura e indutor de alisamento).
- Le – Indutância de excitação série.
- Ia - Corrente de armadura.
- C - Conjugado do motor.
- Cc - Torque de carga ou conjugado resistente.
- J - Momento de inércia (motor + carga).
- n - Velocidade (rpm).
- ω - Velocidade angular (rad/s).

Onde o conjugado do motor CC é dado por:

$$C = K_m \phi I_a \quad (3.1)$$

sendo, K_m constante.

A equação da parte mecânica (motor + carga), desprezando o atrito viscoso de rotação, é:

$$C - C_c = J \cdot \frac{d\omega}{dt} \quad (3.2)$$

Aplicando transformada de Laplace na equação (3.2), tem-se:

$$C - C_c = Js\omega \quad (3.3)$$

A velocidade angular ω pode ser expressa como:

$$\omega = \frac{2\pi}{60} \cdot n \quad (3.4)$$

Substituindo a equação (3.4) na equação (3.3), resulta:

$$\frac{2\pi}{60} \cdot n = \frac{1}{Js} \cdot (C - C_c) \quad (3.5)$$

Logo,

$$\frac{2\pi}{60} \cdot \frac{n}{n_n} \cdot n_n = \frac{1}{Js} \cdot \left(\frac{C - C_c}{C_N} \right) \cdot C_N \quad (3.6)$$

$$\frac{n}{n_n} = \frac{1}{\frac{2\pi}{60} \cdot \frac{Jn_n s}{C_N}} \cdot \left(\frac{C - C_c}{C_N} \right) \quad (3.7)$$

onde,

C_N – Conjugado do motor nominal.

n_N – Velocidade nominal.

Define-se a constante de tempo de aceleração T_H como sendo:

$$T_H = \frac{2\pi}{60} \cdot \frac{Jn_n}{C_N} \quad (3.8)$$

Então,

$$\frac{n}{n_n} = \frac{1}{sT_H} \cdot \left(\frac{C}{C_N} - \frac{C_c}{C_N} \right) \quad (3.9)$$

Definindo em valores “pu” as grandezas conjugado do motor, conjugado de carga, rotação e corrente de armadura, tem-se:

$$\frac{C}{C_N} = c \quad (\text{pu})$$

$$\frac{C_c}{C_N} = c_c \quad (\text{pu})$$

$$\frac{n}{n_n} = n_u \quad (\text{pu})$$

$$\frac{I_a}{I_N} = i_a \quad (\text{pu})$$

onde,

I_N - Corrente de armadura nominal.

Logo,

$$n_u = \frac{1}{sT_H} \cdot (c - c_c) \quad (3.10)$$

A constante de tempo de aceleração pode ser interpretada como sendo o tempo necessário para o motor atingir a velocidade nominal partindo do repouso, quando o mesmo é acelerado por intermédio de um conjugado resultante igual ao conjugado nominal do motor.

A figura 3.2 ilustra o diagrama de blocos da parte mecânica do motor CC.

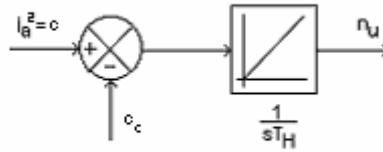


Figura 3.2 – Diagrama de Blocos da Parte Mecânica do Motor CC.

Para o circuito de armadura do motor CC, tem-se:

$$U = R_a I_a + L_a \cdot \frac{dI_a}{dt} + Ea \quad (3.11)$$

Aplicando transformada de Laplace na equação (3.11):

$$I_a = \frac{U - Ea}{R_a + sL_a} \quad (3.12)$$

A constante de tempo da armadura T_a é definida como:

$$T_a = \frac{L_a}{R_a} \quad (3.13)$$

Então,

$$\frac{I_a}{I_N} = \frac{U - Ea}{U_N} \cdot \frac{U_N}{R_a I_N} \cdot \frac{1}{1 + sT_a} \quad (3.14)$$

Define-se em valores “pu” as tensões U e Ea:

$$\frac{U}{U_N} = u \quad (\text{pu})$$

$$\frac{Ea}{U_N} = e \quad (\text{pu})$$

onde,

U_N – Tensão de alimentação nominal.

Logo,

$$i_a = \frac{v_i}{1 + sT_a} \cdot (u - e) \quad (3.15)$$

onde,
$$v_i = \frac{U_N}{R_a I_N}$$

A constante v_i pode ser interpretada como o fator multiplicador da corrente nominal, para a obtenção da corrente com rotor bloqueado, quando tensão nominal é aplicada à armadura.

A figura 3.3 mostra o diagrama de blocos das grandezas elétricas da armadura.

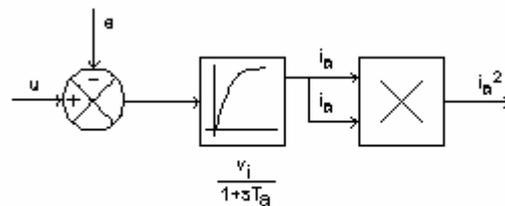


Figura 3.3 – Diagrama de Blocos do Circuito da Armadura da Máquina CC.

A determinação da constante de tempo T_a pode ser feita medindo-se a resistência R_a e a indutância total ($L_a + L_e$) do circuito da armadura. Deve ser levado em consideração a indutância de alisamento incluída em série com o circuito da armadura e a indutância de excitação.

Para um fluxo constante, a tensão induzida é diretamente proporcional à velocidade angular. Portanto, para condição de fluxo nominal, $n_u = e$ (pu).

O diagrama de blocos completo da máquina CC, incluindo o circuito da armadura e a parte mecânica, é mostrado na figura 3.4.

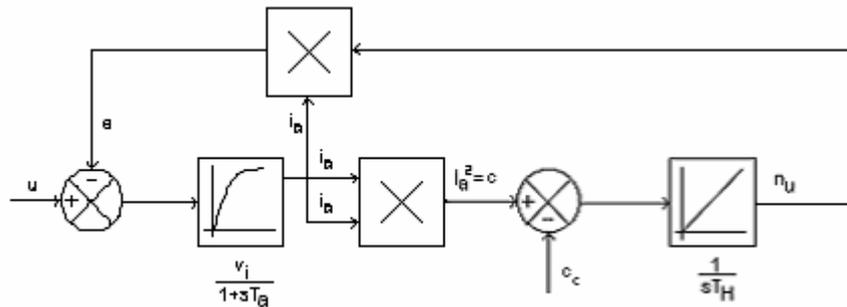


Figura 3.4 – Diagrama de Blocos Completo da Máquina CC.

Para um acionamento em mono-quadrante utilizando uma ponte conversora Graetz totalmente controlada tem-se a seguinte montagem:

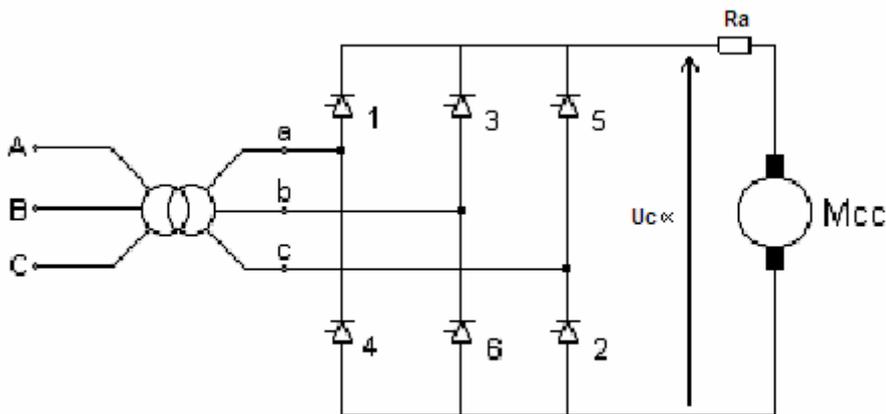


Figura 3.5 – Acionamento em Mono-quadrante de uma Máquina CC Através de uma Ponte Conversora Graetz (totalmente controlada).

O ganho dessa ponte conversora pode ser obtido a partir da equação abaixo:

$$U_{c\alpha} = \frac{1}{(2\pi/6)} \int_{\pi/3}^{2\pi/3} \sqrt{2} \cdot V_{ff} \cdot \text{sen} \omega t \cdot d\omega t$$

$$U_{c\alpha} = \frac{3\sqrt{2}}{\pi} \cdot V_{ff} \cdot \cos \alpha$$

$$U_{c\alpha} = 1,35 \cdot (V_{ff} \cdot \cos \alpha)$$

$$\frac{dU}{d\alpha} = -1,35 \cdot V_{ff} \cdot \operatorname{sen} \alpha$$

Multiplicando membro a membro por (π / U_N) , vem :

$$\frac{d(U / U_N)}{d(\alpha / \pi)} = -1,35 \cdot \frac{V_{ff}}{U_N} \cdot \pi \cdot \operatorname{sen} \alpha$$

O valor de $U_{c\alpha}$ foi determinado de modo que para a tensão nominal na armadura do motor (220 V dc) o valor do ângulo de disparo α é de 30°. Como $U_{c\alpha} = 1,35 \cdot V_{ff} \cdot \cos \alpha$, deste modo $U_{c\alpha} = 188$ [V].

Tem-se :

$$\frac{dU}{d\alpha u} = 1,35 \cdot \frac{188}{220} \cdot \pi \cdot \operatorname{sen} \alpha \quad \rightarrow \quad \frac{dU}{d\alpha u} = -1,15 \cdot \pi \cdot \operatorname{sen} \alpha$$

Sendo $\alpha u = \alpha / \pi$ o valor em (pu) do ângulo de disparo α .

$$V_s = \left| \frac{dU}{d\alpha u} \right| = 1,15 \cdot \pi \cdot \operatorname{sen} \alpha$$

Para $\alpha = 90^\circ$, tem-se o máximo ganho :

$$V_s = \left| \frac{dU}{d\alpha u} \right|_{\alpha=90^\circ} = 1,15 \cdot \pi \cdot 1$$

Para $\alpha = 30^\circ$, tem-se o ganho dado por :

$$V_s = \left. \frac{dU}{d\alpha} \right|_{\alpha=30^\circ} = 1,15 \cdot \pi \cdot 0,5$$

O ganho médio V_s será :

$$V_s = 1,15 \cdot \pi \cdot 0,75 = 2,71 \quad (3.16)$$

O ganho do sistema de regulação de corrente nesse caso será com otimização simétrica:

$$V_{sia} = V_s v_i \quad (3.17)$$

Nesse tipo de acionamento, o controle da corrente de armadura e da velocidade é feito em malha fechada, e usando transdutores para realimentação dos valores instantâneos reais.

O transdutor de corrente é constituído de um sensor Hall, que relaciona a corrente de armadura com uma tensão CC proporcional. A utilização de um filtro do sinal de corrente é recomendável, devido à ondulação.

O transdutor de velocidade é constituído de um taco-gerador acoplado ao eixo do motor CC. A inclusão de um filtro é recomendada, devido à ondulação da tensão de saída do taco gerador.

A malha de controle de velocidade e limitação de corrente completa, incluindo os reguladores de velocidade e corrente, transdutores e filtros, é mostrada na figura 3.6.

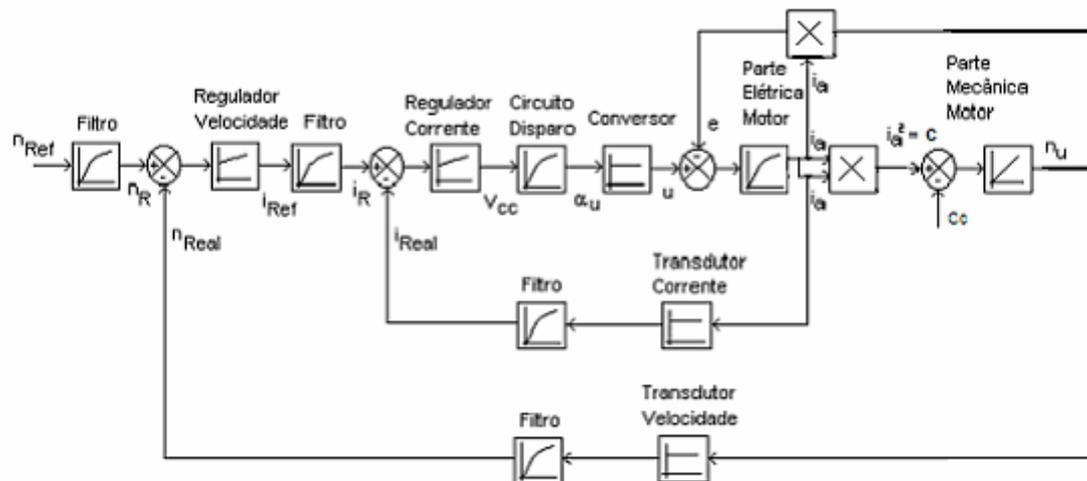


Figura 3.6 – Diagrama de Blocos do Controle de Velocidade em Malha Fechada de uma Máquina CC.

3.2. ESCOLHA E AJUSTE DOS REGULADORES :

A estratégia usada para a escolha e ajuste dos parâmetros dos reguladores é baseada no método de otimização simétrica. A otimização da malha de controle é baseada na relação existente entre as diversas constantes de tempo que aparecem no sistema, fornecendo os dados do motor e as equações do sistema pode-se então determinar as características do controlador mais apropriado.

Os dados do motor de corrente contínua com excitação série utilizado são apresentados na tabela 3.1.

Tabela 3.1 – Dados do motor CC.

Fabricante	Equacional
Tensão de alimentação	$U_N = 220$ [V]
Potência nominal	$P_N = 1,7$ [KW]
Velocidade nominal	$n_N = 1500$ [rpm]
Corrente armadura nominal	$I_N = 7,72$ [A]
Resistência total (armadura e indutor de alisamento)	$R_a = 5,5$ [Ω]
Indutância total (armadura e indutor de alisamento)	$L_a = 444$ [mH]

O conjugado do motor nominal é obtido por:

$$M_N = \frac{P_N}{\frac{2\pi}{60} \cdot n_N} = \frac{1700}{\frac{2\pi}{60} \cdot 1500} = 10,82 \text{ [N.m]} \quad (3.18)$$

O valor do momento de inércia J é $0,07$ [Kg.m²]. A constante de tempo T_H é calculada pela equação (3.8). Então,

$$T_H = 1,20 \text{ [s]}$$

A constante de tempo T_a é obtida pela equação (3.13). Logo,

$$T_a = 80,72 \text{ [ms]}$$

A constante v_i é calculada por:

$$v_i = \frac{U_N}{R_a I_N} = \frac{220}{5,5 \cdot 7,72} = 5,18 \quad (3.19)$$

Levando-se em consideração que em uma ponte de Graetz totalmente controlada, podemos ter um retardo máximo no chaveamento de 1/6 do ciclo, adotamos um valor de 1/12 do ciclo.

Então,

$$T_{ss} = 1,5 \text{ [ms]}$$

O transdutor de corrente utilizado está constituído de um sensor Hall. É incluído um filtro, e a constante de tempo do filtro adotada é:

$$T_{gi} = 1,5 \text{ [ms]}$$

A malha de regulação de corrente completa é ilustrada na figura 3.7.

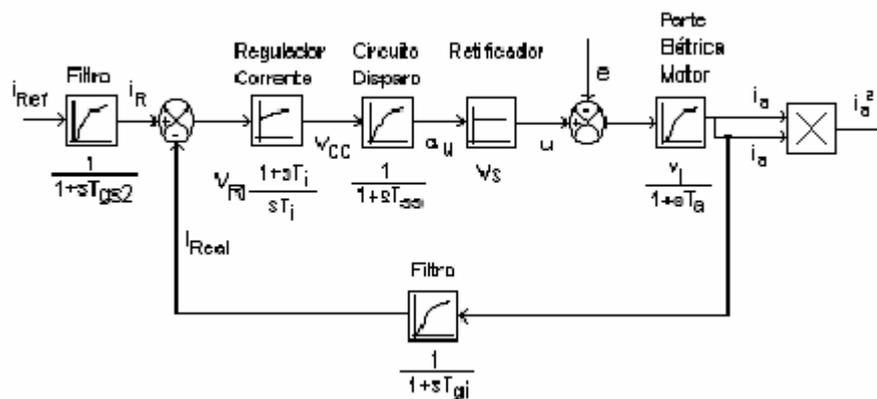


Figura 3.7 – Malha de Regulação de Corrente.

Aplicando a metodologia de otimização e controle proposta por Frohr; Ortttenburger [5], as constantes de tempo podem ser divididas em grandes e pequenas constantes de tempo, sendo uma grande constante de tempo maior que quatro vezes o somatório das demais. A constante de tempo T_a pode ser considerada uma grande constante de tempo de retardo. A escolha do tipo de regulador a ser utilizado obedece ao seguinte procedimento:

- Calcula-se a soma das pequenas constantes de tempo σ e ganho direto V_{si} .

$$\sigma = T_{SS} + T_{gi} = 3,0[ms] \quad (3.20)$$

$$V_{sia} = V_s v_i = 2,71 \cdot 5,18 = 4,03 \quad (3.21)$$

- Determina-se a relação entre a maior constante de tempo T_a e 4 vezes a soma das outras.

$$\frac{T_a}{4\sigma} = 6,7 > 1 \quad (3.22)$$

- Consulta-se a tabela 3.2 para a escolha do tipo de regulador, e do método de otimização a ser utilizado. Assim, para a malha de corrente, o sistema apresenta uma grande constante de tempo de retardo, dada por T_a e a relação $T_a/4\sigma > 1$. Portanto, o controlador requerido é do tipo PI (Proporcional Integral), e deve apresentar um método de otimização simétrica (OS). A constante de tempo de alisamento de sinal (T_{gs2}) e a constante de tempo equivalente de otimização da malha considerada (T_e) devem estar no intervalo $0...4\sigma$ e $2...4\sigma$, respectivamente.

A constante de tempo de alisamento de sinal, para o filtro do valor de referência para a malha de corrente, pode ser obtida por:

$$T_{gs2} = 4\sigma = 12[ms] \quad (3.23)$$

De acordo com a tabela 3.3, tem-se que:

O ganho do regulador de corrente é:

$$V_{Ri} = \frac{T_a}{2V_{st}\sigma} \rightarrow V_{Ri} = 0,95 \quad (3.24)$$

e a constante de tempo do regulador de corrente é:

$$T_i = 4\sigma \cdot \frac{T_a}{T_a + 3\sigma} \rightarrow T_i = 10,79[ms] \quad (3.25)$$

A constante de tempo equivalente de otimização da malha de corrente, incluindo o regulador e o filtro de referência, é dada por:

$$T_e = 2\sigma + \frac{1}{2} \cdot T_{gs2} = 12[ms] \quad (3.26)$$

Com estes dados obtidos, é possível obter a malha de regulação de velocidade, conforme a figura 3.8.

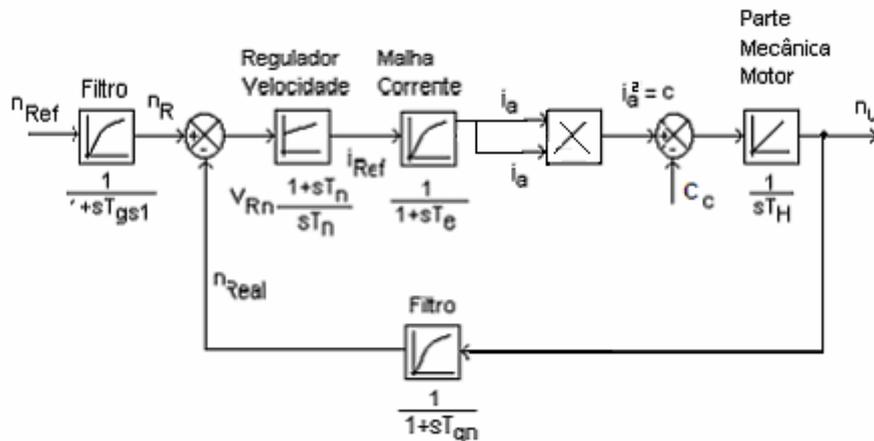


Figura 3.8 – Malha de Regulação de Velocidade.

Um filtro foi adicionado ao transdutor de velocidade devido à ondulação da tensão de saída do tacho gerador. A constante de tempo adotada foi $T_{gn} = 100$ [ms].

Tabela 3.2 – Escolha do tipo de controlador e método adequado de otimização para o sistema.

Grandes constantes de tempo (1)			Relação entre a grande constante e as pequenas constantes de tempo (2) $T/4\sigma$			Controlador requerido		Constante de tempo de alisamento do sinal (4) T_{gs}	Constante de tempo equivalente do sistema T_e
Constante de tempo de ação integral T_0	Constante de tempo de retardo Tr_1 Tr_2					Ação do controlador	Otimização		
			<1	>1	>>1				
	(3)					I	OM	2σ	
						PI	OM	2σ	
						PI	OS	$0...4\sigma$	
						P	OM	2σ	
						PID	OM	2σ	
						PID	OS	$0...4\sigma$	
						PD	OM	2σ	
						PI	OS	4σ	
						P	OM	2σ	
						PID	OS	4σ	
						PD	OM	2σ	

1. Não podem aparecer simultaneamente uma constante de tempo de ação integral T_0 e uma constante de tempo de retardo Tr_1 (a maior constante de tempo deve ser necessariamente T_0 e não Tr_1 , no caso de existirem T_0 e Tr_2 já está implícito que a maior constante de tempo é T_0 e a segunda maior é Tr_2).
2. A relação existente entre a grande constante de tempo T e a constante de tempo equivalente σ , se refere à constante de tempo Tr_1 característica do maior retardo de 1ª ordem ou a constante de tempo de ação integral T_0 .
3. Sistema controlado não inclui nenhum retardo que possa ser considerado como grande.
4. Constante de tempo do elemento de filtragem da entrada do sistema.

Tabela 3.3 – Ajuste dos parâmetros dos reguladores resultantes da aplicação dos métodos da otimização em função da linearidade, do módulo e da simetria.

Otimização	Ajuste dos parâmetros do controlador				
	T_I	T_n	T_V	V_R	
				Sem elementos de ação Integral no sistema	Com elementos de ação Integral no sistema
OL Otimização em função da linearidade	$4V_S\sigma$	T_{r1}	T_{r2}	$\frac{T_{r1}}{4V_S\sigma}$	-----
OM Otimização em função do módulo	$2V_S\sigma$	T_{r1}		$\frac{T_{r1}}{2V_S\sigma}$	
OS Otimização em função da simetria sem elementos de ação integral no sistema	-----	$\frac{4\sigma T_{r1}}{T_{r1} + 3\sigma}$		-----	
OS Otimização em função da simetria com elementos de ação integral no sistema	-----	4σ		$\frac{T_o}{2\sigma}$	
Ação do controlador	I	PI PID	PD PID	P PD	PI PID

Para esta malha de regulação de velocidade, a escolha do tipo de regulador a ser utilizado obedece à mesma metodologia usada para a malha de regulação de corrente.

- A soma das pequenas constantes de tempo σ' é:

$$\sigma' = T_e + T_{gn} = 112[ms] \quad (3.27)$$

- Determina-se a relação $T_H/4\sigma'$.

$$\frac{T_H}{4\sigma'} = 2,67 > 1 \quad (3.28)$$

- Consulta-se a tabela 3.2 para a escolha do tipo de regulador a ser utilizado. Assim, o sistema apresenta uma grande constante de tempo de ação integral T_H , sendo a relação $T_H/4\sigma' > 1$. Então, o controlador requerido é do tipo PI, e deve apresentar um método de otimização simétrica (OS). As constantes de tempo de alisamento de sinal (T_{gs1}) e equivalente de otimização (T_e'), se necessária, seriam ambas no valor de $4\sigma'$.

Logo,

$$T_{gs1} = T_e' = 4\sigma' = 448[ms] \quad (3.29)$$

De acordo com a tabela 3.3, tem-se que o ganho do regulador de velocidade é:

$$V_{Rn} = \frac{T_H}{2\sigma'} \rightarrow V_{Rn} = 5,35 \quad (3.30)$$

e a constante de tempo do regulador de velocidade é:

$$T_n = 4\sigma' \rightarrow T_n = 448[ms] \quad (3.31)$$

4. IMPLEMENTAÇÃO E MONTAGEM DO SISTEMA DE ACIONAMENTO :

4.1. ACIONAMENTO EM MONO-QUADRANTE ATRAVÉS DE UMA PONTE CONVERSORA GRAETZ :

O acionamento em mono-quadrante pode ser facilmente realizado através de uma ponte conversora Graetz totalmente controlada. Esse equipamento é constituído por seis tiristores (SCR's) conectados de acordo com o esquema da figura 4.1 abaixo :

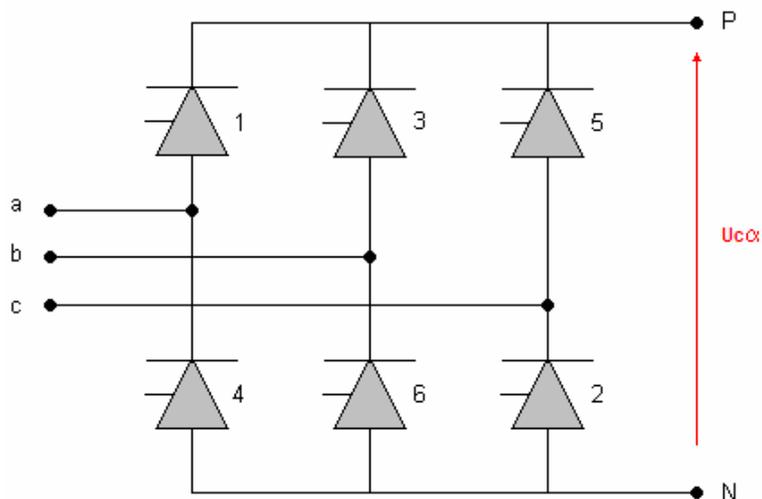


Figura 4.1 – Ponte Conversora Graetz (totalmente controlada).

Controlando o ângulo de disparo dos tiristores (de 0° a 180°) pode-se assim variar a tensão (CC) entre os terminais P e N ($U_{c\alpha}$) – correspondente a tensão de armadura do motor de corrente contínua a ser acionado. Essa tensão pode ser representada pela seguinte expressão :

$$U_{c\alpha} = 2,34 \cdot (V_{ff} \cdot \cos \alpha) \quad (4.1)$$

onde :

α - ângulo de disparo dos tiristores.

V_{ff} - tensão fase-neutro (rms) da rede de alimentação trifásica.

Em geral, o controle do ângulo α é realizado através de circuitos eletrônicos de disparo.

Para esse tipo de acionamento, implementou-se um sistema de acordo com o diagrama abaixo :

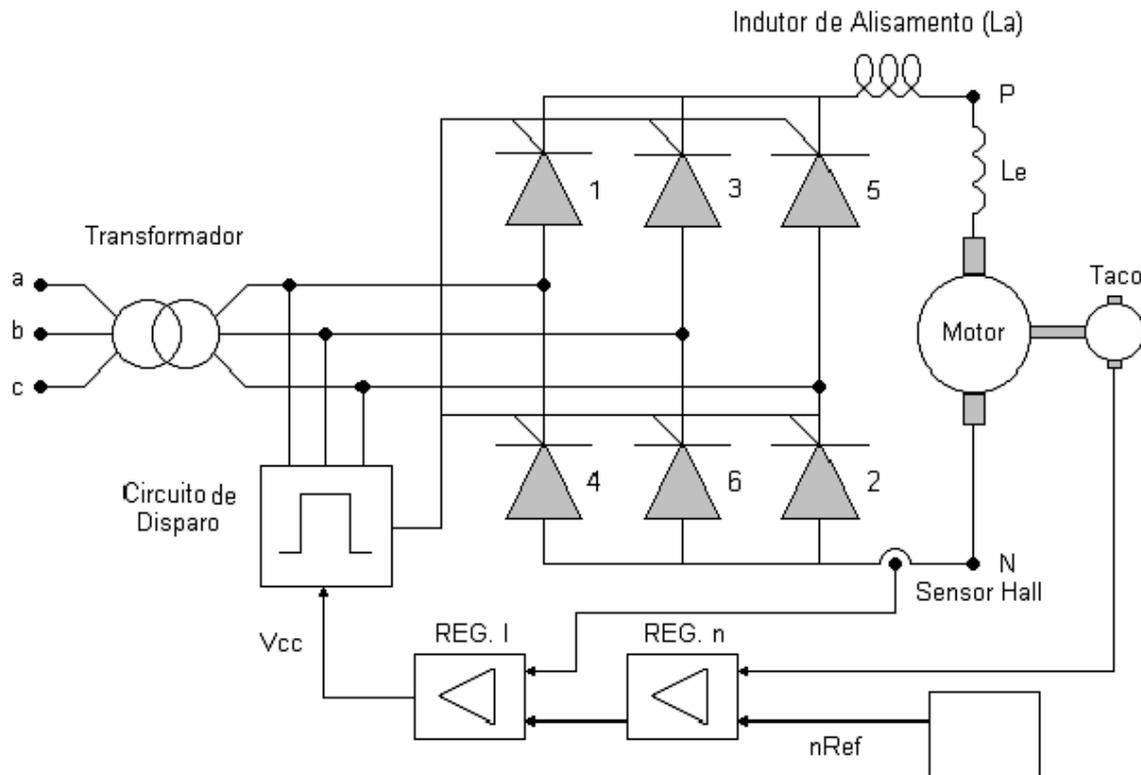


Figura 4.2 – Acionamento em Mono-quadrante Através de uma Ponte Conversora Graetz.

Para simulação de uma carga acoplou-se ao eixo do motor de corrente contínua um gerador síncrono alimentando uma carga resistiva trifásica.

4.2. CIRCUITO DE DISPARO :

Foram utilizados circuitos eletrônicos de disparo do tipo rampa, implementados com o CI TCA 780 (Icotron-Siemens), um circuito integrado desenvolvido para controlar o ângulo de disparo de tiristores, transistores e triacs continuamente de 0° a 180° , garantindo uma faixa total de controle em circuitos elétricos.

A figura 4.3 e 4.4 ilustram as formas de onda típica do TCA 780 e desses dispositivos de disparo utilizando esse ci.

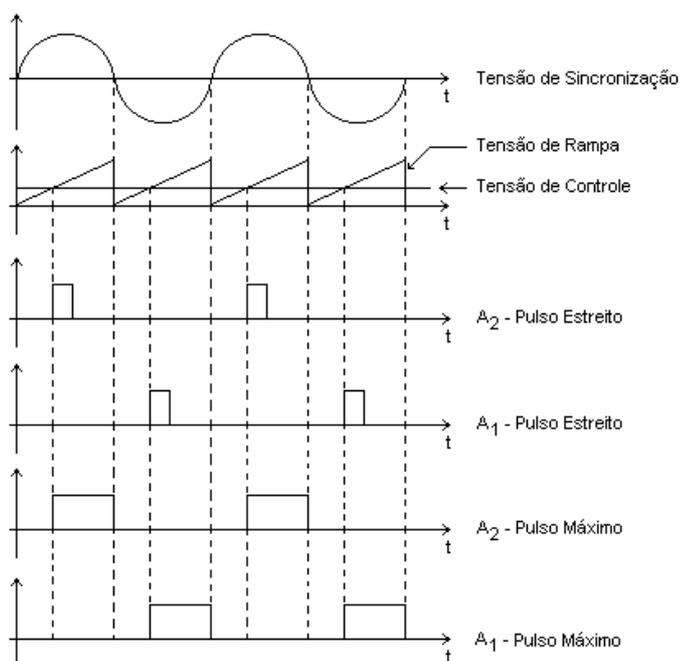


Figura 4.3 – Formas de Ondas Típicas do TCA 780.

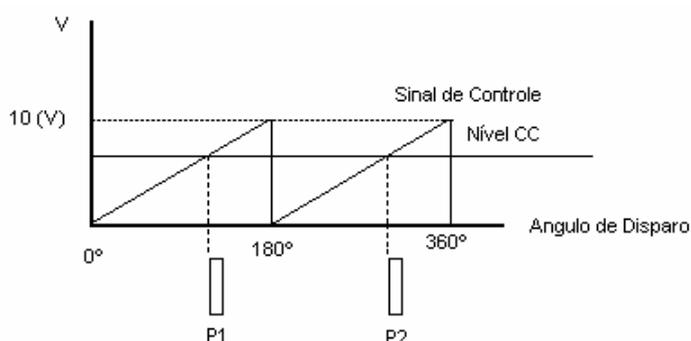


Figura 4.4 – Circuito de Disparo Tipo Rampa.

A interseção do nível CC com a rampa, internamente gerada no TCA 780, produz os pulsos de disparo.

O diagrama esquemático do circuito de disparo utilizado encontra-se no Anexo A.

5. O CONTROLE DIGITAL DOS ACIONAMENTOS :

A introdução de um computador ou microprocessador digital para realizar as operações de controle torna o sistema controlado mais versátil tornando possível a implementação de várias técnicas de controle modais, tais como controle inteligente, controle ótimo e robusto, realimentação de estados, com menores custos, associados à melhoria da precisão obtida.

Serão apresentados nesse trabalho os resultados obtidos pela utilização de um microcomputador PC 486, para o acionamento e controle digital de corrente e velocidade de um motor de corrente contínua (MCC) operando em um quadrante, substituindo assim, os tradicionais reguladores constituídos de circuitos utilizando amplificadores operacionais.

5.1. AQUISIÇÃO E CONVERSÃO DE DADOS :

Nesse computador, o controle do sistema é efetuado diretamente através de um software de controle e uma placa de aquisição de dados com conversores A/D e D/A acoplada junto a esse computador. A malha de controle de velocidade e limitação da corrente de armadura da máquina CC foi implementada em laboratório, conforme mostrado na figura 5.1.

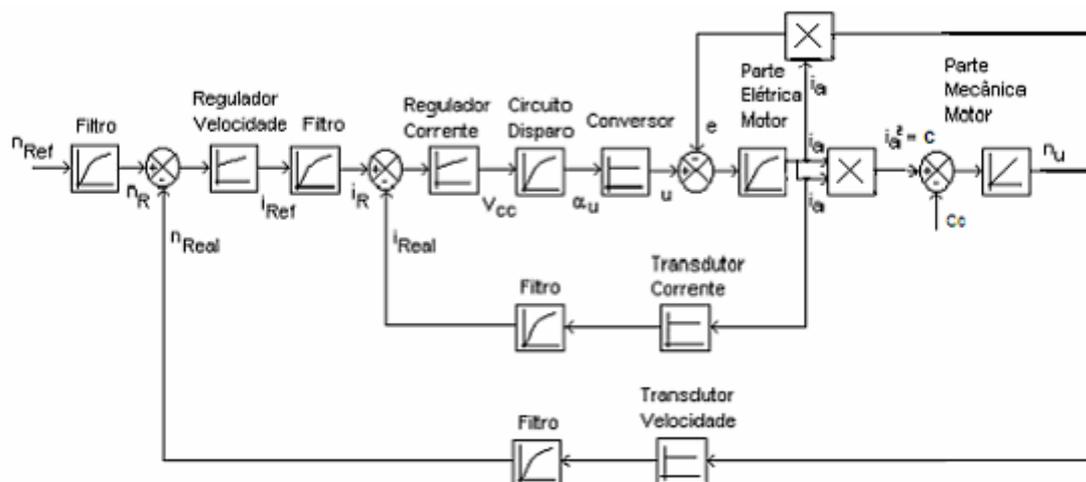


Figura 5.1 – Diagrama Esquemático de Controle Digital em Malha Fechada de uma Máquina CC.

A placa utilizada é uma placa PCL-711B PC-Multilab Card da Advantech Co, com conversão A/D, e D/A, e entradas e saídas digitais. A conversão A/D, tem resolução de 12 bits, com 8 canais de entrada, programáveis para faixas de entrada de ± 5 [V], $\pm 2,5$ [V], $\pm 1,25$ [V], $\pm 0,625$ [V], $\pm 0,3125$ [V], com tempo de conversão de até 25[μ s]. A conversão D/A, tem a mesma resolução (12 bits), mas apenas com um canal de saída, com tempo de acomodação de 30 [μ s], e faixas de saída de 0 a +5 [V] ou 0 a +10 [V].

Foram utilizados dois canais de entrada, para conversão A/D, correspondendo aos sinais de realimentação de velocidade e corrente, e um canal de saída, para o sinal de controle V_{CC} .

O transdutor de corrente utilizado foi um sensor Hall e o transdutor de velocidade um taco gerador acoplado ao eixo do motor de corrente contínua. Ambos os transdutores de corrente e velocidade incluem divisor resistivo para limitar a tensão de entrada para o conversor A/D, sendo em seguida os sinais de tensão convertidos por *software* para condições de velocidade e carga proporcionais à nominal.

Os programas ou *softwares* de controle utilizados foram desenvolvidos em linguagem C++, sendo responsáveis pela aquisição de dados de entrada, pelas conversões A/D e D/A, implementação do algoritmo de controle para as malhas de velocidade e corrente, e geração do sinal de controle (V_{CC}) para o circuito de disparo. Os programas incluem ainda temporização, diagnóstico de falha e ajuste de parâmetros *on-line*.

5.2. ESCOLHA DOS REGULADORES :

5.2.1. REGULADOR PI :

De acordo com o que foi demonstrado no capítulo 3, tem-se que o regulador PI (proporcional - integral) é o mais apropriado para esse tipo de controle, sendo muito utilizado em aplicações industriais onde se deseja que o sistema apresente um comportamento preestabelecido, atendendo a algumas especificações, como: *overshoot*, tempo de acomodação, erro em regime permanente, etc. O que é relativamente fácil de se obter, considerando a grande flexibilidade oferecida por esse tipo de regulador, sendo apenas conveniente

adotar-se um critério de otimização, efetuando assim os ajustes necessários para o controle e regulação de velocidade e corrente em malha fechada.

A equação idealizada de um regulador PI é dada por :

$$Gc(s) = k \cdot \left[\frac{1 + \tau_I \cdot s}{\tau_I \cdot s} \right] \quad (5.1)$$

Onde : K - ganho proporcional.

τ_I - tempo integral.

Através de uma sistemática bem simples, utilizando métodos tradicionais e conceitos básicos de integração numérica e Transformada-Z, pode-se chegar a uma equação recursiva capaz de representar o comportamento desse sistema.

Quando se emprega modelos ou equações, para obter a expressão recursiva, basta substituir os operadores \int ou $1/s$, pela aproximação desejada em Z, arranjar os termos para terem a forma de z^{-k} , os quais são trocados pelos índices n-k. A aproximação trapezoidal é conhecida por transformação bilinear, bastante utilizada pois gera bons resultados nos processamentos.

Através da equação (5.1) e da integração trapezoidal de Arne Buxbaum [4], pode-se enfim obter a equação recursiva capaz de representar o comportamento do regulador PI.

$$\frac{1}{s} \approx \frac{T}{2} \cdot \frac{z+1}{z-1} \quad (5.2)$$

Onde : T é o tempo de amostragem.

Substituindo-se a equação (5.2) em (5.1), tem-se :

$$y_k = y_{k-1} + b_1 \cdot (e_k - b_2 \cdot e_{k-1}) \quad (5.3)$$

sendo:

$$b_1 = k \cdot (1 + c); \quad b_2 = \frac{(1 - c)}{(1 + c)}; \quad c = \frac{T}{2 \cdot \tau_I}$$

O fluxograma do controle digital é ilustrado na figura 5.2.

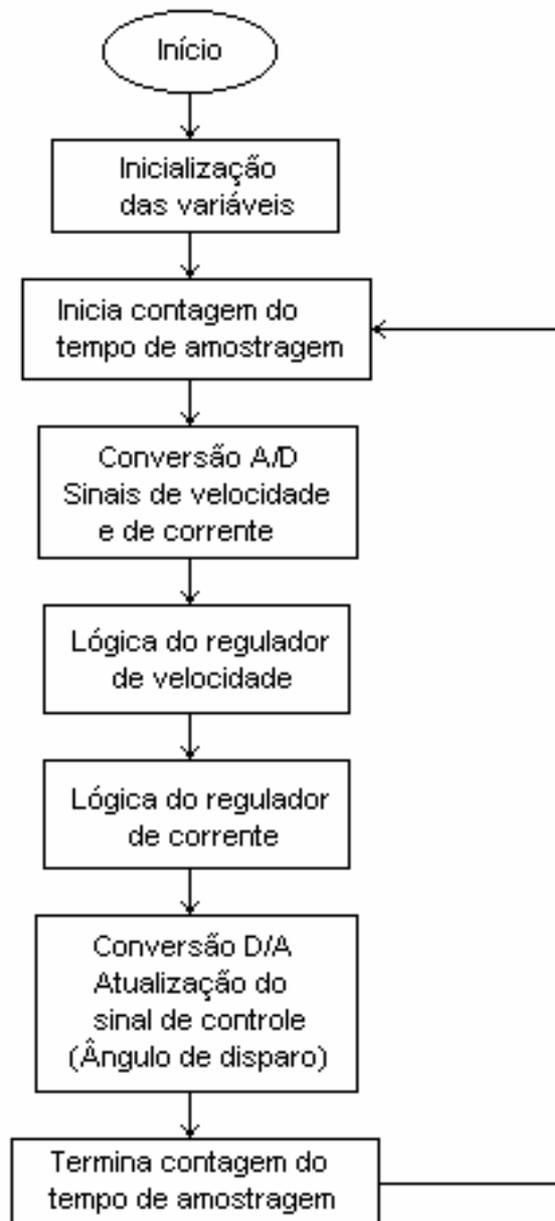


Figura 5.2 – Fluxograma do Controle Digital.

O programa completo, implementado para o acionamento utilizando reguladores PI, é apresentado no Anexo B.

5.2.2. REGULADOR FUZZY :

Para que se possa entender o funcionamento desse tipo de regulador é conveniente entender um pouco da lógica Fuzzy ou Lógica “Nebulosa”. Baseada na teoria dos conjuntos difusos (*fuzzy sets*) proposta por ZADEH em 1965, esta lógica está se mostrando uma das mais interessantes tecnologias para aplicações em sistemas de controle sofisticados, proporcionando de maneira simples uma diminuição nos custos e um aumento na eficiência desses sistemas.

Propõe-se nesse trabalho, a introdução desta tecnologia no acionamento controlado de uma máquina CC com excitação em série visando o controle de velocidade e limitação de corrente, sendo que segundo [8], esse tipo de sistema (série) pode ser considerado um sistema não linear, e onde pode-se encontrar os melhores resultados da Lógica Fuzzy em relação as lógicas convencionais.

5.2.2.1. CONTROLE NEBULOSO (*FUZZY CONTROL*) :

A teoria do controle utilizando lógica nebulosa (*fuzzy logic*) caracteriza as variáveis de interesse através de expressões lingüísticas, tais como “muito grande”, “grande”, “pequeno”, “quente”, “frio”, etc.

Estas expressões lingüísticas são representadas numericamente por conjuntos difusos (*fuzzy sets*), sendo cada conjunto caracterizado por uma função de pertinência que varia de 0 a 1.

O algoritmo de controle fuzzy executa ações de controle escritas em termos de conceitos lingüísticos tendo como referência a intuição e a experiência de um operador, projetista ou pesquisador.

O controle não exige a precisão de um modelo matemático e contudo se adapta bem a um processo onde o modelo é desconhecido ou mal definido.

Os componentes de um sistema de controle convencional e um fuzzy são similares. A diferença reside no fato de que no fuzzy existe um elemento que converte as entradas em suas representações difusas, o fuzificador (*fuzzifier*), e outro que converte a saída difusa inferida em um valor solução, numérico e preciso, o defuzificador (*defuzzifier*). A figura 5.3 ilustra com diagrama de blocos um controlador fuzzy (*fuzzy controller*) e um convencional.

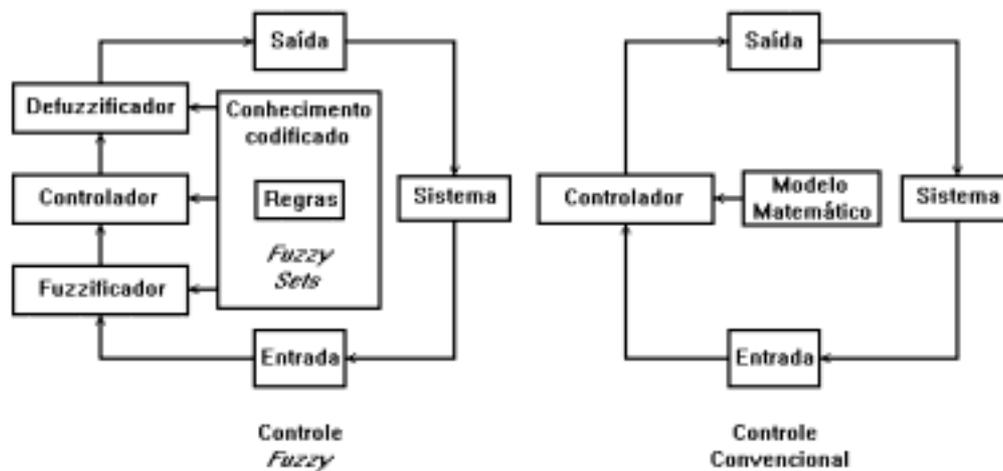


Figura 5.3 - Comparação entre um controlador fuzzy e um controlador convencional.

Em um sistema fuzzy, o valor de uma entrada é convertido pelo fuzzificador. Em seguida, regras de controle que sejam satisfeitas são executadas. Este processo produz um novo conjunto fuzzy representando cada saída ou solução variável. O defuzzificador fornece um valor para a saída variável deste novo conjunto fuzzy. O valor de saída atua no sistema físico. A mudança é captada por sensores e o controle reinicia-se.

As variáveis fuzzy são definidas num intervalo numérico comumente chamado de “universo de discurso”. As regras fuzzy são tipicamente da forma condicional SE-ENTÃO, como segue:

- SE(“x” é A e “y” é B) ENTÃO (“z” é C)

Onde “x” e “y” são variáveis fuzzy e A, B, e C são subconjuntos fuzzy no universo de discurso X, Y, Z respectivamente. Se a condição expressa na regra é satisfeita, então é executada a ação especificada.

Para projetar um controlador fuzzy, uma série de regras tem que ser construídas.

No caso, o erro (NE), a integral do erro (\int_{IE}) e a variação do sinal de controle (V_{cc}) são consideradas variáveis fuzzy, com possíveis valores dados por funções de pertinência (μ) a conjuntos fuzzy tais como pequeno positivo, pequeno negativo, zero, e assim por diante.

As funções de pertinência para a variável erro poderiam ser representadas conforme a figura 5.4.

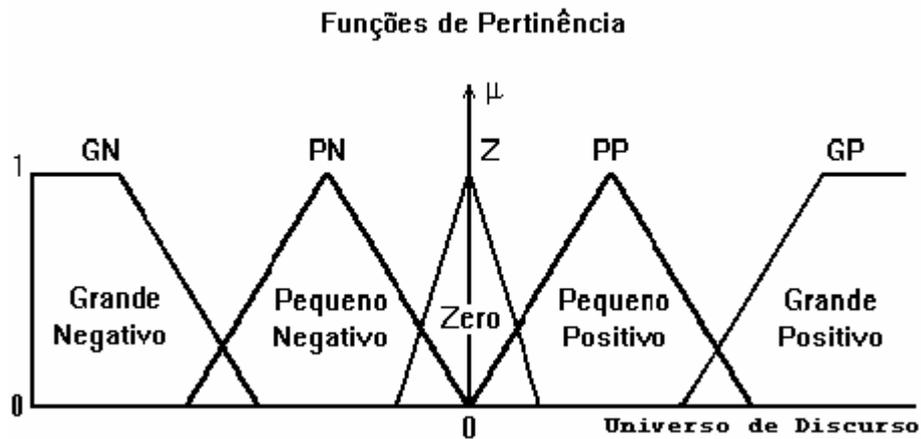


Figura 5.4 - Representação das funções de pertinência típicas para uma variável fuzzy.

5.2.2.2. PROJETO DO CONTROLADOR FUZZY :

Seja uma malha de limitação de corrente. Deseja-se substituir os convencionais reguladores PI por um controlador fuzzy. As variáveis de entrada do controlador fuzzy consideradas de interesse são obtidas por:

$$IE(k) = I_{real}(k) - I_{ref}(k) \tag{5.4}$$

$$IE(k) = ie \times GI \tag{5.5}$$

$$\int_{IE(k)} = \int_{IE(k-1)} + GV \times IE \times dt \tag{5.6}$$

Onde:

dt – Tempo incremental;

ie – Erro de corrente;

IE(k) - Erro de corrente escalonado;

$I_{real}(k)$ - Realimentação de corrente;

$I_{ref}(k)$ - Corrente de referência;

$\int_{IE(k)}$ - IIE- Integral do erro de corrente;

k - Intervalo de amostragem.

GI – Escalonamento de ganho proporcional;

GV - Escalonamento de ganho integral (inverso da constante de tempo integral);

As funções de pertinência são mostradas na figura 5.5. As variáveis estão expressas em “pu”. O universo de discurso é adotado conforme condições operacionais.

Cada variável de controle e de ação é decomposta num conjunto de regiões difusas (fuzzy sets) denominados “labels”. Geralmente cada “label” ou

subconjunto fuzzy tem forma assimétrica, tendendo-se a uma maior concentração próximo à origem. Isto permite maior precisão no controle próximo ao ponto de operação estável.

O número de “labels” associados a uma dada variável deve ser ímpar entre 5 e 9. Além disso, a extremidade de cada label deve sobrepor os vizinhos. Esta sobreposição propicia ao controlador fuzzy uma ação contínua e estável. A sobreposição deve estar entre 10 a 50% do espaço do vizinho e a soma dos pontos verticais da sobreposição deve preferencialmente ser menor que 1.

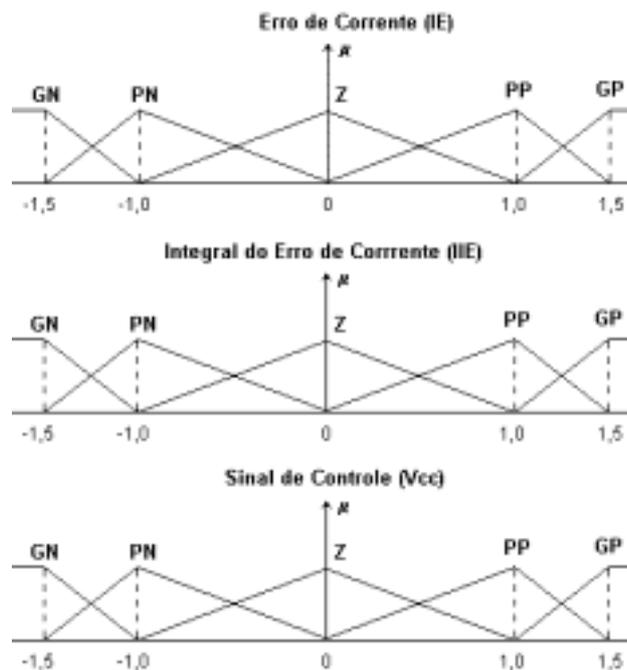


Figura 5.5 – Representação das funções de pertinência adotadas para a malha de controle e limitação de corrente.

Onde:
 IE – Erro de corrente escalonado
 IIE – Integral do erro usado para regulador PI
 GN – Grande negativo
 PN – Pequeno negativo
 Z – Zero
 PP – Pequeno positivo
 GP – Grande positivo

Como ilustrado na figura 5.5, um valor numérico de entrada pode ser um membro de mais do que um conjunto fuzzy. Para isto, basta ele estar situado numa região de sobreposição. Isto significa que, para um específico par de valores IE e IIE, mais do que uma regra pode ser ativada. Portanto, é necessário a

existência de um meio de combinar as ações de controle ativadas por cada regra, de forma que uma ação simples mas significativa, com contribuição de todas estas regras, seja efetuada. Esta é a função do defuzificador.

Existem vários métodos de transformação do conjunto difuso de saída em um valor preciso, representando a solução. O método de centróide é o mais indicado para sistemas de controle. Através dele calculamos o centro de gravidade do conjunto difuso obtido, produzindo um resultado que é sensível a todas as regras e tende a se deslocar de forma suave através da superfície de controle.

A elaboração das regras são baseadas na intuição e experiência do projetista. O número de regras está relacionado com o número de variáveis de controle. No caso em estudo, existem duas variáveis de controle, cada uma delas divididas em 5 subconjuntos fuzzy, produzindo 25 possíveis combinações de entrada. Deste modo, 25 regras são necessárias.

O controlador fuzzy é projetado através de um algoritmo de controle considerando as ações de fuzzificação, defuzzificação e todas as regras.

A tabela 5.1 mostra as regras base adotadas para o controle fuzzy de corrente na forma de uma matriz.

IIE / IE	GN	PN	ZE	PP	GP
GN	GN	GN	GN	PN	ZE
PN	GN	GN	PN	ZE	PP
ZE	GN	PN	ZE	PP	GP
PP	PN	ZE	PP	GP	GP
GP	ZE	PP	GP	GP	GP

Tabela 5.1 - Regras base para o controle fuzzy de corrente.

A defuzzyficação pelo método centróide seleciona a saída como um valor correspondente ao centro de gravidade da função de pertinência de saída, sendo dado pela seguinte expressão:

$$X_0 = \frac{\int x \cdot \mu(x_1) \cdot \mu(x_2) dx}{\int \mu(x_1) \cdot \mu(x_2) dx} = \frac{\sum_{i=1}^n x_i \mu(IE_i) \cdot \mu(II E_i)}{\sum_{i=1}^n \mu(IE_i) \cdot \mu(II E_i)} \quad (5.6)$$

Onde:

X_0 – Centro de gravidade da função de pertinência de saída;

x_i – Centro de gravidade da função de pertinência de saída ativada por cada regra;

$\mu(IE_i) \cdot \mu(II E_i)$ – Produto das funções de pertinência ativadas por cada regra.

Iremos mostrar na tabela abaixo, um comparativo entre os ganhos e constantes de tempo dos reguladores PI convencional e PI fuzzy.

<i>Parâmetros</i>		<i>Projeto</i>		<i>Adotado</i>	
Convencional	Fuzzy	Convencional	Fuzzy	Convencional	Fuzzy
Vrn	ge1	5,35	5,35	5,35	4
Tn	gv1	448 ms	2,23/s	448 ms	2,3/s
Vri	ge2	0,95	0,95	0,1	0,1
Ti	gv2	10,79 ms	92,67/s	10,80 ms	90/s

Tabela 5.2 – Relação entre os parâmetros dos reguladores.

6. RESULTADOS OBTIDOS :

Foi utilizado em laboratório um motor CC de 1,7 [KW], 1500 [rpm], 220 [V] e 7,72 [A] de corrente nominal, acionado em mono-quadrante através de uma ponte Graetz de acordo com o que foi apresentado anteriormente, verificando seu comportamento dinâmico em situações distintas, como na partida (com carga) e diante bruscas variações na referência ou na carga no eixo da máquina em funcionamento, tanto com os reguladores PI como para os reguladores com Lógica Fuzzy.

6.1. RESULTADOS EXPERIMENTAIS :

Todos os resultados apresentados nesse item foram obtidos através de medições realizadas em laboratório, sendo registrados através de um osciloscópio digital (Tektronix) e adquiridos via computador, sendo que na escala de velocidade, cada divisão vertical representa 1000 (rpm) e na escala de corrente, cada divisão vertical representa 5 (A).

Em todos os softwares, a corrente limite foi ajustada computacionalmente em 1,20 [pu] enquanto os demais parâmetros foram ajustados on-line de forma a melhorar a resposta dinâmica dos sistemas analisados, podendo ser observados nos programas em anexo (Anexo B).

As figuras 6.1 e 6.2 referem-se ao sistema com controle digital do tipo **proporcional-integral (PI)**, para **controle de velocidade** de motor de corrente contínua com excitação série **em 1 pu**. Sendo que na primeira condição, partiu-se a máquina com carga e após alguns segundos retirou-se essa carga bruscamente, colocando-a novamente depois. Já na segunda condição (Fig. 6.2) efetuou-se uma ligeira variação na referência de velocidade, de (-0,3 pu), voltando a condição nominal novamente.

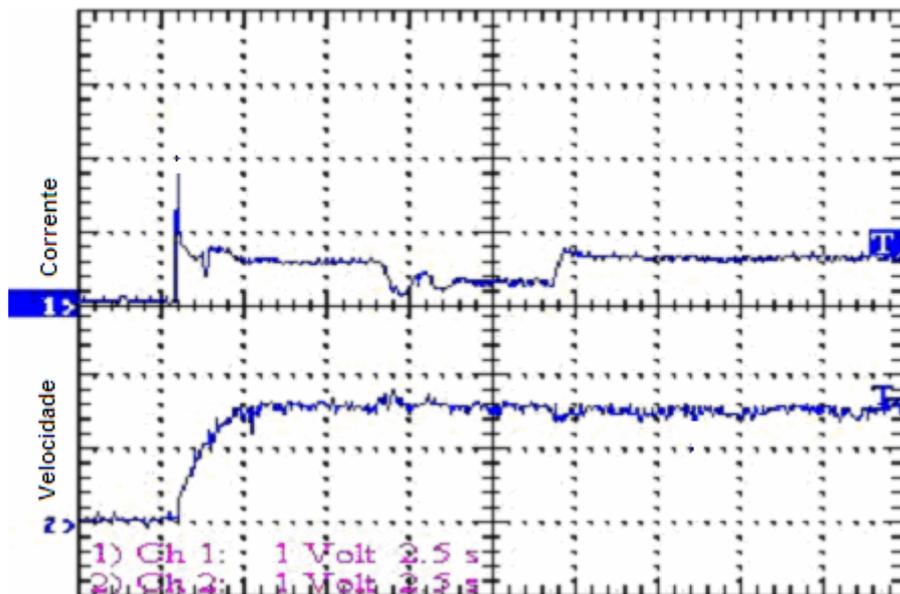


Figura 6.1 – Partida com Carga e Degrau Negativo e Positivo de Carga

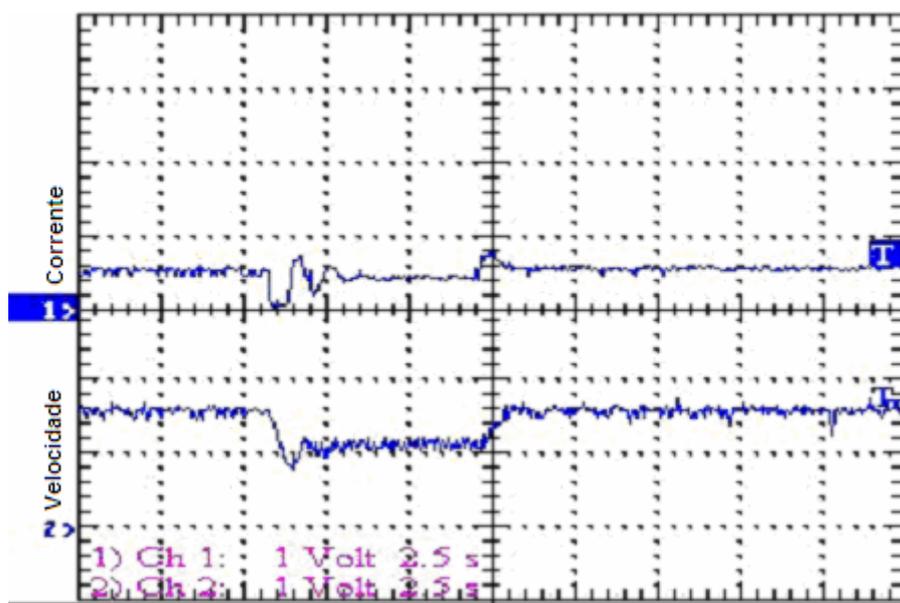


Figura 6.2 – Degrau Negativo e Positivo na Referência

Do mesmo modo, as figuras 6.3 e 6.4 referem-se ao sistema com controle digital do tipo **Fuzzy (PI-Fuzzy)** para **controle de velocidade** de motor de corrente contínua com excitação série em 1 pu. Sendo as condições as mesmas verificadas anteriormente.

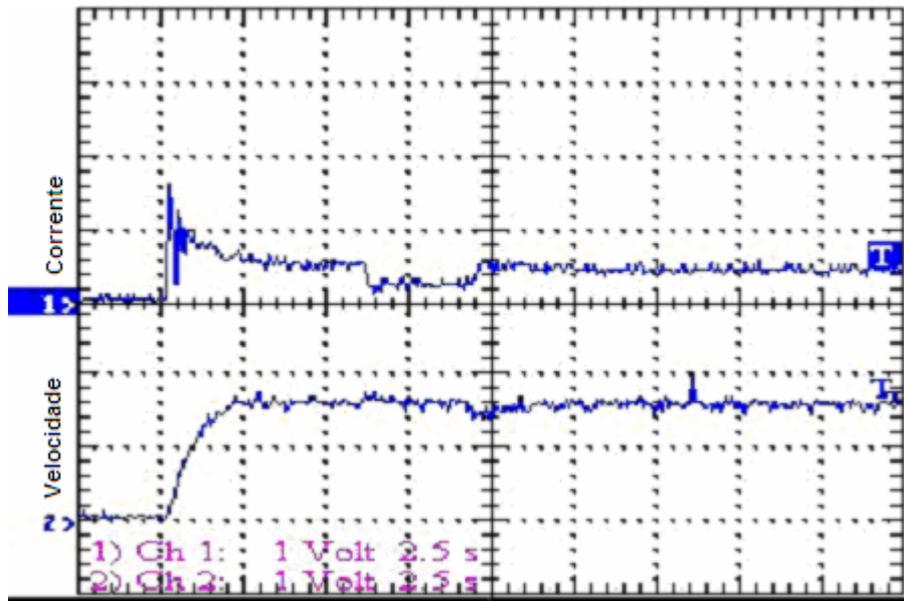


Figura 6.3 – Partida com Carga e Degrau Negativo e Positivo de Carga

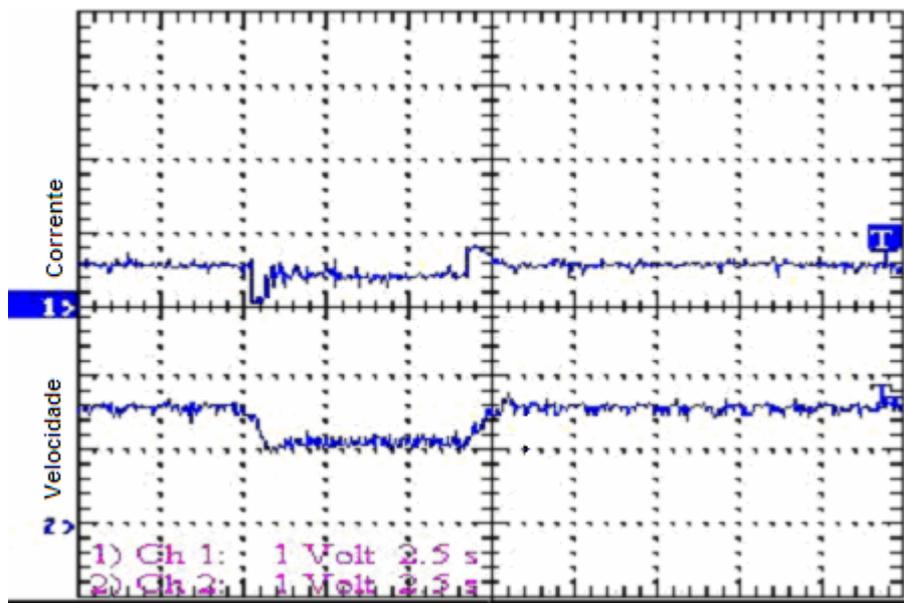


Figura 6.4 – Degrau Negativo e Positivo na Referência

Pode-se observar a rápida resposta do sistema em termos de velocidade para variações na carga e na referência (nref).

Agora, as figuras 6.5 e 6.6 referem-se ao sistema com controle digital do tipo **proporcional-integral (PI)**, para **controle de velocidade** de motor de corrente contínua com excitação série em **0,5 pu**. Sendo que na primeira condição, partiu-se a máquina com carga e após alguns segundos retirou-se essa carga bruscamente, colocando-a novamente depois. Já na segunda condição (Fig. 6.6) efetuou-se uma ligeira variação na referência de velocidade, de (-0,3 pu), voltando a condição nominal novamente.

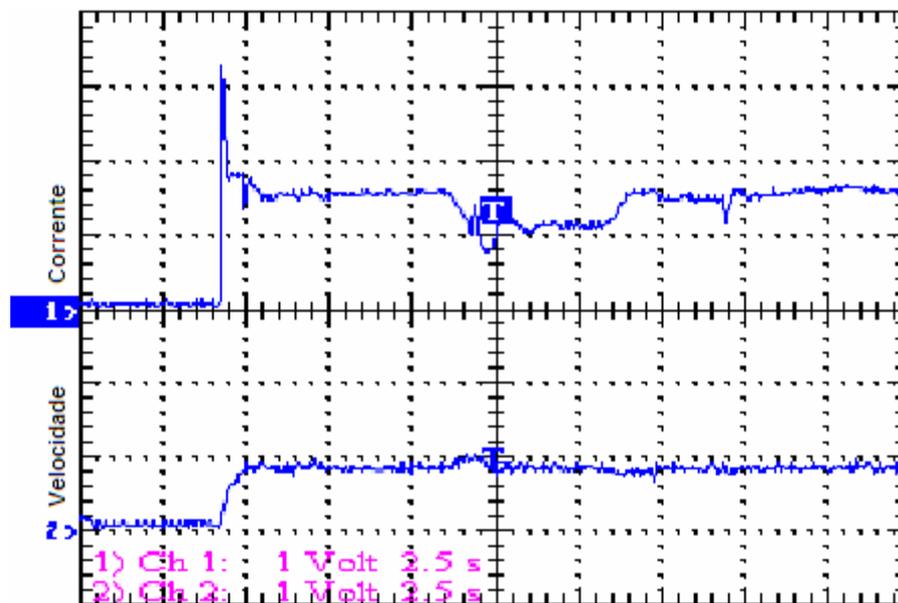


Figura 6.5 – Partida com Carga e Degrau Negativo e Positivo de Carga

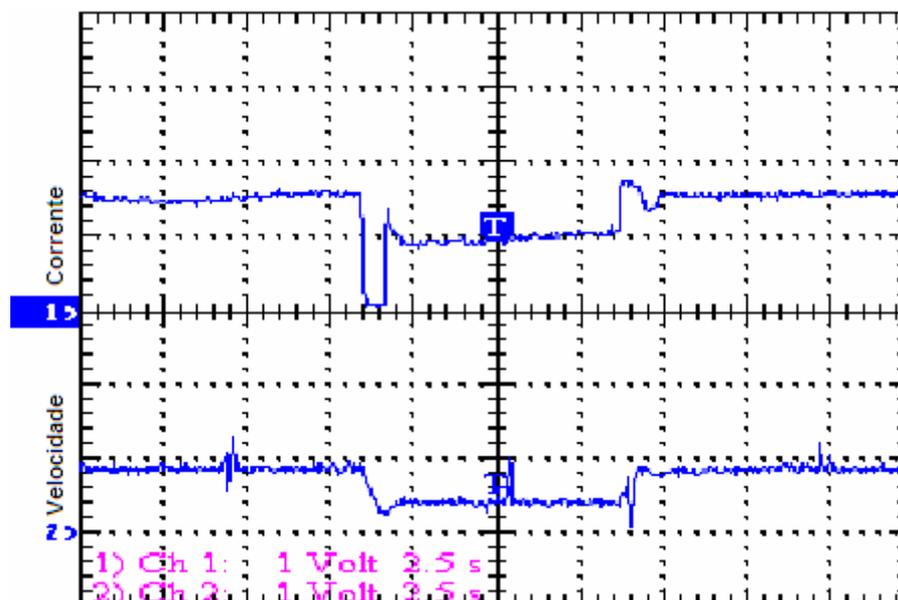


Figura 6.6 – Degrau Negativo e Positivo na Referência

Do mesmo modo, as figuras 6.7 e 6.8 referem-se ao sistema com controle digital do tipo **Fuzzy (PI-Fuzzy)** para **controle de velocidade** de motor de corrente contínua com excitação série em **0,5 pu**. Sendo as condições as mesmas verificadas anteriormente.

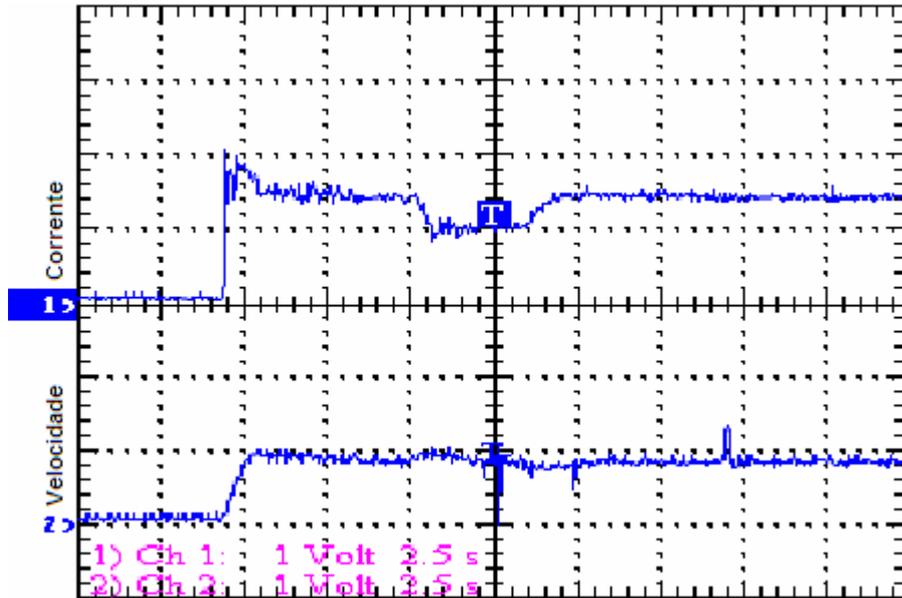


Figura 6.7 – Partida com Carga e Degrau Negativo e Positivo de Carga

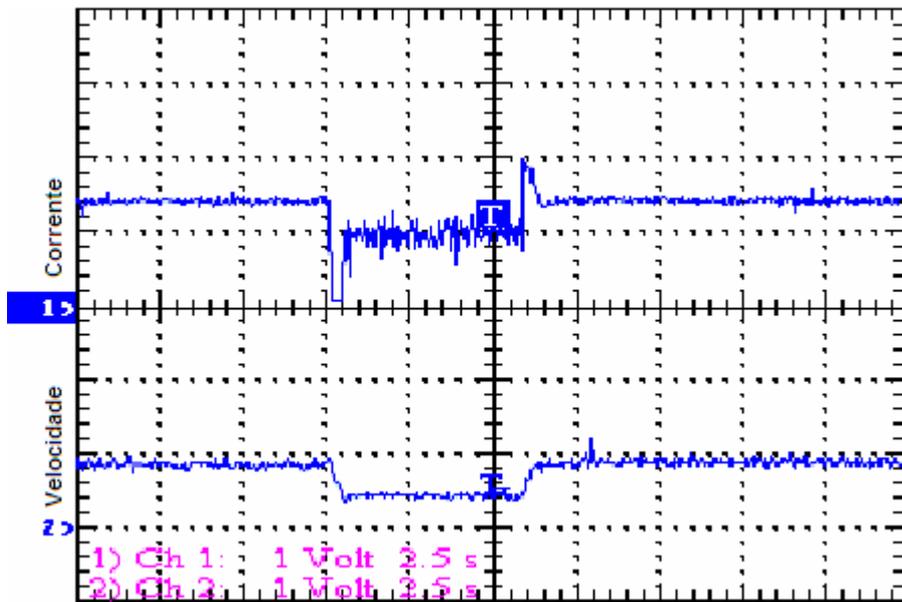


Figura 6.8 – Degrau Negativo e Positivo na Referência

Neste caso, também observamos uma rápida resposta do sistema em termos de velocidade para variações na carga e na referência (n_{ref}).

6.2. OUTROS RESULTADOS OBTIDOS :

Da mesma forma que no item anterior, iremos apresentar as curvas obtidas em laboratório para os reguladores digitais do tipo **proporcional-integral (PI)** e do tipo **Fuzzy (PI-Fuzzy)**, para **controle de velocidade** do motor de corrente contínua com excitação série **em 1 pu**, sendo que nestas aquisições, alteramos alguns parâmetros dos valores de ganho e tempo nas malhas de controle de velocidade e corrente dos controladores, sendo adotados os valores a seguir:

$$V_{rn} = 4$$

$$V_{ri} = 0,1$$

$$T_n = 0,5$$

$$T_i = 0,02$$

As figuras 6.9 e 6.10 referem-se ao sistema com controle digital do tipo proporcional-integral (PI), para controle de velocidade de motor de corrente contínua com excitação série. Sendo que na primeira condição, partiu-se a máquina com carga e após alguns segundos retirou-se essa carga bruscamente, colocando-a novamente depois. Já na segunda condição (Fig. 6.10) efetuou-se uma ligeira variação na referência de velocidade, de (-0,3 pu), voltando a condição nominal novamente.

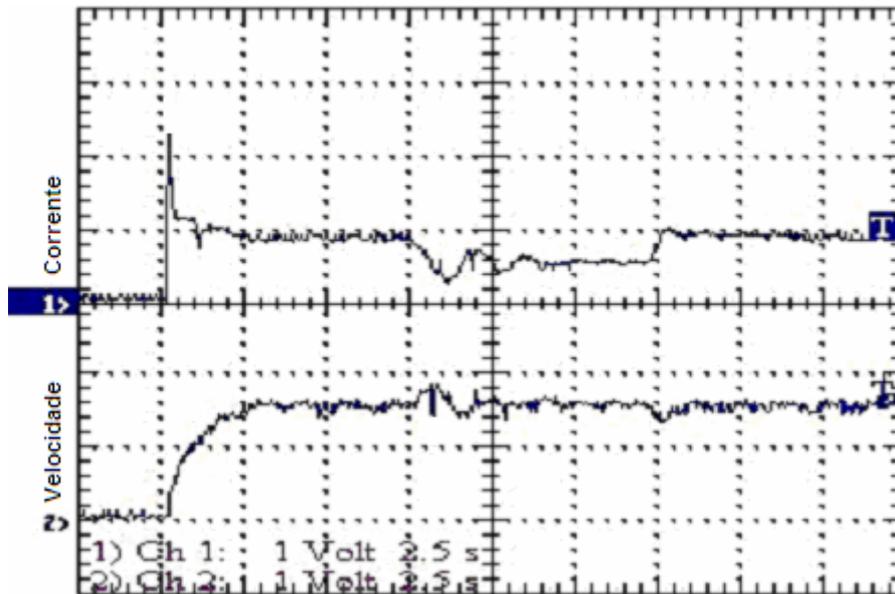


Figura 6.9 – Partida com Carga e Degrau Negativo e Positivo de Carga

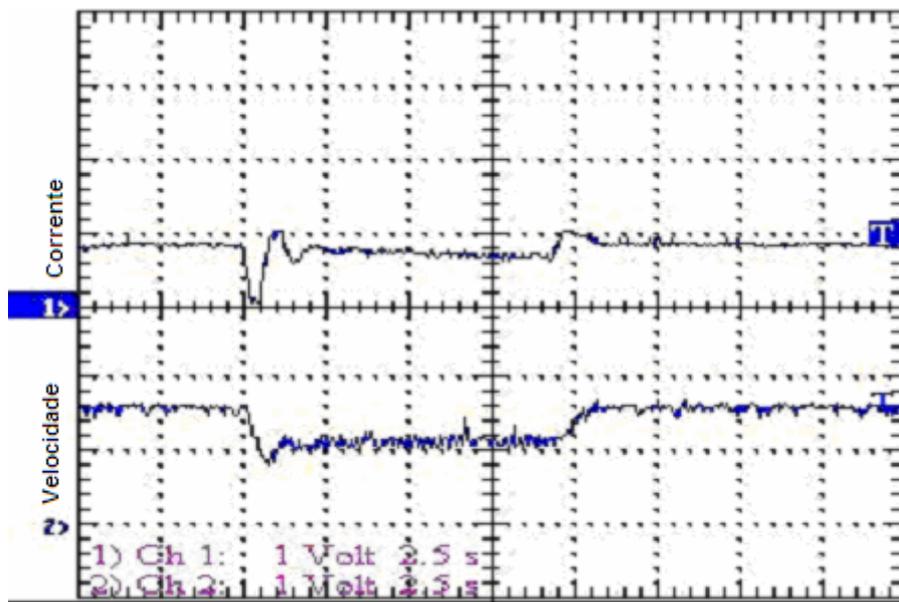


Figura 6.10 – Degrau Negativo e Positivo na Referência

Do mesmo modo, as figuras 6.11 e 6.12 referem-se ao sistema com controle digital do tipo Fuzzy (PI-Fuzzy) para controle de velocidade de motor de corrente contínua com excitação série. Sendo as condições as mesmas verificadas anteriormente.

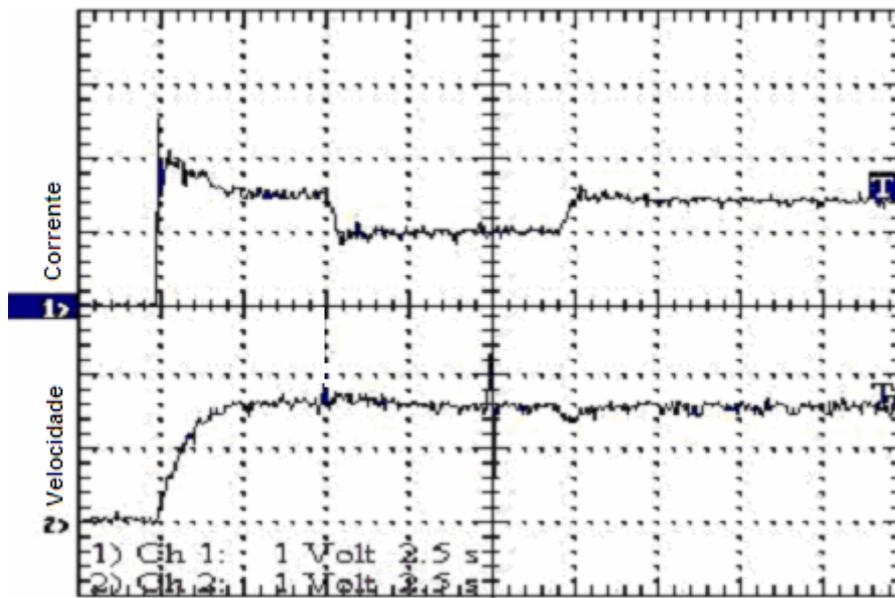


Figura 6.11 – Partida com Carga e Degrau Negativo e Positivo de Carga

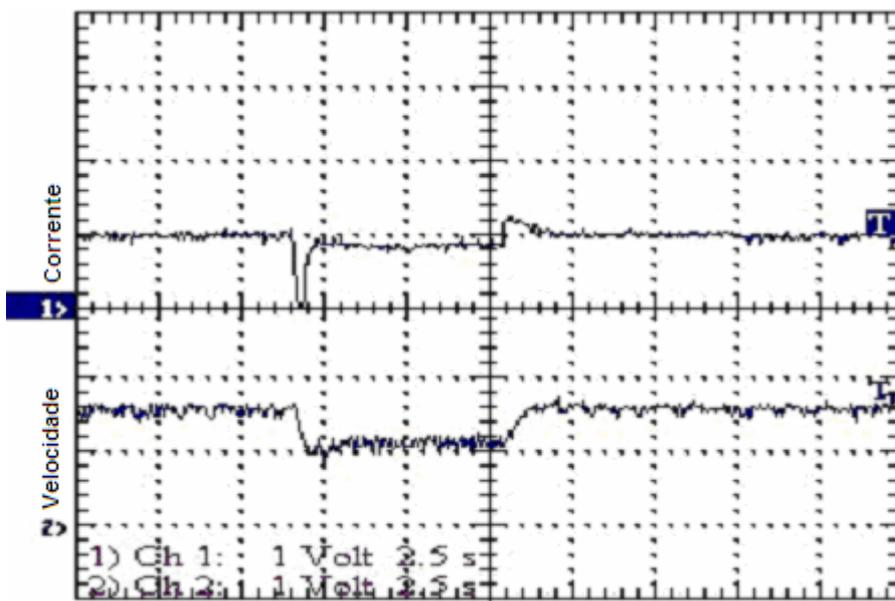


Figura 6.12 – Degrau Negativo e Positivo na Referência

7. CONCLUSÕES :

Os resultados experimentais obtidos em laboratório foram satisfatórios, pois o controle digital com reguladores Fuzzy para o acionamento de uma máquina de corrente contínua com excitação em série se mostrou estável e eficiente, tendo uma resposta mais rápida e menos oscilatória do que aquela apresentada pelo reguladores convencionais do tipo PI.

De fato, sendo o acionamento série em Corrente Contínua um sistema do tipo Não linear essa melhoria era esperada, de acordo com [3]. Os reguladores Fuzzy se mostraram superiores com resposta transitória melhor, ou seja, tanto para os degraus na referência como na carga, houve uma estabilização mais rápida e menos oscilatória na velocidade e corrente da máquina controlada.

A grande contribuição deste trabalho foi a comprovação da melhor performance do regulador digital Fuzzy, comparativamente ao regulador digital convencional, para sistema não linear de acionamento controlado para motor série de corrente contínua. É também um sistema de simples implementação para aplicações em acionamentos, tanto para máquinas de corrente contínua como também para máquinas de corrente alternada, substituindo os tradicionais controladores analógicos e permitindo também um projeto de equipamentos de baixo custo e fácil utilização. A placa de aquisição de dados é o equipamento de maior valor, aproximadamente US\$ 400,00, uma vez que o microcomputador PC 486 100[Mhz] utilizado é atualmente de custo inferior.

Devido a ampla flexibilidade oferecida por esse tipo de controle, pode-se também utilizar o equipamento aqui descrito para controle de outros tipos de sistemas, inclusive utilizando redes neurais e controladores lógicos programáveis (CLPs), muito empregados atualmente na indústria moderna.

REFERÊNCIAS BIBLIOGRÁFICAS :

- [1] ADVANTECH Co., Ltda. “PCL-711B - PC-MultiLab User’s Manual”; Taiwan : Advantech Co., Ltd., aug., 1993.
- [2] Aranda, Gustavo E. I., “Projeto e implementação de um acionamento controlado analógico e digital para máquina de corrente contínua, utilizando chopper de quatro quadrantes”, Itajubá-MG: Escola Federal de Engenharia de Itajubá, 2000 (Dissertação, Mestrado em Engenharia Elétrica).
- [3] Assis W. de O. “Análise comparativa de estratégias de controle para acionamentos industriais utilizando chopper”, Itajubá: Escola Federal de Engenharia de Itajubá, 1997 (Dissertação, Mestrado em Engenharia Elétrica).
- [4] Buxbaum Arne: “Design of Control Systems for DC Drives”, EESSES – Electric Systems and Engineering Series, 1980.
- [5] Fröhr, F.; Orttenger, F.: “Introducción al Control Electrónico”, Siemens, Marcombo S.A., Barcelona, España, 1986.
- [6] Gonçalves Frederico: “Implementação Digital de Controladores Fuzzy”, Itajubá: Escola Federal de Engenharia de Itajubá, 1999 (Anais dos Trabalhos de Diploma, Graduação em Engenharia Elétrica).
- [7] Mendonça, G.R.S. “Circuito eletrônico para o comando de disparo de uma ponte trifásica tiristorizada”, Itajubá: Escola Federal de Engenharia de Itajubá, 2002 (Trabalho de Diploma, Graduação em Engenharia Elétrica).
- [8] Pinheiro, C.A.M.; “Análise e Projeto de Sistemas de Controle Fuzzy: Uma Abordagem no Domínio da Freqüência”, Campinas-SP, Unicamp, 2000 (Tese de Doutorado)

[9] Rezek, A.J.J.; Rodrigues, M.S.; Miranda, V.A.M.; Oliveira, V.A.; Cassula, A.M.; Júnior, R.A.C.; Torres, A.Z.: "Projeto e Simulação de um Sistema de Acionamento Controlado para Máquina de Corrente Contínua", Anais II SIMEAR, Vol. III, EPUSP, São Paulo/SP, 1991, pp. 141-160.

[10] Rosa, P. C. " Implementação de reguladores PID - digitais em sistemas com microprocessadores", Itajubá: Escola Federal de Engenharia de Itajubá, 1989. 87p. (Dissertação, Mestrado em Engenharia Elétrica).

ANEXO A – CIRCUITO ELETRÔNICO PARA O COMANDO DE DISPARO DE UMA PONTE TRIFÁSICA TIRISTORIZADA

I. INTRODUÇÃO

O circuito de controle, conhecido como circuito de baixa potência, gera toda a lógica de formação dos sinais de comando que, após a amplificação, irão disparar os tiristores.

A função do circuito de comando é a de fornecer pulsos de gatilho para tiristores e corrente de disparo (gate) com formas e valores adequados, em instantes especificados.

O componente principal do circuito apresentado neste artigo é o circuito integrado TCA 780. Esse integrado foi desenvolvido pela ICOTRON / SIEMENS para controlar o ângulo de disparo de tiristores, transistores e triacs, continuamente de 0° a 180° , com grande número de opções e utilizando pouquíssimos componentes externos.

A figura 1 mostra o diagrama em blocos da organização básica de um circuito de comando.

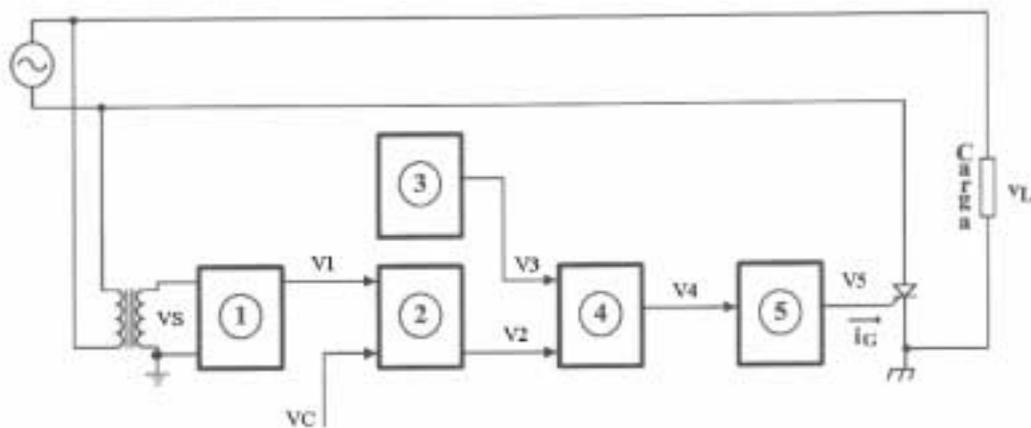


Figura 1 – Organização básica do circuito de comando.

Na figura 1 temos:

- 1 - sincronismo e geração da dente de serra;
- 2 - comprador;
- 3 - oscilador;
- 4 - porta lógica E;
- 5 - amplificação, isolamento e ataque;
- VS - tensão de sincronismo;
- V1 - onda dente de serra;
- VC - tensão de comando;
- V3 - onda retangular;
- V2 - pulso retangular;

V4 - trem de pulso retangular;
V5 - trem de pulso retangular;
IG - corrente de gate.

Entre os circuitos de comando e o de potência, que é formado pela ponte tiristorizada, a carga a ser alimentada e a rede de alimentação, é preciso manter um isolamento elétrico. Isso é feito acoplando-se os estágios 4 e 5, da figura 1, com transformadores de pulso ou por acopladores óticos. No circuito deste artigo foi utilizado o optoacoplador TIL 113 da TEXAS INSTRUMENTS.

Além do optoacoplador, é necessário que os estágios 4 e 5, da figura 1, sejam alimentados por fontes de tensão independentes, provendo assim um perfeito isolamento.

Com a utilização do TCA 780 e o uso de pulso retangular simples no lugar de um trem de pulsos retangulares, aplicados no gate dos tiristores, obtém-se uma redução no número de componentes, já que as funções dos estágios 1 e 2 são assumidas pelo TCA 780 e os estágios 3 e 4 são substituídos por um 555 configurado como monoestável.

A figura 2 mostra o diagrama em blocos do circuito de comando com a utilização dos integrados TCA 780 e 555.

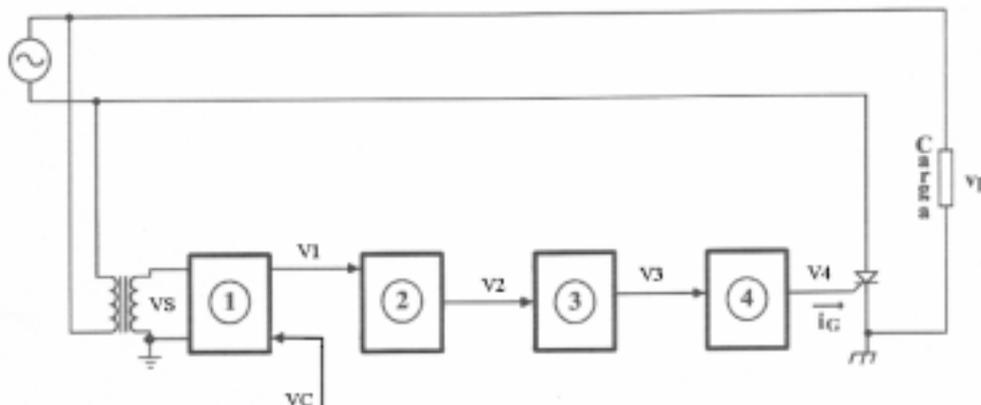


Figura 2 – Diagrama em blocos do circuito de comando utilizando o TCA 780 e o 555 como monoestável.

Na figura 2 temos:

- 1 - TCA 780;
- 2 - monoestável com 555;
- 3 - optoacoplador TIL 113;
- 4 - amplificação, isolamento e ataque;
- VS - tensão de sincronismo;
- V1 - pulso retangular;
- V2, V3 e V4 - pulso retangular “alargado”;

II. DESCRIÇÃO DOS ESTÁGIOS

Neste item serão mostradas as principais características dos circuitos integrados utilizados e suas funções dentro do circuito de comando.

II.1 – Geração dos pulsos (TCA 780)

O TCA 780 é dedicado à aplicação de controle de ângulo de disparo de tiristores, triacs e transistores continuamente de 0° a 180°. Sua configuração interna possibilita uma simplificada seleção de componentes externos para chaveamento, sem tornar muito volumoso o circuito final.

II.1.1 – Características

As principais características deste circuito integrado são:

- Consumo interno de apenas 5 mA;
- Lógica digital altamente imune a ruídos;
- Duas saídas principais (corrente de 55 mA) e duas em coletor aberto (corrente de até 1,5 mA);
- Necessidade de apenas três TCA 780 em sistema trifásico;
- Duração dos pulsos de saída determinada pela colocação de um capacitor externo ;
- Saída de tensão regulada em 3,1 V ;
- Possibilidades de inibição simultânea de todas as saídas;
- Uma saída para controle de TRIACS.

As figuras 3 e 4 mostram respectivamente o diagrama interno e os sinais presentes nas saídas do TCA 780.

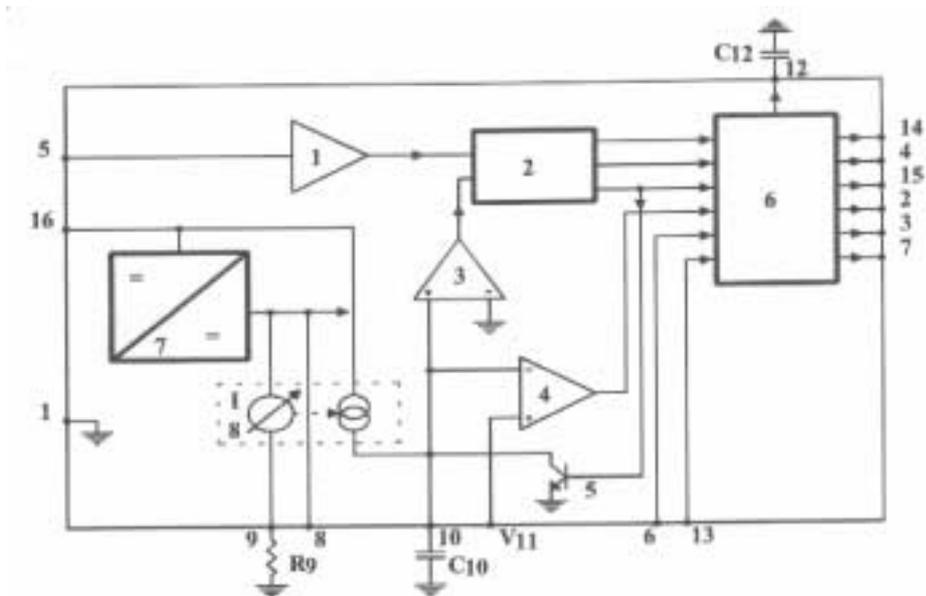


Figura 3 – Diagrama interno do TCA 780.

De acordo com a figura 3 , temos:

- 1 – Detetor de zero;
- 2 – Memória de sincronismo;
- 3 – Monitor de descarga de C10;
- 4 – Comparador de controle;
- 5 – Transistor de descarga;
- 6 – Unidade lógica;
- 7 – Regulador interno de tensão (3,1 V);
- 8 – Fonte de corrente constante.

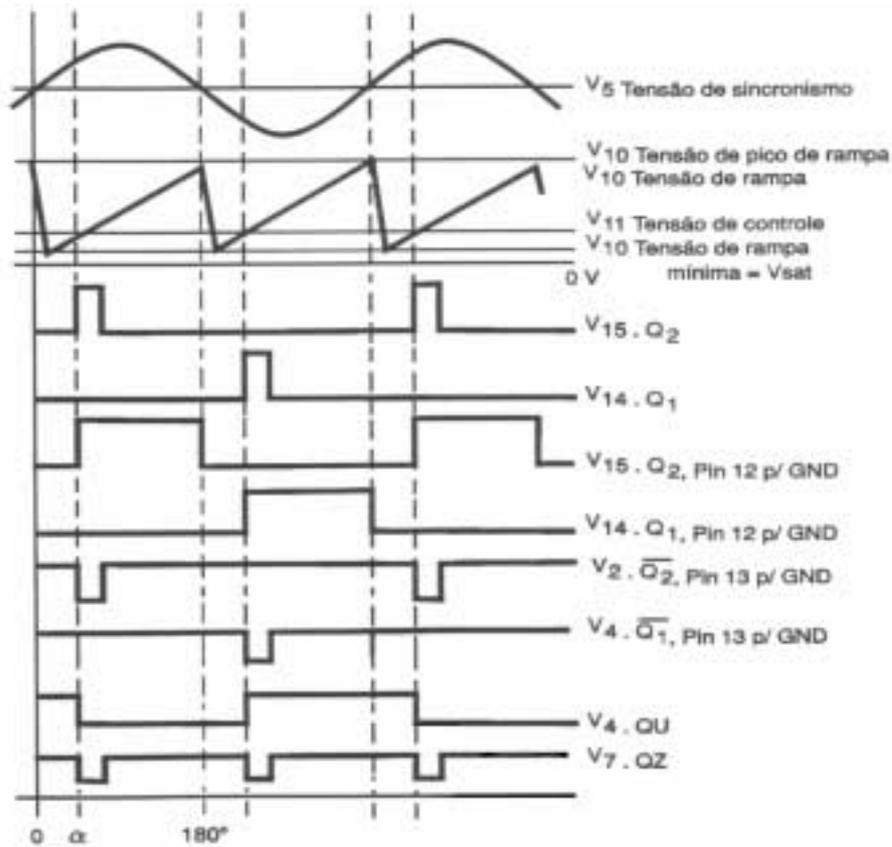


Figura 4 - Diagrama das formas de onda do TCA 780.

A figura 5 ilustra o invólucro e a pinagem do TCA 780

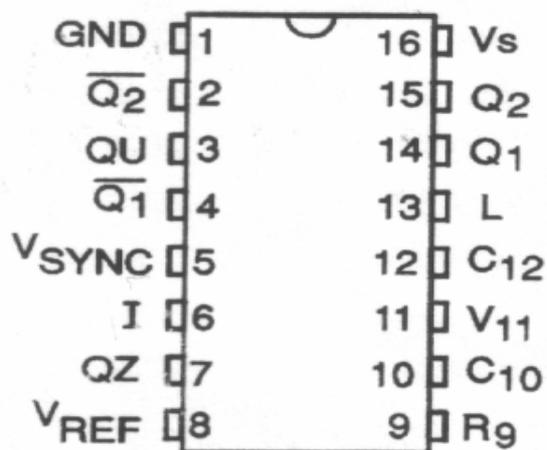


Figura 5 – Invólucro e pinagem do TCA 780.

As funções dos pinos são:

- 01 – Terra.
- 02 – Saída complementar do pino 15, em coletor aberto.
- 03 – Saída de pulso positivo, em coletor aberto.
- 04 – Saída complementar do pino 14, em coletor aberto.
- 05 – Entrada de sincronismo (diodos em antiparalelo).
- 06 – Inibe todas as saídas (quando aterrada).
- 07 – Saída em coletor aberto para acionar TRIACS.
- 08 – Fornece 3,1 V estabilizado.
- 09 – Potenciômetro de ajuste da rampa ($20 < R9 < 500k\Omega$).
- 10 – Capacitor de formação da rampa ($C10 \leq 0,5 \mu F$).
- 11 – Entrada da tensão de controle (nível CC).
- 12 – Controla a largura dos pulsos das saídas 14 e 15.
- 13 – Controla a largura dos pulsos das saídas 02 e 04.
- 14 – Saída de pulso positivo no semiciclo positivo.
- 15 – Saída de pulso positivo no semiciclo negativo.
- 16 – Alimentação CC, não necessariamente estabilizada.

II.1.2 – Funcionamento do TCA 780

A alimentação deste C.I. é feita pelo pino 16 em relação ao terminal terra, pino 1, com uma tensão de 8V a 18V. Internamente, o integrado é alimentado por uma tensão regulada de 3,1V, independente das variações possíveis em sua alimentação externa. O sincronismo é obtido através de um detetor de zero, pino 5, conectado a um registrador de sincronismo. O gerador de rampa, cujo controle está na unidade lógica, consiste de uma fonte de corrente constante que carrega linearmente um capacitor externo C10, corrente essa determinada por uma resistência externa R9, cuja finalidade é ajustar a amplitude da rampa, que vai a zero sempre que a tensão de sincronismo passa por zero. Assim, a combinação R9 C10 determina a inclinação da rampa. O comparador de controle compara a tensão de rampa com a tensão de controle; quando estas forem iguais envia pulsos nas saídas via unidade lógica. Obtém-se, então, V15, pino 15, pulsos positivos no semiciclo positivo da tensão de sincronismo, e V14, pino 14, pulsos positivos no semiciclo negativo da tensão de sincronismo, defasados entre si de 180°. Esses pulsos têm suas larguras determinadas pela conexão de um capacitor externo, C12, entre o pino 12 e o terra, de acordo com a tabela 1, e amplitudes iguais à tensão de alimentação do pino 16. Nos pinos 2 e 4

obtêm-se saídas complementares dos pinos 14 e 15, respectivamente, em coletor aberto, necessitando da ligação externa de um resistor entre os pinos 2 – 16 e 4 – 16 , proporcionando uma corrente máxima de 5mA. A largura dos pulsos pode ser controlada através da conexão de um resistor entre os pinos 13 – 16 . O pino 6 quando aterrado inibe todas as saídas do TCA 780.

C12 em pF	100	220	330	680	1000
Duração dos pulsos em ms	0,080	0,130	0,200	0,370	0,550

Tabela 1 – Larguras dos pulsos dos pinos 14 e 15 para valores determinados de C12

Para suprimir possíveis interferências , inclusive RF, é recomendável a instalação de capacitores cerâmicos e / ou tântalo nos pinos 8, 11 e 16.

C8 = 10nF , C11 = 100nF e C16 = 10µF + 10nF.

II.1.3 – Formulário básico

(a) Corrente de carga do capacitor C10

$$I_{10} = \frac{V_{ref} \times K}{R_9} \quad (1)$$

(b) Tensão da rampa

$$V_{10} = \frac{V_{ref} \times K \times \Delta t}{R_9 \times C_{10}} \quad (2)$$

(c) Ponto de disparo

$$t_z = \frac{R_9 \times C_{10}}{V_{ref} \times K} \times V_{11} \quad (3)$$

$$\alpha = \frac{V_{11}}{V_{10}} \times 180^\circ \quad (4)$$

(D) Largura de Pulso

$$t_p = 30\mu s \text{ SEM C12}$$

$$t_p = \frac{430\mu s}{nF} \text{ COM C12} \quad (5)$$

ONDE:

$$V_{REF} = V_8 = 3,1V ;$$

$$K = 1,25 ;$$

$$C_{10} \leq 0,5 \mu F ;$$

$$25 \text{ K}\Omega < R_9 < 500 \text{ K}\Omega$$

II.2 O Estágio de alargamento de pulso com o 555

O 555 é um C.I. que foi desenvolvido como uma unidade de finalidades gerais, capaz de operar em largas faixas, tanto como monoestável, como astável.

No circuito apresentado neste artigo, o 555 está operando como monoestável, proporcionando em sua saída um pulso de duração maior que o pulso fornecido em sua entrada, ou seja, ele possibilita um alargamento do pulso enviado do TCA 780. Através do potenciômetro de 50 K Ω conectado entre o Vcc da fonte e o pino 6 do 555, ajusta-se a largura do pulso de saída do 555 para 120°. As figuras 6a e 6b mostram respectivamente o 555 como monoestável e o mesmo utilizado no circuito deste artigo.

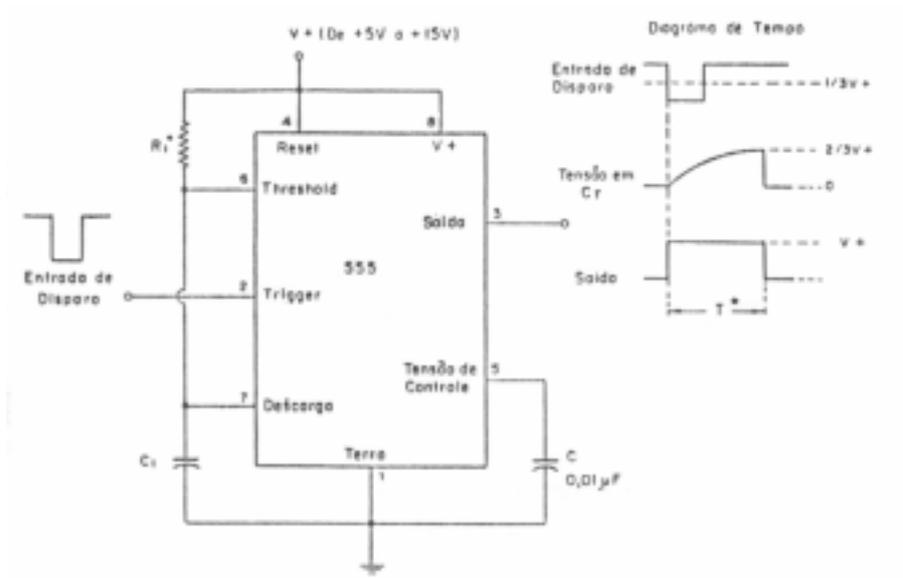


FIGURA 6A – 555 CONECTADO COMO MONOESTÁVEL.

PARA O 555 DA FIGURA 6A TEMOS:

$$T = 1,1.R1.C1$$

FAIXA DE R1 : 10 KΩ A 14 MΩ

FAIXA DE C1 : 100 pF A 1000 μF

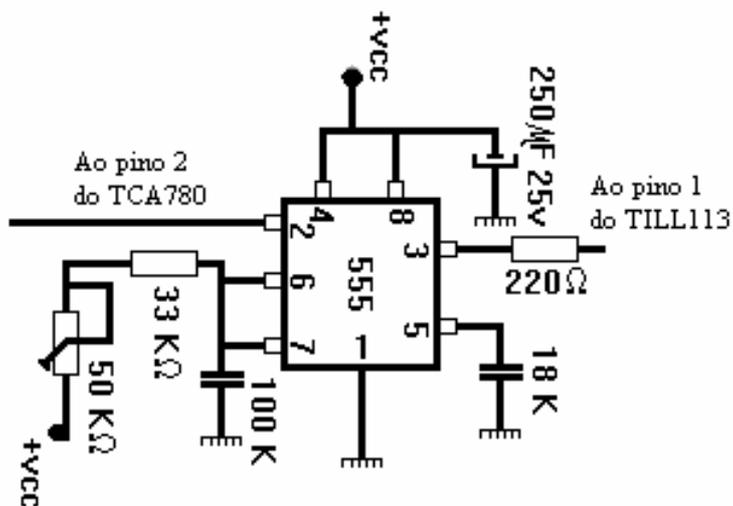


FIGURA 6B – MONOESTÁVEL COM LARGURA DE PULSO DE SAÍDA AJUSTÁVEL.

II.3 ESTÁGIO DE ACOPLAMENTO COM TIL113

A função deste estágio é a de transmitir o pulso para o estágio seguinte e, ao mesmo tempo, prover o isolamento elétrico entre os circuitos de alta e baixa potência. O TIL113 é um optoacoplador que apresenta um LED emissor de infravermelho, entre os pinos 1 e 2, e um fototransistor NPN Darlington nos pinos 4,5 e 6.

A figura 7 ilustra o diagrama interno do TIL113.

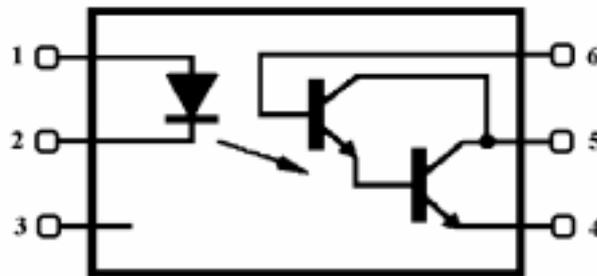


Figura 7- Diagrama interno do TIL113.

É preciso ressaltar que para se ter um perfeito isolamento elétrico, de modo a impedir qualquer caminho elétrico possível para uma corrente de curto, entre os estágios de alta e baixa potência, tanto o diodo como o fototransistor do TIL113 devem ser alimentados por fontes de tensão independentes.

II.4 ESTÁGIO DE ATAQUE (DRIVE)

Esta última parte do circuito tem a função de fornecer, ao pulso de saída do TIL113, a capacidade de corrente necessária para o disparo do gate do tiristor, possibilitando a sua condução. Assim, temos o transistor BC 548 funcionando como chave e em sua base há um capacitor “SPEED UP” que tem a função de acelerar o chaveamento do transistor. O BD 135, configurado como seguidor de emissor, fornece a capacidade de corrente e a tensão de gate, ajustada pelo potenciômetro na sua base, para o pulso de disparo, acoplando-o ao gate do tiristor através de um diodo, onde este impede que tensões negativas cheguem ao gate. Resumindo, as características gerais do estágio de ataque são:

- Amplificar os sinais de comando oriundos dos estágios de sinais;
- Apresentar características de fonte de corrente e não de tensão;
- Impedir que uma tensão negativa seja aplicada na junção gatilho – catodo.

A figura 8 ilustra o circuito do estágio de ataque .

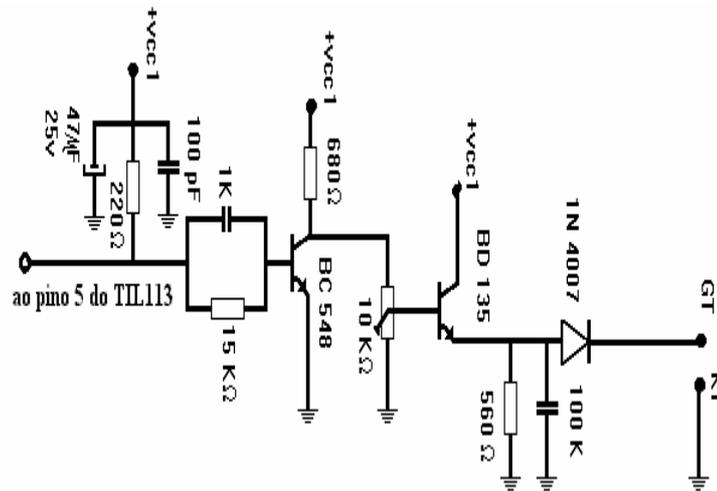


Figura 8 – Circuito do estágio de ataque.

II.5 Tensão de comando

Este estágio fornece a tensão de comando para os três TCA 780 que compõem o circuito de controle de disparo. Tal estágio é basicamente formado por um amplificador operacional 741 configurado como seguidor de tensão, ou seja, é um amplificador não inversor de ganho 1.

Os potenciômetros P1 e P2 determinam os ajustes máximo (180°) e mínimo (0°) para o ângulo de condução. O potenciômetro P3 determina o ângulo de condução dentro da faixa ajustada por P1 e P2.

A figura 9 ilustra o circuito da tensão de controle com o 741.

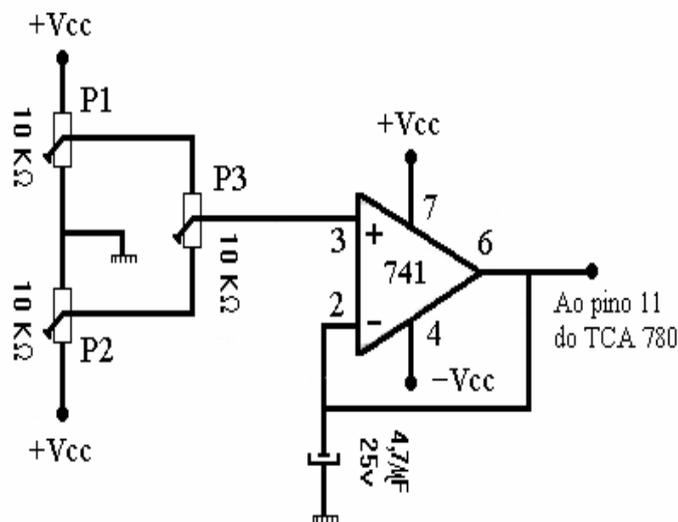


Figura 9 – Circuito da tensão de controle.

III . VISÃO GERAL

Aqui estão ilustrados o diagrama em blocos e o diagrama elétrico do circuito de comando, bem como o desenho da placa de circuito impresso com a disposição dos componentes. A figura 10 mostra o diagrama em blocos.

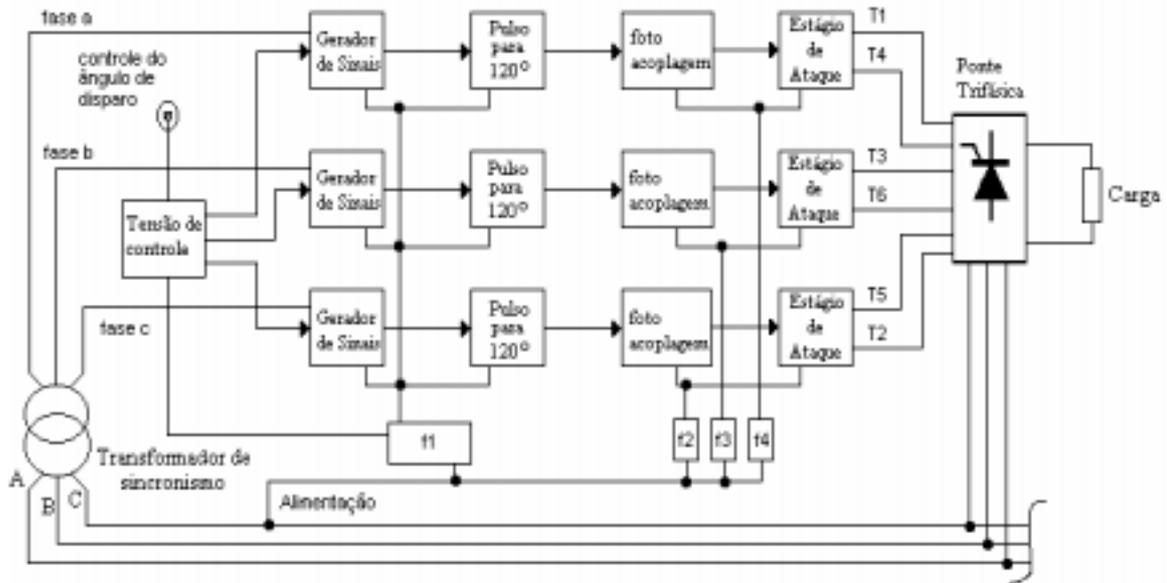
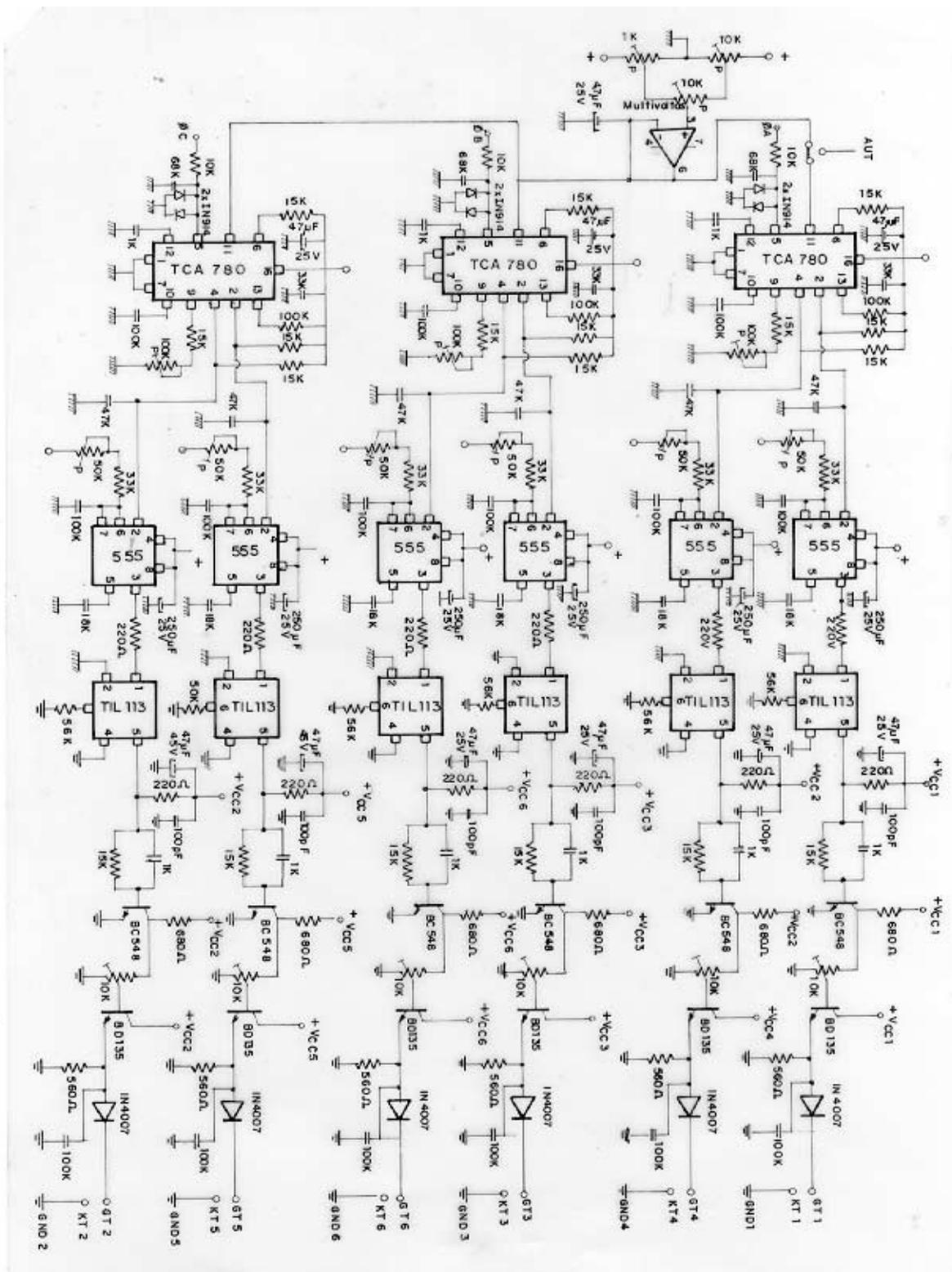


Figura 10 – Diagrama em blocos completo do circuito de comando.

CIRCUITO DE DISPARO PARA UMA PONTE CONVERSORA GRAETZ



ANEXO B :

PROGRAMA PARA O ACIONAMENTO CONTROLADO DE UMA MÁQUINA CC UTILIZANDO REGULADORES PI (PROPORCIONAL-INTEGRAL) :

```
/*
*****
*   Programa       : MCC Control                               *
*   Descrição     : Controle digital para acionamento de uma máquina de *
*                   corrente contínua utilizando a placa PCL-711B          *
*   Versão        : 2.1                                       *
*   Data         : 11/09/2000                                  *
*****
*/

/* Inclusão de Diretivas */

#include <stdio.h>
#include <conio.h>          /* Aceita diretivas, incluindo códigos */
#include <stdlib.h>        /* de fontes de outros */
#include <dos.h>           /* programas ou diretórios. */
#include <timer.h>

/* Declaração de Variáveis Globais */

extern "C" pcl711(int, unsigned int *); /* Inclui função "pcl711" definida
                                         inteiro, sem sinal em um módulo
                                         separado utilizando linguagem "C" */

unsigned int param[60]; /* Definição de um vetor de dados -
                         array - que formam a tabela de
                         parâmetros inteiros e s/ sinal */

unsigned int datain[200], dataout[200]; /* Buffer de 10 dados inteiros +
                                         para conversão */

unsigned int far * datin, * datout; /* Endereço do buffer de dados acima
                                     - pointer - tipo inteiro e longo:
                                     2 palavras c/ range de 1Mbyte */

int tecla,i; /* Variáveis de leitura do teclado
              e número de canais */

/* Variáveis pontos flutuantes do controle */

float nRef=0.72, nReal=0.0, ne=nRef;
float ne_1=0.1, nReal_1=0.0;
float iRef=1.20, iR=1.20, ie=-1.20, iReal=0.0, llim=1.20, iRefer=1.20;
float iRef_1=1.20, iR_1=1.20, ie_1=-1.20, iReal_1=0.0, iRefer_1=1.20;
float Tn=0.487, Tgs2=0.022, Ti=0.021;
float VRn=4.50, VRi=1.0, Vcc=0.1, Vcc_1=0.1, Vcontr=0.1;
float a1, a2, b1, b2, b3, b4, T=0.003;
float DataBuf[3];

/* Declaração de variáveis void - significa que não retorna um valor */

void conv_ad(void);
void conv_da(void);
void control(void);
void control1(void);
void control2(void);
void teclado(void);
```

```
/* Conversão AD - Tabela de parâmetros */
```

```
void conv_ad()
{
    unsigned int i;
    /* Pointer - Espaço de memória - Variável que contém um endereço que,
       normalmente , é o endereço de outra variável." */
    datin = datain; /* Atribui ao pointer datin o valor
                     equivalente à variável datain */
    param[0] = 0; /* Número da placa */
    param[1] = 0x220; /* Endereço de Base I/O */
    /* Freqüência de amostragem = Freqüência de base da placa/(C1 * C2) */
    /* 2M / (10 * 10) = 20 KHz */
    param[5] = 10; /* Divisor constante pacer C1 */
    param[6] = 10; /* Divisor constante pacer C2 */
    param[7] = 0; /* Modo Trigger, 0 : pacer trigger
                  Permite funções D/I */
    /* Offset do Buffer , o endereço de memória (buffer) onde os dados
       serão guardados. Segmento, o comprimento do buffer de dados */
    param[10] = FP_OFF(datin); /* Offset do Buffer A do A/D */
    param[11] = FP_SEG(datin); /* Segmento do Buffer A do A/D */
    param[12] = 0; /* Endereço do Buffer B (não usado)*/
    param[13] = 0; /* Segmento- Não usado, setar em 0 */
    /* A conversão A/D envolve dois canais de entrada, canal 1 - corrente,
       e canal 0 - velocidade, com valores em pu ajustados em +/- 5 V */
    param[14] = 2; /* Número de conversões A/D */
    param[15] = 0; /* Canal de início da conversão A/D*/
    param[16] = 1; /* Canal de parada da conversão A/D*/
    param[17] = 0; /* Ganho dos canais, 0 : +/- 5V */
}
```

```
/* Indicação de falha na conversão A/D */
```

```
pcl711(3, param); /* Func. 3 : Inicialização do Hardware */
if (param[45] != 0) { /* Se parâmetro 45 diferente de 0, fazer: */
    clrscr(); /* Limpar a tela */
    printf("\n FALHA NA INICIALIZAÇÃO DO DRIVER !"); /* Imprimir */
    getch(); /* Mostrar a tela de saída */
    exit(1); /* Fecha o loop e sai com status 1 - Erro */
}
```

```
pcl711(4, param); /* Func 4 : Inicialização do conversor A/D*/
if (param[45] != 0) {
    clrscr();
    printf("\n FALHA NA INICIALIZAÇÃO DO A/D !");
    getch();
    exit(1);
}
```

```
pcl711(5, param); /* Func 5 : Verificação número conversões A/D*/
if (param[45] != 0) {
    clrscr();
    printf("\n FALHA NO SOFTWARE DE TRANSFERÊNCIA DE DADOS A/D !");
    getch();
    exit(1);
}
```

```

/* Conversões A/D */

for (i = 0; i < param[14]; i++)      /* Dados amostrados - canais 0 e 1 */
{
    DataBuf[i] = datain[i] & 0xFFFF;
    /* Coleta de dados para o buffer no endereço 0xFFFF
    (os três primeiros dígitos hexadecimais podem ser zerados pois o
    restante, é suficiente para suportar 4096 dígitos binários ) */
    DataBuf[i] = ((5.0 - (-5)) * DataBuf[i] / 4096) + (-5);
    /* Conversão para que o sinal de tensão seja disponível para
    aplicação nas equações recursivas de controle */
    (5 - (-5)) : Faixa de entrada A/D (-5V to 5V)
    4096       : Faixa da escala do A/D - 12 bit
    DataBuf    : Dado de entrada do A/D
    (-5)       : Início da escala do A/D "-5" V
    */
}
/* Leitura da tensão de realimentação para a malha de velocidade
e de corrente, sob condições de velocidade e carga nominal */
/* Conversão do sinal de velocidade - correção para pu */
nReal=(DataBuf[0]/1.500);
/* Conversão do sinal de corrente - correção para pu */
iReal=(DataBuf[1]/1.493);
}

/* Controle */

/* Equações Recursivas para efetuar funções de controle */

void control()
{
    /* Adotado tempo de amostragem T */

    a1=T/((2*Tgs2)+T);
    /*Tgs2= Constante de tempo do filtro */
    a2=((2*Tgs2)-T)/(T+(2*Tgs2));

    b1=VRn+((VRn*T)/(2*Tn));
    /* VRn= Ganho do regulador de velocidade */
    /* Tn= Constante de tempo do reg. velocidade */
    b2=((VRn*T)/(2*Tn))-VRn;

    b3=VRi+((VRi*T)/(2*Ti));
    /* VRi= Ganho do regulador de corrente */
    /* Ti= Constante de tempo do reg. de corrente */
    b4=((VRi*T)/(2*Ti))-VRi;
}

/* Regulador de Velocidade */

void control1()
{
    iRef=(b1*ne)+(b2*ne_1)+iRef_1;
    /* nreal= Realimentação de velocidade */
    /* nRef= Velocidade referência após filtro */
    iRef_1=iRef;
    /* ne= Erro de velocidade */
    ne_1=ne;
    /* iRef= Corrente referência - Saída
    regulador de velocidade */
}

```

```

/* Filtro da Corrente de Referência */

iRefer=iRef;
    iR=a1*(iRefer+iRefer_1)+a2*iR_1; /* Sendo: */
    iR_1=iR; /* iR= Corrente referência após filtro */
    iRefer_1=iRefer; /* iRef= Corrente referência */
    if(iR>=Ilim) iR=Ilim;
    if(iR<=-Ilim) iR=-Ilim; /* Limitação da corrente de referência */
}

/* Regulador de Corrente */
void control2()
{
    Vcc=(b3*ie)+(b4*ie_1)+Vcc_1; /* Sendo: */
    Vcc_1=Vcc; /* ireal= realimentação de corrente */
    ie_1=ie; /* iR= Corrente ref. após filtro */
    if(Vcc>0.95) Vcontr=0.95; /* ie= Erro corrente- Ent. regulador */
    if(Vcc<.05) Vcontr=.05; /* Limitação da tensão de controle */
    if(Vcc<=0.95 && Vcc>=.05) Vcontr=Vcc;
}

/* Alteração nos parâmetros do sistema */

void teclado()
{
/* O acionamento destas teclas de subrotina, permitem ajuste on-line de
parâmetros do sistema, com o ajuste sendo mostrado na tela de saída */

    /* Teclas "s" e "d" atuando na valor da velocidade de referência */
    if (tecla==115 && nRef<1.0) nRef=nRef+1.44; /* Ajuste limitado ao */
    if (tecla==100 && nRef>-1.0) nRef=nRef-1.44; /* intervalo -1.0<nRef<1.0 */

    /* Teclas "k" e "l" atuando na valor da velocidade de referência */
    if (tecla==107 && nRef<1.0) nRef=nRef+0.01; /* Ajuste limitado ao */
    if (tecla==108 && nRef>-1.0) nRef=nRef-0.01; /* intervalo -1.0<nRef<1.0 */

    /* Teclas "o" e "p" atuando na valor da velocidade de referência */
    if (tecla==111 && nRef<1.0) nRef=nRef+0.05; /* Ajuste limitado ao */
    if (tecla==112 && nRef>-1.0) nRef=nRef-0.05; /* intervalo -1.0<nRef<1.0 */

    /* Teclas "f" e "v" atuando no ganho VRn */
    if (tecla==102 && VRn<40.0) VRn=VRn+0.05;
    if (tecla==118 && VRn>0.02) VRn=VRn-0.05;

    /* Teclas "g" e "b" atuando na constante de tempo Tn */
    if (tecla==103 && Tn<5.00) Tn=Tn+0.05;
    if (tecla==98 && Tn>0.06) Tn=Tn-0.05;

    /* Teclas "h" e "n" atuando no ganho VRi */
    if (tecla==104 && VRi<10.0) VRi=VRi+0.05;
    if (tecla==110 && VRi>0.01) VRi=VRi-0.05;

    /* Teclas "j" e "m" atuando na constante de tempo Ti */
    if (tecla==106 && Ti<1.000) Ti=Ti+0.005;
    if (tecla==109 && Ti>0.006) Ti=Ti-0.005;
}

```

```
/* Conversão D/A - Tabela de Parâmetros */
```

```
void conv_da()
```

```
{  
    datout=dataout;          /* Atribui ao pointer datout o valor  
                             equivalente à variável dataout */  
    param[0]=0;             /* Número da placa */  
    param[1]=0x220;        /* Endereço de base I/O */  
    /* Offset do Buffer , é o endereço de memória (buffer) onde os dados  
    serão guardados. Segmento , é o comprimento do buffer de dados */  
    param[20] = FP_OFF(datout); /* Offset do buffer A dados saída D/A */  
    param[21] = FP_SEG(datout); /* Segmento do buffer A dados saída D/A */  
    param[22] = 0;          /* Endereço do Buffer B saída (não usado) */  
    param[23] = 0;          /* Segmento saída - Não usado, setar em 0 */  
    param[24] = 1;          /* Número de conversões D/A */  
    param[25] = 0;          /* Canal de início da conversão D/A */  
    param[26] = 0;          /* Canal de parada da conversão D/A */
```

```
/* Indicação de falha na conversão D/A */
```

```
    pcl711(3, param); /* Func 3 : Inicialização do hardware */  
    if (param[45] != 0) { /* Se parâmetro 45 diferente de 0, fazer: */  
        clrscr(); /* Limpar a tela */  
        printf("\n FALHA NA INICIALIZAÇÃO DO DRIVER !"); /* Imprimir */  
        getch(); /* Mostrar a tela de saída */  
        exit(1); /* Fecha o loop e sai com status 1 - Erro */  
    }  
    pcl711(12, param); /* Func 12: Inicialização do conversor D/A */  
    if (param[45] != 0) {  
        clrscr();  
        printf("\n FALHA NA INICIALIZAÇÃO DO D/A !");  
        getch();  
        exit(1);  
    }  
    pcl711(13, param); /* Func 13: Verificação número conversões D/A*/  
    if (param[45] != 0) {  
        clrscr();  
        printf("\n FALHA NO SOFTWARE DE TRANSFERÊNCIA DE DADOS D/A !");  
        getch();  
        exit(1);  
    }  
    /* Conversão para que o sinal de tensão de saída das equações  
    recursivas de controle em pu ocupe um espaço de endereço do  
    buffer de dados de saída. */
```

```
    dataout[0]=(4095*Vcontr);  
}
```

```
/* Programa principal */
```

```
void main(void) /* Garante que as variáveis globais não retornam valores */  
{ /* Declaração de variáveis */  
    int xx=0; /* Inicializa o número de interações em 0 */  
    Timer t; /* Contador de tempo "t" */  
  
    clrscr(); /* Limpa a tela */  
    gotoxy(1,10); cprintf("[S] - Inverte vel. [+]");
```

```

gotoxy(40,10); cprintf("[D] - Inverte vel. [-]");
gotoxy(1,11); cprintf("[O] - Incrementa vel. [0.05]");
gotoxy(40,11); cprintf("[P] - Decrementa vel. [0.05]");
gotoxy(1,12); cprintf("[K] - Incrementa vel. [0.01]");
gotoxy(40,12); cprintf("[L] - Decrementa vel. [0.01]");

do{

    tecla=0;
    if (kbhit()) tecla=getch();

    xx++;          /* Incrementa 1 no número de interações */
    t.reset();     /* Reseta o temporizador "t" */
    t.start();     /* Inicia a temporização */

    asm cli;
    conv_ad();    /* Efetua a subrotina de conversão A/D */
    asm sti;

    control();
    teclado();   /* Efetua a subrotina que inspeciona o acionamento de tecla */

    ne=nRef-nReal;
    if(iR<llim && iR>-llim) control1();
    if(iR==llim && ne<0.0) control1();
    if(iR==-llim && ne>0.0) control1();

    ie=iReal-iR;
    if(Vcc<0.95 && Vcc>.05)control2();
    if(Vcc>=0.95 && ie<0.0 ) control2();
    if(Vcc<=.05 && ie>0.0 ) control2();

    asm cli;
    conv_da();   /* Efetua a subrotina de conversão D/A */
    asm sti;

    if (xx>=10) {/* Imprime na tela resultados instantâneos cada 100 interações */
        xx=0;
        gotoxy(1,1);
        cprintf("VRn= %1.3f Tn= %1.4f VRi= %1.3f Ti= %1.4f      ", VRn, Tn, VRi, Ti);
        gotoxy(1,2);
        cprintf("Tempo gasto: %fs      ", T);
        gotoxy(1,3);
        cprintf("canal[%3d] = % 1.3f (pu) canal[%3d] = % 1.3f (pu)      ", 0, nReal, 1, iReal);
        gotoxy(1,4);
        cprintf("nRef= %1.2f Vcontr= %1.2f iR= %1.4f ie= %1.2f      ", nRef, Vcontr, iR, ie);
        gotoxy(1,5);
        cprintf("Said.R.Veloc= %1.3f Said.R.Corr= %1.3f      ", iRef, Vcc);
        /* %1.4f V, significa, real, em 1 campo, com 4 dígitos decimais, em "V".
        %3d significa, inteiro, em 3 campos. */
    }
    t.stop();     /* Termina a temporização */
    T=t.time();  /* Consideramos o tempo de amostragem c/ valor inicial
    T=0.003 s. Se o tempo de execução do programa "t" , maior que "T", necessário
    tentar reduzi-lo. Se "t" e menor fazemos T=t.*/
    }
    while (tecla!=27);
}

```

PROGRAMA PARA O ACIONAMENTO CONTROLADO DE UMA MÁQUINA CC UTILIZANDO REGULADORES COM LÓGICA FUZZY (PI-FUZZY):

```
/*
*****
* Programa      : Controlador PI Fuzzy          *
* Descrição    : Controle digital para acionamento de uma máquina de      *
* corrente contínua utilizando o cartão PCL-711B *
* Versão       : 2                             *
* Data        : 25/07/2001                    *
*****
*/

#include <stdio.h>
#include <conio.h>      /* Aceita diretivas, incluindo códigos */
#include <stdlib.h>    /* de fontes de outros */
#include <dos.h>       /* programas ou diretórios. */
#include <timer.h>     /* temporização do programa */
#include <math.h>      /* funções matemáticas */

/* Declaração de Variáveis Globais */

extern "C" pcl711(int, unsigned int *); /* Inclui função "pcl711" definida
                                         inteiro, sem sinal em um módulo
                                         separado utilizando linguagem "C"
                                         */

unsigned int param[60]; /* Definição de um vetor de dados -array - que formam a tabela de
parâmetros inteiros e s/ sinal */

unsigned int datain[200], dataout[200]; /* Buffer de 10 dados inteiros +
                                         para conversão */

unsigned int far * datin, * datout;
/* Endereço do buffer de dados acima
- pointer - tipo inteiro e longo:
2 palavras c/ range de 1Mbyte */

int tecla,i, cont1, cont2=0; /* Variáveis de leitura do teclado
e numero de canais */

/* Variáveis e Pontos flutuantes do controle */

float nRef= 1.00, nReal=0.0, ne=nRef, iRef, iR,ie, iReal=0.0 , ilim=1.2;
float a1, a2, b1, b2, b3, b4;
float DataBuf[3];
float Vc1, e, v , Vi1=0.0, j, dd1 ,u1 , Gv1 = 5.0, Ge1=5.00, dt=0.003;
float
ceNL=-1.2, ceNS=-1.0, ceZE= 0.0, cePS= 1.0, cePL= 1.2,
beNL= 1.0, beNS= 1.0, beZE= 1.0, bePS= 1.0, bePL= 1.0,
cvNL=-1.2, cvNS=-1.0, cvZE= 0.0, cvPS= 1.0, cvPL= 1.2,
bvNL= 1.0, bvNS= 1.0, bvZE= 1.0, bvPS= 1.0, bvPL= 1.0,
caNL=-1.2, caNS=-1.0, caZE= 0.0, caPS= 1.0, caPL= 1.2,
ueNL, ueNS, ueZE, uePS, uePL,
uvNL, uvNS, uvZE, uvPS, uvPL;
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;
```

```

float Vc2,e2,v2, Vi2=0.0, dd2, Gv2 = 50.0 , Ge2 = 0.10 , u2 ;
float
ceNL2=-1.5, ceNS2=-1.0, ceZE2= 0.0, cePS2= 1.0, cePL2= 1.5,
beNL2= 1.0, beNS2= 1.0, beZE2= 1.0, bePS2= 1.0, bePL2= 1.0,
cvNL2=-1.5, cvNS2=-1.0, cvZE2= 0.0, cvPS2= 1.0, cvPL2= 1.5,
bvNL2= 1.0, bvNS2= 1.0, bvZE2= 1.0, bvPS2= 1.0, bvPL2= 1.0,
caNL2=-1.5, caNS2=-1.0, caZE2= 0.0, caPS2= 1.0, caPL2= 1.5,
ueNL2, ueNS2, ueZE2, uePS2, uePL2,
uvNL2, uvNS2, uvZE2, uvPS2, uvPL2;
float r26, r27, r28, r29, r30, r31, r32, r33, r34, r35, r36, r37, r38, r39, r40, r41,
r42, r43, r44, r45, r46, r47, r48, r49, r50;

```

/ Declaração de variáveis void - significa que não retorna um valor */*

```

void conv_ad(void);
void conv_da(void);
void control1(void);
void control2(void);
void teclado(void);

```

/ Sub-rotina para Conversão Analógica Digital */*

```

void conv_ad()
{
    unsigned int i;
    /* Pointer - Espaço de memória - Variável que contém um endereço que,*/
    /* normalmente é o endereço de outra variável." */

    datin = datain;          /* Atribui ao pointer datin o valor
    equivalente ... variável datain */

    param[0] = 0;           /* Número do cartão */
    param[1] = 0x220;       /* Endereço de Base I/O */

    /* frequência de amostragem = frequência de base do cartão/(C1 * C2) */
    /* 2M / (10 * 10) = 20 kHz */

    param[5] = 10;         /* Divisor constante C1 */
    param[6] = 10;         /* Divisor constante C2 */
    param[7] = 0;          /* Modo Trigger, 0 : pacer trigger
    Permite funções D/I */

    /* Offset do Buffer , o endereço de memória (Buffer) onde os dados
    serão guardados. Segmento , o comprimento do Buffer de dados */

    param[10] = FP_OFF(datin); /* Offset do Buffer A do A/D */
    param[11] = FP_SEG(datin); /* Segmento do Buffer A do A/D */
    param[12] = 0;           /* Endereço do Buffer B (não usado)*/
    param[13] = 0;          /* Segmento- Não usado, setar em 0 */

    /* A conversão A/D envolve dois canais de entrada, canal 1 - corrente,e canal 0 - velocidade, com
    valores em pu ajustados em +/- 5 V */

    param[14] = 2;          /* Número de conversões A/D */
    param[15] = 0;          /* Canal de inicio da conversão A/D*/
    param[16] = 1;          /* Canal de parada da conversão A/D*/

```

```

param[17] = 0;          /* Ganho dos canais, 0 : +/- 5V */

/* indicação de falha na conversão A/D */

pcl711(3, param);     /* Func. 3 : Inicialização do Hardware */
if (param[45] != 0) { /* Se parâmetro 45 diferente de 0, fazer: */
    clrscr();         /* Limpar a tela */
    printf("\n FALHA NA INICIALIZAÇÃO DO DRIVER !"); /* Imprimir */
    getch();         /* Mostrar a tela de saída */
    exit(1);         /* Fecha o loop e sai com status 1 - Erro */
}

pcl711(4, param);     /* Func 4 : Inicialização do conversor A/D*/
if (param[45] != 0) {
    clrscr();
    printf("\n FALHA NA INICIALIZAÇÃO DO A/D !");
    getch();
    exit(1);
}

pcl711(5, param);     /* Func 5 : verificação número conversões A/D*/
if (param[45] != 0) {
    clrscr();
    printf("\n FALHA NO SOFTWARE DE TRANSFERÊNCIA DE DADOS A/D !");
    getch();
    exit(1);
}

/* Conversões A/D */

for (i = 0; i < param[14]; i++) /* Dados amostrados - canais 0 e 1 */
{
    DataBuf[i] = datain[i] & 0xFFFF;
    /* Coleta de dados para o Buffer no endereço 0xFFFF
    (os três primeiros dígitos hexadecimais podem ser zerados pois o
    restante , suficiente p/ suportar 4096 dígitos binários) */
    DataBuf[i] = ((5.0 - (-5)) * DataBuf[i] / 4096) + (-5);
    /* conversão para que o sinal de tensão seja disponível para
    aplicação nas equações recursivas de controle
    (5 - (-5)) : Faixa de entrada A/D (-5V to 5V)
    4096      : Faixa da escala do A/D - 12 bit
    DataBuf   : Dado de entrada do A/D
    (-5)      : Início da escala do A/D "-5" V
    */
}

/* Leitura da tensão de realimentação para a malha de velocidade
e de corrente, sob condições de velocidade e carga nominal */

/* conversão do sinal de velocidade - correção para pu */

nReal=(DataBuf[0]/1.500);

/* conversão do sinal de corrente - correção para pu */

iReal=(DataBuf[1]/1.493);
}

void control1()
{

```

```

ne = nRef - nReal ;          /* Erro */
Vi1 = Vi1 + Gv1*ne*dt;      /* Integral do Erro */
if (Vi1 < cvNL) Vi1 = cvNL ; /* Limites */
if (Vi1 > cvPL) Vi1 = cvPL ;
e = Ge1 * ne ;
v = Vi1 ;

teclado(); /* Efetua a subrotina que inspeciona o acionamento de tecla */

```

/* Fuzzificação : Funções Triangulares */

```

/* Analisando em Relação a "e" */

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

if ( ( ( ceNS - beNS ) <= e ) && ( e <= ( ceNS + beNS ) ) )
ueNS = 1.0 - ( fabs( ceNS - e ) ) / beNS ;
else ueNS = 0.0 ;

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

if ( ( ( cePS - bePS ) <= e ) && ( e <= ( cePS + bePS ) ) )
uePS = 1.0 - ( fabs( cePS - e ) ) / bePS ;
else uePS = 0.0 ;

if ( ( ( cePL - bePL ) <= e ) && ( e <= ( cePL + bePL ) ) )
uePL = 1.0 - ( fabs( cePL - e ) ) / bePL ;
else uePL = 0.0 ;
if ( e > cePL )
uePL = 1.0 ;

```

/* Analisando em Relação a "v" */

```

if ( ( ( cvNL - bvNL ) <= v ) && ( v <= ( cvNL + bvNL ) ) )
uvNL = 1.0 - ( fabs( cvNL - v ) ) / bvNL ;
else uvNL = 0.0 ;
if ( v < cvNL )
uvNL = 1.0 ;

if ( ( ( cvNS - bvNS ) <= v ) && ( v <= ( cvNS + bvNS ) ) )
uvNS = 1.0 - ( fabs( cvNS - v ) ) / bvNS ;
else uvNS = 0.0 ;

if ( ( ( cvZE - bvZE ) <= v ) && ( v <= ( cvZE + bvZE ) ) )
uvZE = 1.0 - ( fabs( cvZE - v ) ) / bvZE ;
else uvZE = 0.0 ;

if ( ( ( cvPS - bvPS ) <= v ) && ( v <= ( cvPS + bvPS ) ) )
uvPS = 1.0 - ( fabs( cvPS - v ) ) / bvPS ;
else uvPS = 0.0 ;

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

```

```

else uvPL = 0.0 ;
if ( v > cvPL )
uvPL = 1.0 ;

```

```

/* Implicações Lógicas - Operador Produto */

```

```

r1 = ueNL * uvNL ;
r2 = ueNL * uvNS ;
r3 = ueNL * uvZE ;
r4 = ueNL * uvPS ;
r5 = ueNL * uvPL ;
r6 = ueNS * uvNL ;
r7 = ueNS * uvNS ;
r8 = ueNS * uvZE ;
r9 = ueNS * uvPS ;
r10 = ueNS * uvPL ;
r11 = ueZE * uvNL ;
r12 = ueZE * uvNS ;
r13 = ueZE * uvZE ;
r14 = ueZE * uvPS ;
r15 = ueZE * uvPL ;
r16 = uePS * uvNL ;
r17 = uePS * uvNS ;
r18 = uePS * uvZE ;
r19 = uePS * uvPS ;
r20 = uePS * uvPL ;
r21 = uePL * uvNL ;
r22 = uePL * uvNS ;
r23 = uePL * uvZE ;
r24 = uePL * uvPS ;
r25 = uePL * uvPL ;

```

```

/* Defuzzificação */

```

```

dd1 =
caNL*r1+caNL*r2+caNL*r3+caNS*r4+caZE*r5+caNL*r6+caNL*r7+caNS*r8+
caZE*r9+caPS*r10+caNL*r11+caNS*r12+caZE*r13+caPS*r14+caPL*r15+
caNS*r16+caZE*r17+caPS*r18+caPL*r19+caPL*r20+caZE*r21+caPS*r22+
caPL*r23+caPL*r24+caPL*r25;

```

```

u1 = dd1 /
(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);

```

```

//u = u/10.0 ;
if (u1 < -1.0) u1 = -ilim;
if (u1 > 1.0) u1 = ilim;

```

```

}

```

```

void control2()
{

```

```

/* Malha de Corrente */

```

```

if(iRef > ilim) iRef = ilim ;
if(iRef < -ilim) iRef = -ilim ;
ie = iReal - iRef ;          /* Erro */

```

```

Vi2 = Vi2 + Gv2*ie*dt;      /* Integral do Erro */

```

```

if (Vi2 < cvNL2) Vi2 = cvNL2 ; /* Limites */
if (Vi2 > cvPL2) Vi2 = cvPL2 ;
e2 = Ge2 * ie ;
v2 = Vi2 ;

```

```

/* Fuzzificação : Funções Triangulares */

```

```

/* Analisando em Relação a "e2" */

```

```

if ( ( ( ceNL2 - beNL2 ) <= e2 ) && ( e2 <= ( ceNL2 + beNL2 ) ) )
ueNL2 = 1.0 - ( fabs( ceNL2 - e2 ) ) / beNL2 ;
else ueNL2 = 0.0 ;
if ( e2 < ceNL2 )
ueNL2 = 1.0 ;

```

```

if ( ( ( ceNS2 - beNS2 ) <= e2 ) && ( e2 <= ( ceNS2 + beNS2 ) ) )
ueNS2 = 1.0 - ( fabs( ceNS2 - e2 ) ) / beNS2 ;
else ueNS2 = 0.0 ;

```

```

if ( ( ( ceZE2 - beZE2 ) <= e2 ) && ( e2 <= ( ceZE2 + beZE2 ) ) )
ueZE2 = 1.0 - ( fabs( ceZE2 - e2 ) ) / beZE2 ;
else ueZE2 = 0.0 ;

```

```

if ( ( ( cePS2 - bePS2 ) <= e2 ) && ( e2 <= ( cePS2 + bePS2 ) ) )
uePS2 = 1.0 - ( fabs( cePS2 - e2 ) ) / bePS2 ;
else uePS2 = 0.0 ;

```

```

if ( ( ( cePL2 - bePL2 ) <= e2 ) && ( e2 <= ( cePL2 + bePL2 ) ) )
uePL2 = 1.0 - ( fabs( cePL2 - e2 ) ) / bePL2 ;
else uePL2 = 0.0 ;
if ( e2 > cePL2 )
uePL2 = 1.0 ;

```

```

/* Analisando em Relação a "v2" */

```

```

if ( ( ( cvNL2 - bvNL2 ) <= v2 ) && ( v2 <= ( cvNL2 + bvNL2 ) ) )
uvNL2 = 1.0 - ( fabs( cvNL2 - v2 ) ) / bvNL2 ;
else uvNL2 = 0.0 ;
if ( v2 < cvNL2 )
uvNL2 = 1.0 ;

```

```

if ( ( ( cvNS2 - bvNS2 ) <= v2 ) && ( v2 <= ( cvNS2 + bvNS2 ) ) )
uvNS2 = 1.0 - ( fabs( cvNS2 - v2 ) ) / bvNS2 ;
else uvNS2 = 0.0 ;

```

```

if ( ( ( cvZE2 - bvZE2 ) <= v2 ) && ( v2 <= ( cvZE2 + bvZE2 ) ) )
uvZE2 = 1.0 - ( fabs( cvZE2 - v2 ) ) / bvZE2 ;
else uvZE2 = 0.0 ;

```

```

if ( ( ( cvPS2 - bvPS2 ) <= v2 ) && ( v2 <= ( cvPS2 + bvPS2 ) ) )
uvPS2 = 1.0 - ( fabs( cvPS2 - v2 ) ) / bvPS2 ;
else uvPS2 = 0.0 ;

```

```

if ( ( ( cvPL2 - bvPL2 ) <= v2 ) && ( v2 <= ( cvPL2 + bvPL2 ) ) )
uvPL2 = 1.0 - ( fabs( cvPL2 - v2 ) ) / bvPL2 ;
else uvPL2 = 0.0 ;
if ( v2 > cvPL2 )
uvPL2 = 1.0 ;

```

```
/* Implicações Lógicas - Operador Produto */
```

```
r26 = ueNL2 * uvNL2 ;  
r27 = ueNL2 * uvNS2 ;  
r28 = ueNL2 * uvZE2 ;  
r29 = ueNL2 * uvPS2 ;  
r30 = ueNL2 * uvPL2 ;  
r31 = ueNS2 * uvNL2 ;  
r32 = ueNS2 * uvNS2 ;  
r33 = ueNS2 * uvZE2 ;  
r34 = ueNS2 * uvPS2 ;  
r35 = ueNS2 * uvPL2 ;  
r36 = ueZE2 * uvNL2 ;  
r37 = ueZE2 * uvNS2 ;  
r38 = ueZE2 * uvZE2 ;  
r39 = ueZE2 * uvPS2 ;  
r40 = ueZE2 * uvPL2 ;  
r41 = uePS2 * uvNL2 ;  
r42 = uePS2 * uvNS2 ;  
r43 = uePS2 * uvZE2 ;  
r44 = uePS2 * uvPS2 ;  
r45 = uePS2 * uvPL2 ;  
r46 = uePL2 * uvNL2 ;  
r47 = uePL2 * uvNS2 ;  
r48 = uePL2 * uvZE2 ;  
r49 = uePL2 * uvPS2 ;  
r50 = uePL2 * uvPL2 ;
```

```
/* Defuzzificação */
```

```
dd2 =  
caNL2*r26+caNL2*r27+caNL2*r28+caNS2*r29+caZE2*r30+caNL2*r31+caNL2*r32+caNS2*r33+  
caZE2*r34+caPS2*r35+caNL2*r36+caNS2*r37+caZE2*r38+caPS2*r39+caPL2*r40+  
caNS2*r41+caZE2*r42+caPS2*r43+caPL2*r44+caPL2*r45+caZE2*r46+caPS2*r47+  
caPL2*r48+caPL2*r49+caPL2*r50;
```

```
u2 = dd2 /  
(r26+r27+r28+r29+r30+r31+r32+r33+r34+r35+r36+r37+r38+r39+r40+r41+r42+r43+r44+  
r45+r46+r47+r48+r49+r50);
```

```
//u = u/10.0 ;
```

```
if (u2 < 0.0) u2 = 0.0;  
if (u2 > 1.0) u2 = 1.0;
```

```
}
```

```
void teclado()  
{
```

```
/* O acionamento destas teclas de subrotina, permitem ajuste on-line de  
parâmetros do sistema, com o ajuste sendo mostrado na tela de saída */
```

```
/* Teclas "k" e "l" atuando na valor da velocidade de referência */
```

```
if (tecla==108 && nRef<1.0) nRef=nRef+0.01; /* Ajuste limitado ao */  
if (tecla==107 && nRef>-1.0) nRef=nRef-0.01; /* intervalo -1.0<nRef<1.0 */
```

```

/* Teclas "o" e "p" atuando na valor da velocidade de referência */

    if (tecla==112 && nRef<1.0) nRef=nRef+0.05; /* Ajuste limitado ao */
    if (tecla==111 && nRef>-1.0) nRef=nRef-0.05; /* intervalo -1.0<nRef<1.0 */

/* Teclas "q" e "w" atuando no ganho Gv1 */

    if (tecla==119 && Gv1<99.95) Gv1 = Gv1+0.05 ;
    if (tecla==113 && Gv1>0.05) Gv1 = Gv1-0.05 ;

/* Teclas "s" e "d" atuando no ganho Ge1 */

    if (tecla==100 && Ge1< 99.95) Ge1=Ge1+0.05;
    if (tecla==115 && Ge1> 0.05) Ge1=Ge1-0.05;

/* Teclas "g" e "h" atuando no ganho Gv2 */

    if (tecla==103 && Gv2< 9.95) Gv2=Gv2+0.05;
    if (tecla==104 && Gv2< 0.05) Gv2=Gv2-0.05;

/* Teclas "m" e "n" atuando no ganho Ge2 */

    if (tecla==109 && Ge2< 9.95) Ge2=Ge2+0.05;
    if (tecla==110 && Ge2> 0.05) Ge2=Ge2-0.05;
}

/* Sub-rotina para Conversão Digital-Analógica */

void conv_da()

{

    datout=dataout; /* Atribui ao pointer datout o valor
equivalente ... variável dataout */

    param[0]=0; /* Número do cartão */
    param[1]=0x220; /* Endereço de base I/O */

/* Offset do Buffer , o endereço de memória (Buffer) onde os dados
serão guardados. Segmento , o comprimento do Buffer de dados */

    param[20] = FP_OFF(datout); /* Offset do Buffer A dados saída D/A */
    param[21] = FP_SEG(datout); /* Segmento do Buffer A dados saída D/A */
    param[22] = 0; /* Endereço do Buffer B saída(não usado) */
    param[23] = 0; /* Segmento saída- Não usado, setar em 0 */
    param[24] = 1; /* Número de conversões D/A */
    param[25] = 0; /* Canal de inicio da Conversão D/A */
    param[26] = 0; /* Canal de parada da Conversão D/A */

/* Indicação de falha na Conversão D/A */

    pci711(3, param); /* Func 3 : Inicialização do hardware */
    if (param[45] != 0) { /* Se parâmetro 45 diferente de 0, fazer: */
        clrscr(); /* Limpar a tela */
        printf("\n FALHA NA INICIALIZAÇÃO DO DRIVER !"); /* Imprimir */
        getch(); /* Mostrar a tela de saída */
        exit(1); /* Fecha o loop e sai com status 1 - Erro */
    }
}

```

```

    }
    pci711(12, param); /* Func 12: Inicialização do conversor D/A */
    if (param[45] != 0) {
        clrscr();
        printf("\n FALHA NA INICIALIZAÇÃO DO D/A !");
        getch();
        exit(1);
    }
    pci711(13, param); /* Func 13: verificação número conversões D/A*/
    if (param[45] != 0) {
        clrscr();
        printf("\n FALHA NO SOFTWARE DE TRANSFERÊNCIA DE DADOS D/A !");
        getch();
        exit(1);
    }

/* conversão para que o sinal de tensão de saída das equações
recursivas de controle em pu ocupe um espaço de endereço do
Buffer de dados de saída. */
    dataout[0]=(4095*u2); /* Base pu = 4095 */
}

/* Programa principal */

void main(void)
{

    Timer t; /* Contador de tempo "t" */
    clrscr(); /* Limpa a Tela */
    delay(500);
    textbackground(4); /* Define Fundo Vermelho (4) */
    gotoxy(5,2);
    cprintf("CONTROLADOR FUZZY PARA CONTROLE DE VELOCIDADE DE UM MCC : );
    gotoxy(5,4);
    delay(500);
    cprintf(" Aluno : Otavio Henrique Salvi Vicentini n 9025 ");
    gotoxy(5,5);
    cprintf(" Orientador : prof. Angelo J. J. Rezek Data : 25/09/2000 ");

    delay(500);
    textbackground(1); /* Define Fundo Azul (1) */

    gotoxy(1,15); cprintf(" [O] - Diminui a Velocidade [0.05] ");
    gotoxy(40,15); cprintf(" [P] - Aumenta a Velocidade [0.05] ");
    gotoxy(1,16); cprintf(" [K] - Diminui a Velocidade [0.01] ");
    gotoxy(40,16); cprintf(" [L] - Aumenta a Velocidade [0.01] ");
    gotoxy(1,17); cprintf(" [Q] - Diminui o Ganho Gv1 [0.05] ");
    gotoxy(40,17); cprintf(" [W] - Aumenta o Ganho Gv2 [0.05] ");
    gotoxy(1,18); cprintf(" [S] - Diminui o Ganho Ge1 [0.05] ");
    gotoxy(40,18); cprintf(" [D] - Aumenta o Ganho Ge1 [0.05] ");
    gotoxy(1,19); cprintf(" [G] - Diminui o Ganho Gv2 [0.05] ");
    gotoxy(40,19); cprintf(" [H] - Aumenta o Ganho Gv2 [0.05] ");
    gotoxy(1,20); cprintf(" [M] - Diminui o Ganho Ge2 [0.05] ");
    gotoxy(40,20); cprintf(" [N] - Aumenta o Ganho Ge2 [0.05] ");

    textbackground(4); /* Define Fundo Vermelho (4) */
    gotoxy(13,24); cprintf(" PRECIONE ESC PARA FINALIZAR O PROGRAMA ");

```

```

do
{
    tecla=0;

    if (kbhit()) tecla=getch(); /* Controle por Teclado */

    j++;          /* Incrementa 1 no numero de interações */
    t.reset();    /* Reseta o temporizador "t" */
    t.start();    /* Inicia a temporização */

    asm cli;
    conv_ad();    /* Efetua a subrotina de conversão A/D */
    asm sti;

    ne = nRef-nReal;

    control1();
    if(iRef<ilim && iRef>-ilim) control1();
    if(iRef==ilim && ne<0.0) control1();
    if(iRef==-ilim && ne>0.0) control1();

    iRef = u1;
    ie=iReal-iRef;

    if(u2<0.9 && u2>.10) control2();
    if(u2>=0.9 && ie<0.0) control2();
    if(u2<=.10 && ie>0.0) control2();

    asm cli;
    conv_da(); /* Efetua a subrotina de Conversão D/A */
    asm sti;

    if (j>=100) /* Imprime na tela resultados instantâneos cada 100 interações */
    {
        j=0;
        gotoxy(7,8);
        textbackground(1);
        cprintf(" Tempo gasto : %1.6f (s)", dt);
        gotoxy(7,9);
        cprintf(" Canal [0] - Velocidade : % 1.3f (pu) Gv1 %1.3f Ge1 %1.3f " , nReal,Gv1,Ge1);
        gotoxy(7,10);
        cprintf(" Canal [1] - Corrente : % 1.3f (pu) Gv2 %1.3f Ge2 %1.3f " , iReal,Gv2,Ge2);
        gotoxy(7,11);
        cprintf(" Veloc. de Referencia : %1.2f (pu) iRef : %1.2f (pu) " , nRef,iRef);
        gotoxy(7,12);
        cprintf(" ne : %1.2f ie : %1.2f Vcontr : %1.2f (pu) " , ne, ie, u2);
    }

    t.stop(); /* Termina a temporização */
    dt = t.time(); /* Consideramos o tempo de amostragem c/ valor inicial */
} while (tecla!=27); /* Termina o Programa quando a tecla Esc e' pressionada */
}

```