Unifei - Universidade Federal de Itajubá

Grupo de Estudos em Manutenção Eletro-Eletrônica e Instalações

Programa de Pós-graduação em Engenharia da Energia

"IMPLEMENTAÇÃO DE UM ESTIMADOR DE VELOCIDADE EM LINGUAGEM C++ EM SUBSTITUIÇÃO AO TACO GERADOR PARA CONTROLE DE VELOCIDADE DE MOTOR DE CORRENTE CONTÍNUA"

Vinicius Zimmermann Silva

Dissertação apresentada à Universidade Federal de Itajubá para obtenção do Título de Mestre em Ciências em Engenharia da Energia

Orientador: Ângelo José Junqueira Rezek, Dr.

Itajubá –2004

DEDICATÓRIA

A minha esposa Cintya que sempre me motivou e valorizou os momentos que direcionei a estes estudos.

A meu pai José Carlos, minha mãe Maria das Graças e meus irmãos Mila, Léo e Mara, meus eternos incentivadores.

À DEUS, fonte de luz e sabedoria.

AGRADECIMENTOS

A Deus.

Aos meus pais, irmãos e esposa.

Ao Professor Ângelo José Junqueira Rezek por sua amizade e brilhante orientação.

À FAPEMIG-MG , pela aprovação do projeto de pesquisa " Implementação e Avaliação de Reguladores Digitais em Acionamentos Elétricos", processo TEC 2917-98, cuja concessão possibilitou a realização deste trabalho de pesquisa.

Aos funcionários do Laboratório do Departamento de Eletrotécnica, Edmundo F. da Silva, J. C. Anselmo, Raimundo R. da Silva, e Ana M. Tavares pelo apoio recebido durante os testes em laboratório.

Aos funcionários da Secretaria de Pós-Graduação, especialmente à Valéria Marcondes e à Cristina Silva pelo apoio durante todo o curso.

Ao Eng. José Marcelo Morais por responder minhas dúvidas acerca da linguagem de programação utilizada.

Aos meus chefes na ALSTOM, Eduardo Benazzi, Mauricio Nahas e Ernesto Kuwabara por me conceder liberação do trabalho a fim de assistir às aulas do curso de mestrado e desenvolver esta dissertação.

RESUMO

O presente trabalho tem como objetivo a implementação, via software em linguagem C++, de um estimador de velocidade de motor de corrente continua - MCC, substituto do taco gerador, para viabilizar o controle de velocidade do MCC utilizando a corrente da armadura como único dado monitorado de entrada do software.

Para se obter a velocidade real do motor através da corrente da armadura, utilizou-se de equações oriundas do modelamento matemático do MCC e ponte retificadora tiristorizada de seis pulsos. Os reguladores proporcional integral – PI - utilizados no software, foram modelados matematicamente através de equações recursivas. Os parâmetros necessários para ajustes destes reguladores como constante de tempo de velocidade, ganho do regulador para sinal de velocidade, constante de tempo de corrente, ganho do regulador para sinal de corrente entre outros foram disponibilizados e ajustados em tempo real na tela principal de execução do software. Software que ao longo deste trabalho será chamado de CHOPPV.

Para a montagem em laboratório foi utilizado o sistema de controle para o acionamento de um motor de corrente contínua – MCC- em mono-quadrante, realizado através de uma ponte conversora totalmente controlada. Para o sistema implementado, utilizou-se um microcomputador, uma placa de aquisição de dados, um circuito eletrônico de disparo e um transdutor de corrente, sensor Hall. O microcomputador com o software em linguagem C++ e a placa de aquisição responderam pela aquisição dos dados de entrada e controle das malhas de velocidade e corrente, e envio de um sinal de controle pelo qual se controla o circuito de disparo da ponte. A ponte retificadora através do ângulo de disparo alfa determina a tensão de alimentação do motor de corrente contínua.

No módulo de controle dos pulsos foi utilizado circuito eletrônico de disparo do tipo rampa, implementado com o CI TCA 780 (Icotron-Siemens), um circuito integrado desenvolvido para controlar o ângulo de disparo de tiristores, transistores e triacs continuamente de 0º a 180º.

Será possível verificar que o estimador de velocidade implementado em laboratório apresentou boa regulação, inclusive quando submetido à degrau de carga e a alteração da velocidade de referência. Foi constatado também grande similaridade dos valores de velocidade do eixo do motor com os valores referência apresentados pelo software. O sistema

RESUMO

utilizado está detalhado nesta dissertação assim como os resultados obtidos, os cálculos e os parâmetros adotados.

Palavras Chaves: Motor; Corrente Contínua; Software C++; Estimador de Velocidade; Taco gerador; Regulação; Armadura.

ABSTRACT

The purpose of this work is to implement via a software in C++ language a estimator of speed for a DC motor, standing for taco generator, in order to evaluate the speed control of MCC using the armature current of motor as the only input monitored by the software.

To obtain the real speed of the motor from its armature current, it has been used equations from the mathematics model of the DC current motor and of six pulses fully controlled rectifier bridge. The proportional integral regulators PI – used in the software have been mathematically modulated from recursive equations. The parameters needed to adjust these regulators among which speed time constant, regulators gain of the speed signal, current time constant, regulators gain of the current signal have been defined and adjusted in real time from the main operating window of the software. This software will be named CHOPPV.

For the assembly in laboratory, it has been used a system of drive controlling of the DC motor based on a fully controlled rectifier. The system has been implemented by one computer, one data acquisition board, one triggering electronic circuit and a current sensor – Hall sensor. The input data is acquired through the data acquisiter board and handled by the software. Then, this one processes the speed and the current and issues one control signal to the triggering electronic circuit which adjusts the firing angle of rectifier bridge. Then, the rectifier bridge fixes the supply voltage of the DC motor from the firing angle.

In the module of controlling the pulses, it has been used triggering electronic circuits of ramp type made with CI TCA 780 (Icotrons – Siemens) which is a integrated circuit developed to control the firing angle of the thyristors, transistors and triacs steadely from 0º to 180º.

It was possible to verify that the speed estimator developed in laboratory implements correct regulation even in case of load changes and of alteration of reference speed. It was also possible to verify that the speed of the motor axis and the one presented by the software were similar.

The system above is detailed in this article together with the results, calculations and used parameters.

ABSTRACT

Key Words: Motor; DC Current; C++ Software; Speed Estimator; Taco Generator; Regulation; Armature.

SUMÁRIO

1 INTRODUÇÃO 1

2 OBJETIVOS 3

2.1 OBJETIVO GERAL 3

2.2 OBJETIVOS ESPECÍFICOS 3

3 DESCRIÇÃO GERAL DOS EQUIPAMENTOS INTEGRANTES DO CIRCUITO OBJETO DE ESTUDO 4

3.1 INTRODUÇÃO 4

3.2 PONTE RETIFICADORA DE SEIS PULSOS 6

3.3 MOTOR DE CORRENTE CONTÍNUA 13

3.3.1 Equacionamento do MCC 15

3.4 ESTIMADOR DE VELOCIDADE DO MCC 23

3.5 GERADOR DE INDUÇÃO TRIFÁSICO 24

3.5.1 Introdução 24

3.5.2 Processo de Partida do Gerador de Indução Auto-Excitado 24

3.5.3 Circuito de Carga do Gerador de Indução Trifásico 27

3.5.4 Entrada de Carga 29

3.6 PLACA DE AQUISIÇÃO DE DADOS 29

3.7 TRANSDUTOR DE CORRENTE – SENSOR HALL 29

3.8 FILTRO DO SINAL DE CORRENTE 29

3.9 FILTRO DO SINAL DE VELOCIDADE 30

4 SOFTWARE EM LINGUAGEM C++ PARA ESTIMAÇÃO E CONTROLE DE VELOCIDADE DO MCC 31

4.1 INTRODUÇÃO 31

4.2 REGULADORES 35

4.3 CÓDIGO FONTE DO SOFTWARE PARA CONTROLE DE VELOCIDADE DO MCC 36

4.4 PROCEDIMENTO PARA EXECUÇÃO DO SOFTWARE CHOPPV 55

4.5 ENERGIZAÇÃO DO CIRCUITO E PARTIDA DO MOTOR DE CORRENTE CONTÍNUA 62

5 SISTEMÁTICA UTILIZADA PARA VALIDAÇÃO DAS EQUAÇÕES E PARÂMETROS ADOTADOS PARA ESTIMAÇÃO DE VELOCIDADE VIA SOFTWARE 63

5.1 INTRODUÇÃO 63

5.2 EQUAÇÕES PARA CÁLCULO DA VELOCIDADE REAL DO MOTOR 63

5.2.1 Cálculo da Força Contra-Eletromotriz 64

5.2.2 Cálculo da Velocidade Real do Motor 65

5.2.2.1 Introdução 65

SUMÁRIO

5.2.2.2 Equacionamento Teórico 65
5.2.2.3 Cálculo da Velocidade Real do Motor no Software CHOPPV 67
5.3 EQUAÇÕES VERIFICADAS PARA O CÁLCULO DE VA – TENSÃO DE ALIMENTAÇÃO DO MCC 69
5.3.1 Introdução 69
5.3.2 Alternativas para Cálculo de Va 69
5.3.2.1 Cálculo de Va utilizando cosseno linearizado 69
5.3.2.2 Cálculo de Va utilizando cosseno real 70
5.3.2.3 Equação polinomial para cálculo de Va 71
5.3.3 Análise das Alternativas para Cálculo de Va 72

6 ANÁLISE DOS RESULTADOS OBTIDOS 74

6.1 PARTIDA DO MOTOR 75
6.2 DEGRAU DE CARGA (6A) 76
6.3 DEGRAU DE CARGA (7,7A) 77
6.4 DEGRAU DE VELOCIDADE DE REFERÊNCIA DE 0,2PU A CARGA DE 6A 78
6.5 DEGRAU DE VELOCIDADE DE REFERÊNCIA DE 0,2PU A CARGA DE 7,7A 79

7 CONCLUSÃO 80

APÊNDICE A – MONTAGEM EM LABORATÓRIO 82
APÊNDICE B – INSTALAÇÃO DO SOFTWARE GERENCIADOR DA PLACA DE AQUISIÇÃO PCL 711 91
APÊNDICE C - PROCEDIMENTO PARA CRIAÇÃO DE PROJETO EM LINGUAGEM C++ 92
ANEXO A – MEMORIAL DE CÁLCULO PARA OBTENÇÃO DAS EQUAÇÕES RECURSIVAS 103
ANEXO B – CÓDIGO FONTE PARA ACIONAMENTO E CONTROLE DE VELOCIDADE DE MOTOR DE CORRENTE CONTÍNUA UTILIZANDO TACO GERADOR 107
REFERÊNCIAS BIBLIOGRÁFICAS 126

Lista de Figuras

Figura 1.1 -Diagrama de Blocos do Controle de Velocidade em Malha Fechada de uma Máquina CC.com Estimador de velocidade 2

Figura 1.2- Diagrama de Blocos do Controle de Velocidade em Malha Fechada de uma Máquina CC.com Taco Gerador 2

Figura 3.1 -Acionamento em Mono-quadrante Através de uma Ponte Conversora Graetz. 4

Figura 3.2 – Filosofia do Circuito de Disparo Tipo Rampa 5

Figura 3.3– Ponte conversora totalmente controlada de seis pulsos 6

Figura 3.4: Formas de onda para ponte retificadora controlada para $\alpha = 0°$ 11

Figura 3.5: Formas de onda para ponte retificadora controlada para $\alpha = 30°$ 12

Figura 3.6 – Motor de Corrente Contínua 13

Figura 3.7 – Esquema de ligações do motor de corrente contínua. 13

Figura 3.10 - Representações da Parte Mecânica e do Circuito da Armadura do Motor CC com Excitação Independente 15

Figura 3.11 – Diagrama de Blocos da Parte Mecânica do Motor CC. 19

Figura 3.12– Diagrama de Blocos do Circuito da Armadura da Máquina CC 21

Figura 3.13 - Diagrama de Blocos Completo da Máquina CC. 21

Figura 3.14 - Acionamento em Mono-quadrante de uma Máquina CC Através de uma Ponte Conversora Graetz (totalmente controlada). 22

Figura 3.15 – Auto – Excitação do Gerador de Indução 25

Figura 3.16– Conjunto Motor de Corrente Contínua e Gerador de Indução 26

Figura 3.17 – Circuito de Controle da Tensão para Carga do Gerador 27

Figura 4.1 – Fluxograma do software para controle de velocidade do MCC 34

LISTA DE FIGURAS

Figura 4.2: Primeira Tela para Execução do Software CHOPPV 55

Figura 4.3: Segunda Tela para Execução do Software CHOPPV 56

Figura 4.4: Terceira Tela de Execução do Software CHOPPV 57

Figura 4.5: Quarta Tela de Execução do Software CHOPPV 58

Figura 4.6: Quinta Tela de Execução do Software CHOPPV 59

Figura 4.7: Sexta Tela de Execução do Software CHOPPV 60

Figura 4.8: Tela de Execução do Software CHOPPV 61

Figura 3.10 - Representação da Parte Mecânica e do Circuito Elétrico da Armadura do Motor CC com Excitação Independente 65

Figura 5.1 –Função Cosseno Linearizada 70

Figura 5.2: - Equação Polinomial Va=f(Vcc) 71

Figura 5.3 – Equação com cosseno real para Cálculo de Va - Alternativa 2 73

Figura 5.4 – Cálculo de Va por Equação Polinomial - Alternativa 3 73

Figura 6.1 – Partida com Estimador de Velocidade 75

Figura 6.2: Partida com Taco 75

Figura 6.3 Degrau de Carga (6A) com Estimador de Velocidade 76

Figura 6.4: Degrau de Carga (6A) com Taco 76

Figura 6.5: Degrau de Carga Nominal com Estimador de Velocidade 77

Figura 6.6: Degrau de Carga Nominal com Taco 77

Figura 6.7: Degrau de Nref. de 0,2PU a Carga de 6A com Estimador 78

Figura 6.8: Degrau de Nref. de 0,2PU a Carga de 6A com Taco 78

Figura 6.9: Degrau de Nref. de 0,2PU a Carga de 7,7A com Estimador de Velocidade 79

Figura 6.10: Degrau de Nref. de 0,2PU a Carga de 7,7A com Taco 79

LISTA DE FIGURAS

Figura A.1 – Esquema Elétrico de Ligação Geral com Estimador de Velocidade 83

Figura A.2 – Esquema de Ligação Geral com Taco Gerador- 84

Figura A.3 – MCC – Ger. Indução – Indutor e Instrumentos de medida 85

Figura A.4 – Painel de Energia e Micro-computador 86

Figura A.5- Vista da Bancada 86

Figura A.6 – Detalhe do Módulo de Controle de Sinais do Conversor 87

Figura A.7 – Micro computador e Painel de Alimentação do Circuito 87

Figura A.8 – MCC, Taco Motor, Gerador, Reostato e Amperimetro 88

Figura A.9 – Sensor Hall e Filtro de Corrente 88

Figura A.10 – Configuração com utilização do Taco 89

Figura A.11 – Aquisição de Formas de Onda 89

Figura A.12 – Placa de Aquisição de Dados PCL 711 90

Figura A.13 – Circuito de Controle de Tensão da Carga do Gerador de Indução 90

Figura C1: Primeira Tela para Criação de Projeto 92

Figura C2: Segunda Tela para Criação de Projeto 93

Figura C3: Terceira Tela para Criação de Projeto 94

Figura C4: Quarta Tela para Criação de Projeto 95

Figura C5: Quinta Tela para Criação de Projeto 96

Figura C6: Sexta Tela para Criação de Projeto 97

Figura C7: Sétima Tela para Criação de Projeto 98

Figura C8: Oitava Tela para Criação de Projeto 99

Figura C9: Nona Tela para Criação de Projeto 100

LISTA DE FIGURAS

Figura 4.8: Tela de Execução do Software CHOPPV 101

Figura C10: Instrução para Fechamento de Projeto 102

Figura AA1 - Método de aproximação por integração trapezoidal. 104

Figura AB1 – Fluxograma do Software com Taco Gerador 107

Lista de Tabelas

Tabela I – Dados de Placa do MCC 14

Tabela II – Dados de Placa do Reostato Demarrador..... 14

Tabela III – Dados de Placa do Reostato de Campo 14

Tabela IV – Dados de Placa do Gerador de Indução..... 26

Lista de Abreviaturas e Siglas

MCC : Motor de Corrente Contínua

REG.I : Regulador de Corrente

REG.n : Regulador de Velocidade

CHOPPV : Software em Linguagem C++ para Estimação e Controle de Velocidade do Motor de Corrente Contínua

CHOPPV1: Software em Linguagem C++ para Estimação e Controle de Velocidade do Motor de Corrente Contínua Revisão 1

A/D : Conversor Analógico Digital

D/A : Conversor Digital Analógico

Taco : Taco Gerador

LISTA DE SÍMBOLOS

e =E : Força contra eletromotriz

n_u : Velocidade em PU do Motor de Corrente Contínua

n_{Real} : Velocidade de Realimentação do Motor de Corrente Contínua

I_{Ref} : Corrente de Referência

U2 = eff : Tensão de Alimentação AC fase-fase

V_{dR} =Va=u=U : Tensão DC entre o pólo positivo e o pólo negativo para alimentação do Motor de Corrente Contínua

U_{pn} : Tensão pólo positivo neutro

U_{Nn} : Tensão pólo negativo neutro

U_{ac1} : Tensão anodo-catodo no tiristor 1

xd : Reatância Indutiva da bobina (indutor) de alisamento

ia, ib , ic : Corrente em cada fase de alimentação da ponte retificadora (ia: Corrente na fase a de alimentação na ponte conversora)

i1, i2 , i3, i4, i5, i6 : Corrente em cada tiristor (i1 : Corrente no tiristor 1; i3 : Corrente no tiristor 3)

Ia : Corrente Contínua na saída da ponte retificadora (Corrente no Link DC)

α : ângulo de disparo dos tiristores

Vd0 : tensão de saída da ponte para ângulo de disparo igual a zero graus

E_{FN} : tensão fase neutro de alimentação da ponte conversora

Em : Valor máximo (pico) da tensão fase neutro de alimentação

Ra : Resistência total (armadura e indutor de alisamento)

La : Indutância total (armadura e indutor de alisamento)

LISTA DE SÍMBOLOS

ϕ : Fluxo magnético do motor

Ia : Corrente de armadura

M : Conjugado do motor

Mc : Torque de carga ou conjugado resistente

J : Momento de inércia (motor + carga)

n : Velocidade (rpm)

ω : Velocidade angular (rad/s)

M_N : Conjugado do motor nominal

n_o : Velocidade a vazio

I_N : Corrente de armadura nominal

vi : Fator multiplicador da corrente nominal para a obtenção da corrente com rotor bloqueado, quando tensão nominal é aplicada à armadura

T_H : Constante de tempo de aceleração

Ta : Constante de tempo da armadura

Vccd: Sinal gerado pelo software CHOPPV para controle do ângulo alfa dos tiristores

Vcc: Sinal gerador pelo software CHOPPV para controle do ângulo alfa dos tiristores após conversor D/A

UGi: Tensão Induzida no Gerador de Indução Trifásico

IGi: Corrente Induzida no Gerador de Indução Trifásico

iRef: Corrente de Referência

IR: Corrente de Referência após Filtro

Ie: Erro da corrente

iReal: Corrente de Realimentação

LISTA DE SÍMBOLOS

ilim.: Corrente limite ajustada via software para fins de proteção do motor

Tn: Constante de Tempo de Velocidade

Tgs2: Constante de Tempo dd Filtro

Ti: Constante de Tempo do Regulador de Corrente

VRn: Ganho do Regulador de Velocidade

VRi: Ganho do Regulador de Corrente

a1, a2,: Representam partes das equações recursivas para desempenhar a função do filtro do sinal gerador pelo regulador de velocidade

b1, b2, b3, b4: Representam partes das equações recursivas para desempenhar as funções do reguladores. de velocidade e corrente

T: Tempo de amostragem do software

DataBuf[3]: Variável para aquisição dos sinais de corrente e velocidade através da placa de aquisição

K : ganho proporcional

τ_I : tempo integral

y_k - Saída do regulador

e_k - Entrada do regulador

1 Introdução

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 foi possível desenvolver novas técnicas de controle digital, como por exemplo, a adaptativa e a vetorial [16].

Apesar de mais caras, as máquinas de corrente contínua ainda 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 laminação, nas indústrias siderúrgicas. São também muito empregadas no acionamento de veículos de tração elétrica, como trens e metrôs, e em componentes eletromotivos, para acionamento de vidros elétricos e limpadores de pára-brisa e na área médica em cadeiras elétricas transportadoras e esteiras para teste ergométrico. O ajuste otimizado dos reguladores de velocidade, de corrente e filtros, possibilita a operação estável da máquina nas condições de degraus de conjugados de carga e na referência de velocidade, possibilitando uma resposta rápida da máquina a estes distúrbios.

Propõe-se neste trabalho a substituição do taco gerador, usualmente utilizado como transdutor de velocidade para as malhas fechada de controle, por um estimador de velocidade desenvolvido em C++ o qual utiliza-se de equações recursivas e o sinal da corrente da armadura recebido via transdutor de corrente, sensor hall, e placa de aquisição, para estimação da velocidade do MCC. Esta velocidade estimada via software, aquisitada pela malha fechada de controle como velocidade real do motor, é o dado a que o software também se baseará para o controle da velocidade do motor. É válido ressaltar que a malha fechada de controle utilizada baseia-se em reguladores digitais proporcional e integral – PI.

O software concebido para estimação de velocidade foi uma complementação do software pré-existente que utiliza-se do taco gerador como transdutor de velocidade.

As figuras 1.1 e 1.2 a seguir ilustram respectivamente os diagramas de blocos que exemplificam as malhas de controle de velocidade do sistema com estimador de velocidade e com taco gerador.

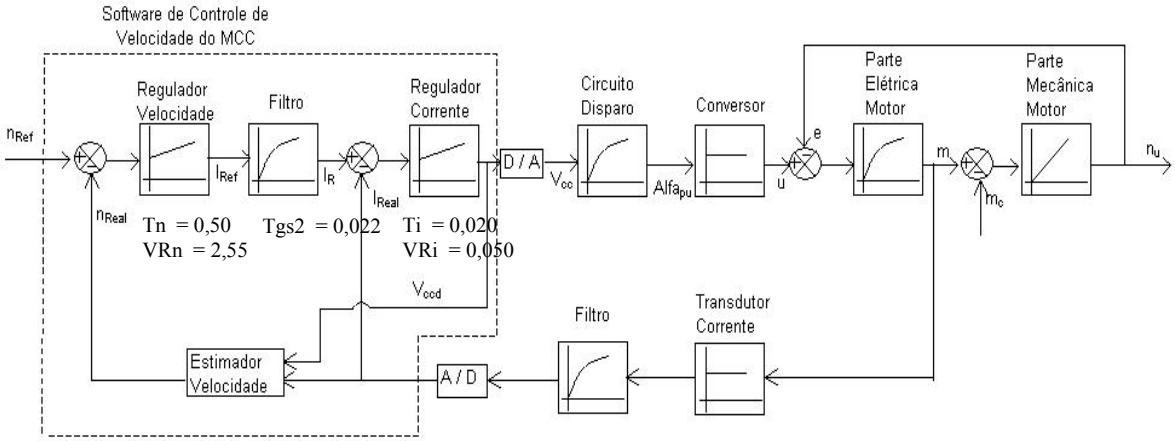

Figura 1.1 -Diagrama de Blocos do Controle de Velocidade em Malha Fechada de uma Máquina CC.com Estimador de velocidade

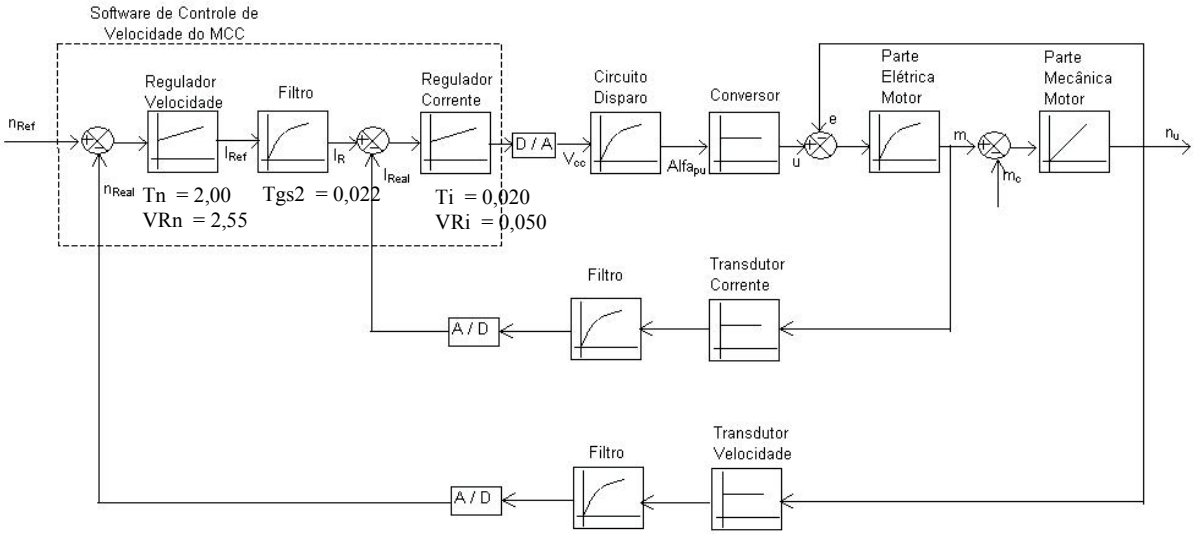

Figura 1.2- Diagrama de Blocos do Controle de Velocidade em Malha Fechada de uma Máquina CC.com Taco Gerador

2 Objetivos

2.1 Objetivo Geral

O objetivo geral deste trabalho é mostrar que o estimador de velocidade do MCC, desenvolvido via software, é capaz de estimar a velocidade real do eixo do motor através do sinal de corrente da armadura, obtido via transdutor de corrente e placa de aquisição, e enviar este valor para malhas fechadas de controle, que também foram concebidas via software e, assim, controlar a velocidade do motor com boa regulação.

2.2 Objetivos Específicos

Para se atingir o objetivo geral descrito no item 1.1 foram definidos os seguintes objetivos específicos:

- Descrição geral dos equipamentos integrantes do circuito objeto de estudo;
- Apresentação do Software em linguagem C++ para controle de velocidade de um MCC complementado com as equações e parâmetros que estabelecem o estimador de velocidade;
- Apresentar as equações, os cálculos e sistemática utilizada para se definir, dentre as opções, as equações mais adequadas e os melhores parâmetros visando obter boa regulação e controle de velocidade do motor;
- Análise dos resultados obtidos em laboratório visando mensurar qualitativamente todo o sistema implementado.

3 Descrição Geral dos Equipamentos Integrantes do Circuito Objeto de Estudo

3.1 Introdução

A figura 3.1 a seguir ilustra o circuito geral montado em laboratório para controle de velocidade do motor de corrente contínua – MCC – com o estimador de velocidade desenvolvido via software.

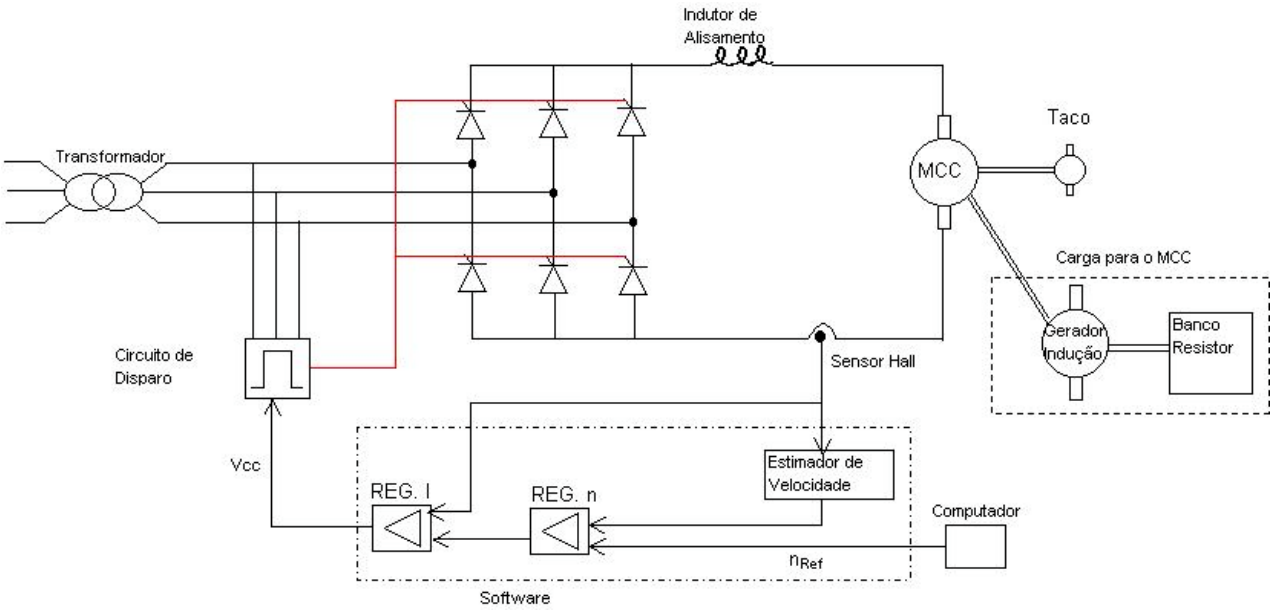

Figura 3.1 -Acionamento em Mono-quadrante Através de uma Ponte Conversora Graetz.

Como demonstrado pela figura 3.1, o circuito é composto por uma ponte retificadora Graetz de seis pulsos totalmente controlada, um motor de corrente contínua, um gerador de indução (carga para o MCC), um transdutor de corrente (sensor Hall), micro-computador 486 com software em linguagem C++. Além disso, instrumentos de medição como amperímetros, multímetros, reostatos de ajuste da corrente de campo, reostato demarrador, bobina para diminuição do "ripple" de corrente e filtro de corrente complementam o sistema. As informações detalhadas sobre os componentes do circuito serão apresentadas ao longo deste capítulo.

O sinal Vcc é gerado através do software e de uma placa de aquisição de dados, contendo conversores A/D e D/A que em função do sinal de corrente da armadura e de velocidade gera um sinal Vcc que enviado ao circuito eletrônico de disparos controla a tensão de alimentação da armadura do MCC.

DESCRIÇÃO GERAL DOS EQUIPAMENTOS INTEGRANTES DO CIRCUITO OBJETO DE ESTUDO

Na figura 3.2 mostra como o circuito eletrônico de disparos implementado com o circuito integrado TCA 780** interpreta o sinal Vcc recebido do conversor D/A da placa de aquisição do micro-computador.

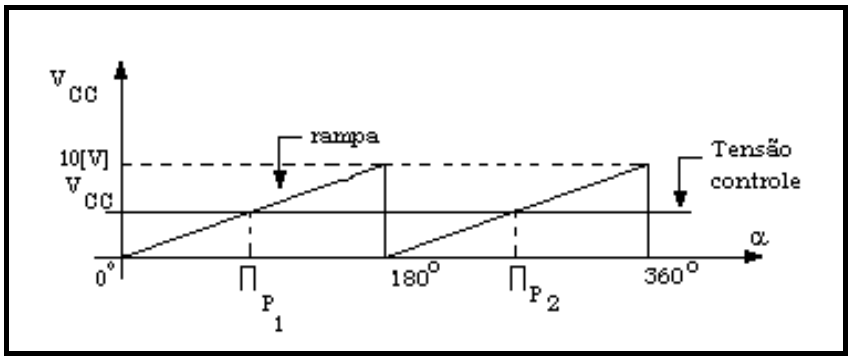

Figura 3.2 – Filosofia do Circuito de Disparo Tipo Rampa

** TCA 780 (Icotron-Siemens) é um circuito integrado desenvolvido para controlar o ângulo de disparo de tiristores, transistores e triacs continuamente de 0o a 180o.

A ponte retificadora é alimentada pela rede em aproximadamente 220Vca.

3.2 Ponte Retificadora de Seis Pulsos

A figura 3.3 ilustra a ponte conversora totalmente controlada de seis pulsos utilizada em laboratório para alimentação elétrica do motor de corrente contínua [14]. A seguir será apresentado o conceito matemático de cada parâmetro ilustrado na figura 3.3, de modo a facilitar o entendimento do funcionamento do retificador.

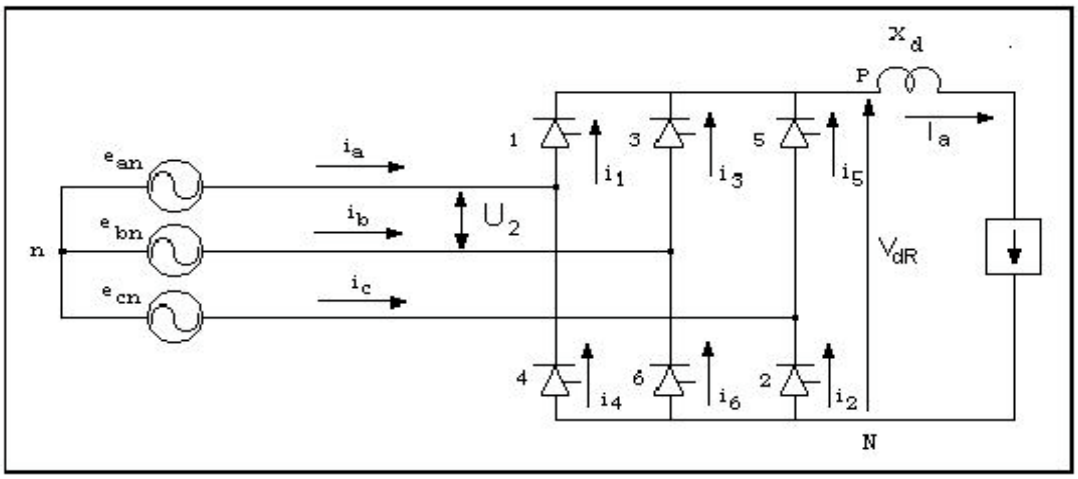

Figura 3.3– Ponte conversora totalmente controlada de seis pulsos

$\mathbf{e_{an}}$; $\mathbf{e_{bn}}$; $\mathbf{e_{cn}}$: Tensão AC fase-neutro, de alimentação da Ponte Conversora Retificadora;

eff : Tensão de alimentação AC fase-fase

$\mathbf{V_{dR}}$: Tensão DC, entre o pólo positivo e o pólo negativo ($\mathbf{U_{PN}}$)

$\mathbf{U_{Pn}}$: Tensão pólo positivo neutro

$\mathbf{U_{Nn}}$: Tensão pólo negativo neutro

$\mathbf{U_{AC1}}$: Tensão anodo-catodo no tiristor 1

$\mathbf{x_d}$: Reatância Indutiva da bobina (indutor) de alisamento

$\mathbf{i_a}$, $\mathbf{i_b}$, $\mathbf{i_c}$: Corrente em cada fase de alimentação da ponte retificadora ($\mathbf{i_a}$: Corrente na fase **a** de alimentação na ponte conversora)

$\mathbf{i_1}$, $\mathbf{i_2}$, $\mathbf{i_3}$, $\mathbf{i_4}$, $\mathbf{i_5}$, $\mathbf{i_6}$: Corrente em cada tiristor ($\mathbf{i_1}$: Corrente no tiristor **1**; $\mathbf{i_3}$: Corrente no tiristor **3**)

$\mathbf{I_a}$: Corrente Contínua na saída da ponte retificadora (Corrente no Link DC)

As forças eletromotrizes de alimentação da ponte conversora retificadora são dadas pelas equações:

$$e_{an} = E_m \operatorname{sen}(\omega t + \pi/6) \qquad (3.1)$$

$$e_{bn} = E_m \operatorname{sen}(\omega t - \pi/2) \qquad (3.2)$$

$$e_{cn} = E_m \operatorname{sen}(\omega t + 5\pi/6) \qquad (3.3)$$

Onde

$\boldsymbol{E_m}$: Valor máximo (pico) da tensão fase neutro de alimentação da ponte conversora.

As correntes nas fases são dadas por:

$$i_a = i_1 - i_4 \qquad (3.4)$$

$$i_b = i_3 - i_6 \qquad (3.5)$$

$$i_c = i_5 - i_2 \qquad (3.6)$$

O Valor médio da corrente em cada tiristor é dado por:

$$I_m = \frac{Ia}{3} \qquad (3.7)$$

Sendo:

$\mathbf{I_m}$: Valor médio da corrente em cada tiristor;

$\mathbf{I_a}$: corrente contínua na saída da ponte.

O Valor eficaz desta corrente é dado por:

$$I_{rms} = \frac{Ia}{\sqrt{3}} \qquad (3.8)$$

O Valor médio de tensão na saída da ponte é dado por:

$$V_{dR} = \frac{3\sqrt{2}}{\pi} U_2 \cos\alpha \qquad (3.9)$$

$$V_{do} = \frac{3\sqrt{2}}{\pi} U_2 \qquad (3.10)$$

$$V_{dR} = V_{d0} \cos\alpha \qquad (3.11)$$

Sendo:

α : ângulo de disparo;

$\mathbf{U_2 = E_{FF}}$: Valor eficaz da tensão fase fase de alimentação da ponte conversora;

V_{d0} : tensão de saída da ponte para ângulo de disparo igual a zero graus.

Também pode-se escrever:

$$V_{dR} = \frac{3\sqrt{6}}{\pi} E_{FN} \cos\alpha \qquad (3.12)$$

Sendo:

$$U_2 = E_{FF} = \sqrt{3}\ E_{FN} \qquad (3.13)$$

$$E_m = \sqrt{2}\ E_{FN} \qquad (3.14)$$

Onde:

E_{FN} : tensão fase neutro de alimentação da ponte conversora.

E_m : Valor máximo (pico) da tensão fase neutro de alimentação.

$$V_{dR} = \frac{3\sqrt{3}}{\pi} E_m \cos\alpha \qquad (3.15)$$

Como informado anteriormente, o ângulo α será definido em função do software que enviará sinal digital à placa de aquisição que converterá para um sinal analógico entre 0 e 10V o qual será interpretado pelo módulo de controle de pulsos conforme figura 3.1.2.

A título de exemplificação são apresentados abaixo as formas de onda VdR para ângulo α ajustado para 0° e 30°.

A figura 3.4 ilustra as formas de onda da ponte retificadora de seis pulsos para ângulo de disparo α = 0°. Com este ajuste, a ponte tiristorizada se comporta como se fosse uma ponte não controlada a diodos.

Como

$$V_{dR} = V_{d0} \cos\alpha \qquad (3.16) \qquad \cos\alpha = \cos 0 = 1$$

$$V_{dR} = V_{do} = \frac{3\sqrt{2}}{\pi} U_2 = 1{,}35U_2 \qquad (3.17)$$

$$V_{dR} = 1{,}35U_2 \cos\alpha \qquad (3.18)$$

As formas de onda da ponte conversora para α = 30°, são apresentadas na figura 3.5. Os pulsos de disparo são aplicados aos tiristores possibilitando a entrada em condução dos mesmos quando α=30º.

Os números que aparecem juntos aos tiristores representam a ordem com que os mesmos são disparados e entram e condução [13].

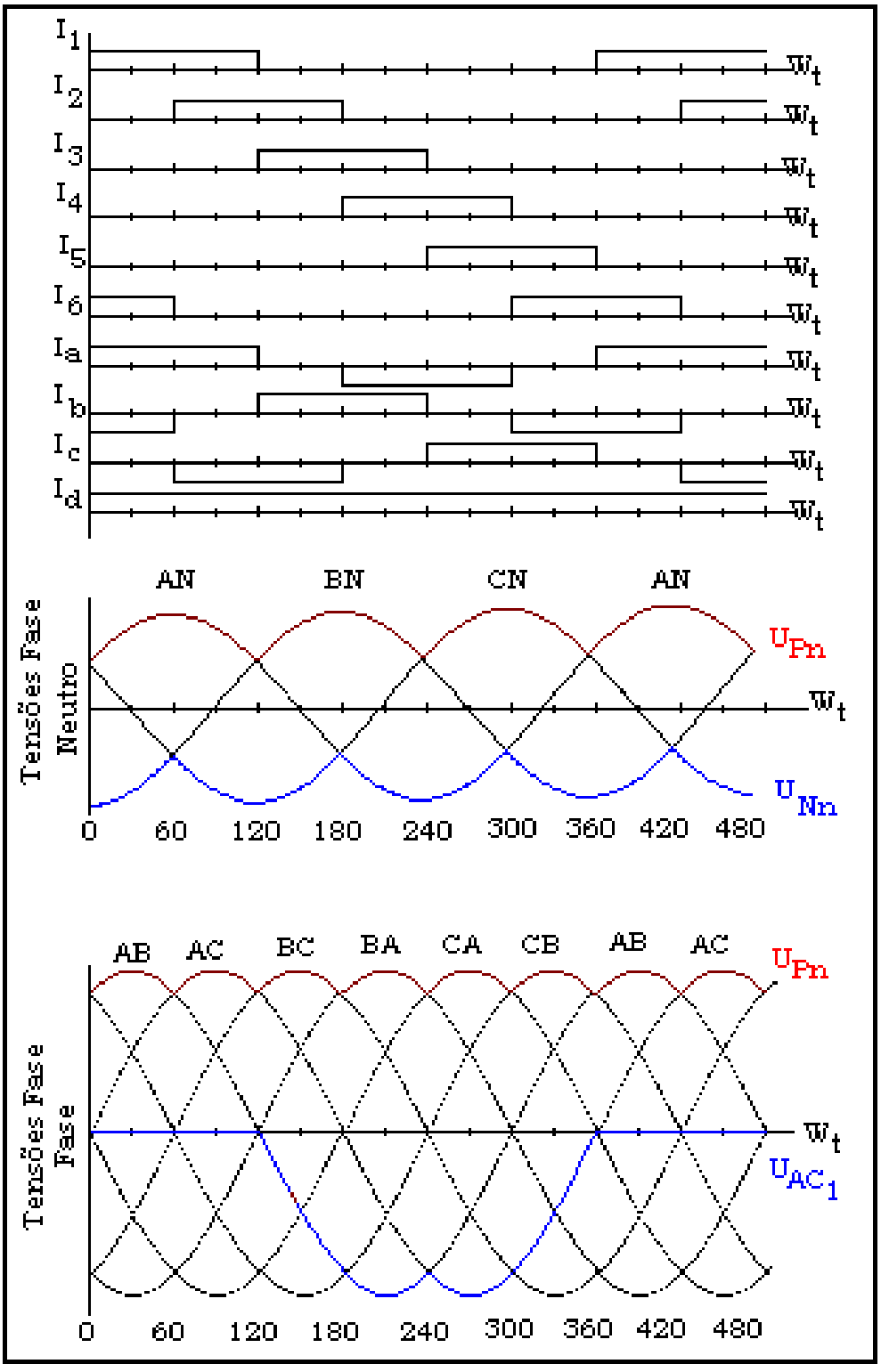

Figura 3.4: Formas de onda para ponte retificadora controlada para $\alpha = 0°$

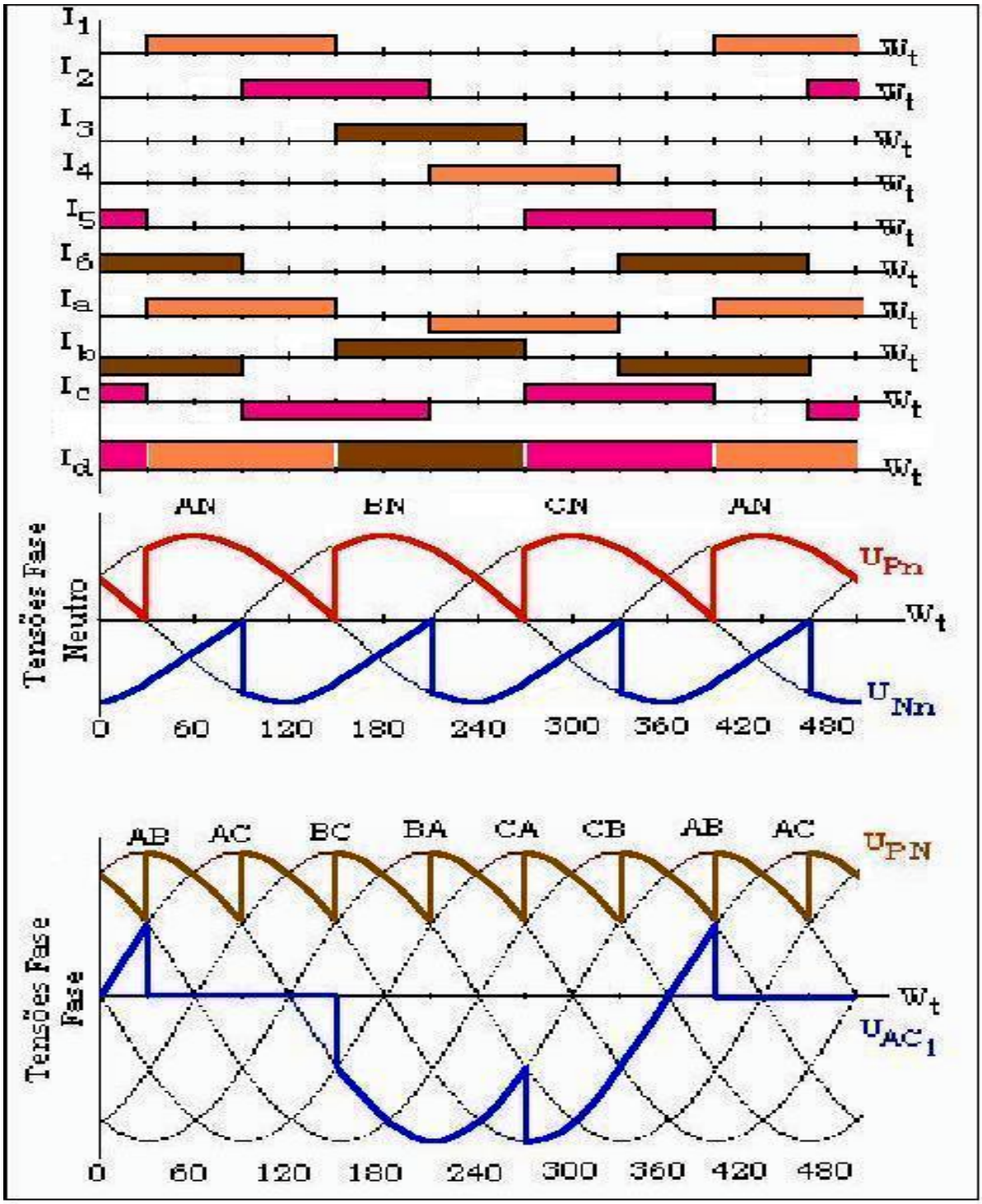

Figura 3.5: Formas de onda para ponte retificadora controlada para $\alpha = 30°$

3.3 Motor de Corrente Contínua

A figura 3.6 mostra o motor de corrente contínua cuja velocidade será controlada via software e através de estimação de velocidade, ou seja, o taco gerador, que também aparece na foto abaixo, não será utilizado como transdutor de velocidade para o sistema de controle.

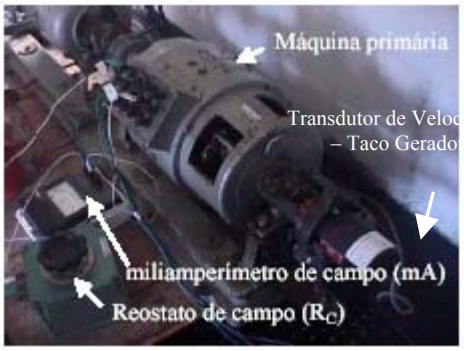

Figura 3.6 – Motor de Corrente Contínua

Para proteção do motor, a partida deste é feita mediante acionamento de um reostato demarrador colocado no circuito de armadura. Os bornes c e d das figuras 3.7 e 3.8 são os terminais da bobina de retenção.

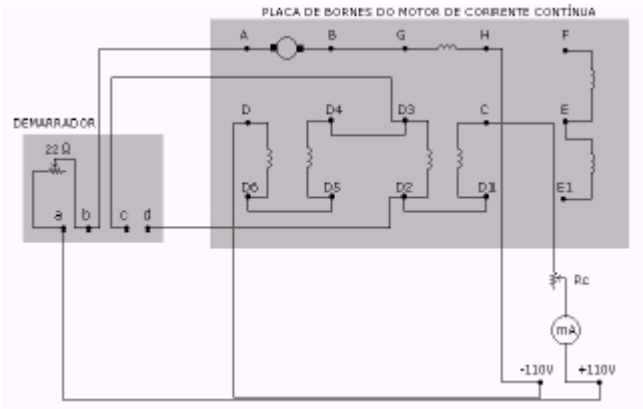

Figura 3.7 – Esquema de ligações do motor de corrente contínua.

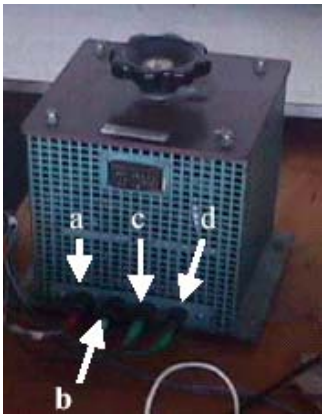

Figura 3.8 – Rostato Demarrador

Figura 3.9 – Foto do Reostato de Campo

A tabela I ilustra os dados de placa do MCC da figura 3.6:

Fabricante:	ELETROMÁQUIN
Tipo	CG1-4
Potência	1,7 [kW]
Tensão nominal de armadura	220 [V]
Rotação	1500 [RPM]
Corrente nominal de armadura	7,72 [A]
Corrente de campo máxima	0,6 [A]

Tabela I – Dados de Placa do MCC

A tabela II ilustra os dados de placa do Reostato Demarrador da figura 3.8:

Fabricante	EQUACIONAL
Número	2772
Potência	1,5 [kW]
Corrente	10 [A]
Resistência	22 [Ω.]

Tabela II – Dados de Placa do Reostato Demarrador

A tabela III ilustra os dados de placa do Reostato de Campo da figura 3.9:

Fabricante	EQUACIONAL
Número	2776
Tensão máxima	220 [V]
Corrente máxima	0,6 [A]
Resistência	700 [Ω.]

Tabela III – Dados de Placa do Reostato de Campo

3.3.1 Equacionamento do MCC

As representações da parte mecânica e do circuito da armadura do motor de corrente contínua com excitação independente estão mostradas na figura 3.10 [18]. Sucede-se a estas representações, o modelamento matemático do motor de corrente contínua visando favorecer amplo entendimento das diversas situações operacionais que este será submetido.

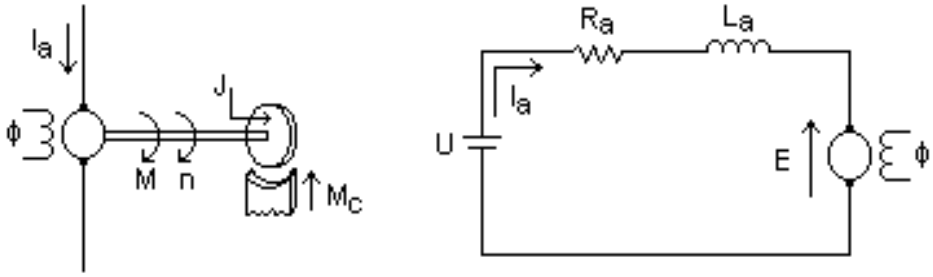

Figura 3.10 - Representações da Parte Mecânica e do Circuito da Armadura do Motor CC com Excitação Independente

E - Força contra eletromotriz;

U - Tensão de alimentação;

Ra - Resistência total (armadura e indutor de alisamento);

La - Indutância total (armadura e indutor de alisamento);

ϕ - Fluxo magnético do motor;

Ia - Corrente de armadura;

M - Conjugado do motor;

Mc - 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 dador por:

$$M = K_m \phi I_a \qquad (3.19)$$

sendo, K_m e ϕ constantes.

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

$$M - M_c = J \cdot \frac{d\omega}{dt} \qquad (3.20)$$

Aplicando transformada de Laplace na equação (3.20), tem-se:

$$M - M_c = J s \omega \qquad (3.21)$$

A velocidade angular ω pode ser expressa como:

$$\omega = \frac{2\pi}{60} \cdot n \qquad (3.22)$$

Substituindo a equação (3.22) na equação (3.21), resulta:

$$\frac{2\pi}{60}\cdot n = \frac{1}{Js}\cdot\left(M - M_c\right) \tag{3.23}$$

Logo,

$$\frac{2\pi}{60}\cdot\frac{n}{n_o}\cdot n_o = \frac{1}{Js}\cdot\left(\frac{M - M_c}{M_N}\right)M_N \tag{3.24}$$

$$\frac{n}{n_o} = \frac{1}{\frac{2\pi}{60}\cdot\frac{Jn_o s}{M_N}}\cdot\left(\frac{M - M_c}{M_N}\right) \tag{3.25}$$

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

$$T_H = \frac{2\pi}{60}\cdot\frac{Jn_o}{M_N} \tag{3.26}$$

Então,

$$\frac{n}{n_o} = \frac{1}{sT_H}\cdot\left(\frac{M}{M_N} - \frac{M_c}{M_N}\right) \tag{3.27}$$

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

$$\frac{M}{M_N} = m \qquad (pu) \qquad (3.28)$$

$$\frac{M_c}{M_N} = m_c \qquad (pu) \qquad (3.29)$$

$$\frac{n}{n_o} = n_u \qquad (pu) \qquad (3.30)$$

$$\frac{I_a}{I_N} = i_a \qquad (pu) \qquad (3.31)$$

onde,

M_N – Conjugado do motor nominal;

n_o – Velocidade a vazio;

I_N – Corrente de armadura nominal.

Logo,

$$n_u = \frac{1}{sT_H} \cdot (m - m_c) \qquad (3.32)$$

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

A figura 3.11 ilustra o diagrama de blocos da parte mecânica do motor CC

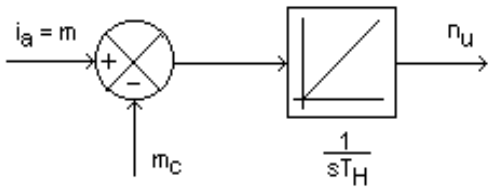

Figura 3.11 – Diagrama de Blocos da Parte Mecânica do Motor CC.

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

$$U = R_a I_a + L_a \cdot \frac{d I_a}{dt} + E \qquad (3.33)$$

Aplicando transformada de Laplace na equação (3.33), tem-se:

$$I_a = \frac{U - E}{R_a + sL_a} \qquad (3.34)$$

A constante de tempo da armadura Ta é definida como:

$$T_a = \frac{L_a}{R_a} \qquad (3.35)$$

Então,

$$\frac{I_a}{I_N} = \frac{U - E}{U_N} \cdot \frac{U_N}{R_a I_N} \cdot \frac{1}{1 + sT_a} \qquad (3.36)$$

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

$$\frac{U}{U_N} = u \qquad \text{(pu)} \qquad (3.37)$$

$$\frac{E}{U_N} = e \qquad \text{(pu)} \qquad (3.38)$$

onde,

UN – Tensão de alimentação nominal.

Logo,

$$i_a = \frac{v_i}{1 + sT_a} \cdot (u - e) \qquad (3.39)$$

onde,

$$v_i = \frac{U_N}{R_a I_N} \qquad (3.40)$$

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

A figura 3.12 mostra o diagrama de blocos das grandezas elétricas da armadura.

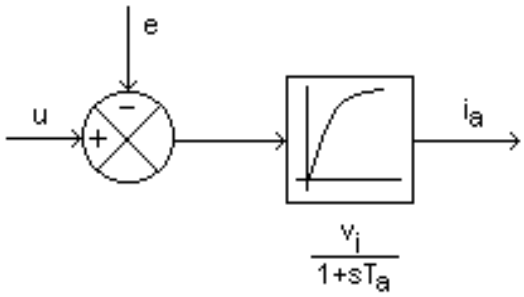

Figura 3.12– Diagrama de Blocos do Circuito da Armadura da Máquina CC

A determinação da constante de tempo T_a pode ser feita medindo-se a resistência R_a e a indutância L_a do circuito da armadura. Deve ser levada em consideração a indutância de alisamento incluída em série com o circuito da armadura.

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

O diagrama de blocos da máquina CC, incluindo o circuito da armadura e a parte mecânica, é mostrado na figura 3.13.

A figura 3.14 ilustra um diagrama de blocos representativo de um acionamento em mono-quadrante utilizando uma ponte conversora Graetz.

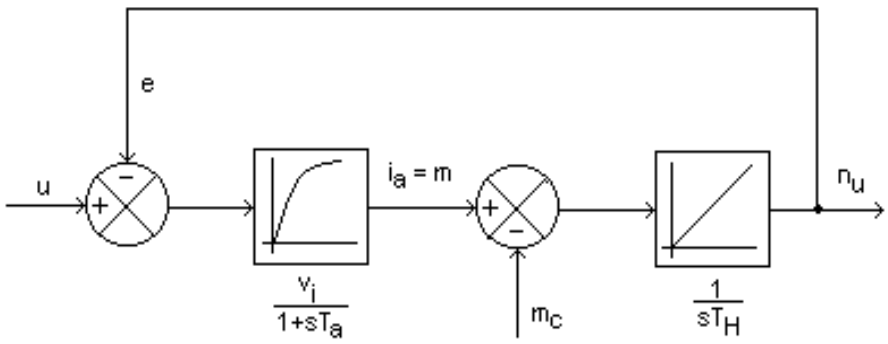

Figura 3.13 - Diagrama de Blocos Completo da Máquina CC.

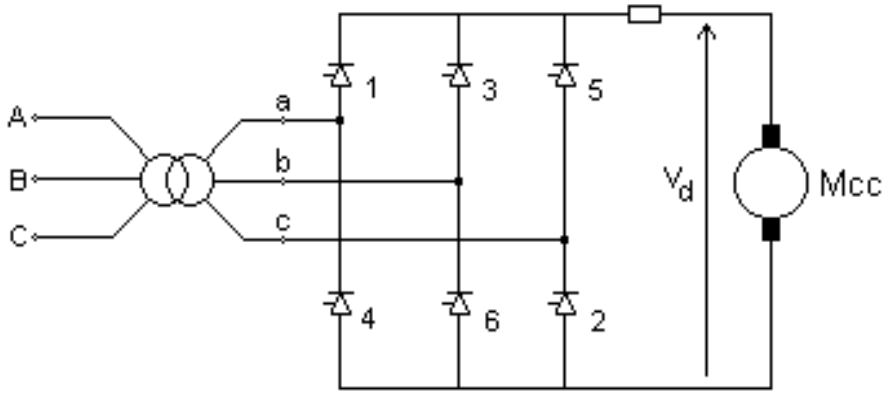

Figura 3.14 - Acionamento em Mono-quadrante de uma Máquina CC Através de uma Ponte Conversora Graetz (totalmente controlada).

3.4 Estimador de Velocidade do MCC

O estimador de velocidade será representado neste item através de sua equação representativa utilizada no software CHOPPV. A dedução matemática deste componente do circuito está mostrada e comentada nos capítulos 4 e 5 deste trabalho.

Estimador de Velocidade:

$$n_{PU} = \frac{1{,}35 * eff * \cos(Vcc * 3{,}1415 * 1{,}089)}{En} - \frac{RaIa}{En} \quad (3.41)'$$

n_{pu}: Velocidade Real do Motor em pu;

Ra: Resistência da Armadura;

Ia: Corrente da Armadura;

En: Força Contra eletromotriz nominal;

eff: Tensão fase-fase de alimentação da ponte retificadora;

Vcc: Sinal gerado pelo software CHOPPV para controle do ângulo alfa dos tiristores após conversor D/A;

1,089: Coeficiente de ajuste obtido empiricamente em laboratório.

Como eff, Ra, e En são valores constantes, verifica-se que n_{pu} está em função de Vcc e Ia.

3.5 Gerador de Indução Trifásico

3.5.1 Introdução

O gerador de indução utilizado tem a função de atuar como carga para o motor de corrente contínua, MCC, já que este atua como máquina primária para o gerador. Para o gerador atuar como carga devemos entender como acontece o processo de magnetização do circuito de campo do Gerador e, conseqüentemente, a partida deste gerador [15].

A potência reativa necessária para produzir o campo magnético do gerador não pode ser fornecida pela máquina primária, MCC, nem pelo rotor em gaiola de esquilo (não há terminais no rotor). A máquina de indução, portanto, somente pode operar como gerador fornecendo potência ativa se conectada a uma fonte externa de potência reativa.

Uma máquina de indução de uma determinada capacidade não pode desenvolver a mesma potência elétrica no modo gerador igual a que ele absorve de uma rede de energia elétrica na operação motor, pois as perdas (cobre, ferro, atrito e ventilação) reduzem a potência elétrica de saída. Teoricamente, a potência mecânica de entrada poderia ser aumentada para compensar as perdas e chegar a uma maior potência elétrica de saída. Contudo, isto é limitado pelo fato de que uma potência de entrada muito grande vai rapidamente sobrecarregar a máquina, isto é, sobreaquecê-la, e finalmente, queimar os enrolamentos do estator. Desta forma, as condições de operação como gerador são determinadas pela corrente do estator. Desta forma, as condições de operação como gerador são determinadas pela corrente do estator, que não deve exceder a corrente de placa do motor para a qual os enrolamentos do estator foram projetados.

3.5.2 Processo de Partida do Gerador de Indução Auto-Excitado

Quando se der partida no grupo turbina-gerador não há nenhuma corrente disponível nos enrolamentos do estator para produzir um campo magnético. Como, então o processo pode ser iniciado?

O fato dos núcleos de ferro do estator e rotor terem sido magnetizados durante prévia operação freqüentemente faz com que eles mantenham uma pequena quantidade de magnetismo residual. A figura 3.15 representa este magnetismo residual pelo “offset” da

curva de magnetização. Acionado pela turbina, o rotor da máquina de indução começa a girar e as linhas de fluxo deste magnetismo residual cortam o enrolamento do estator, no qual é induzida uma tensão U1.

A potência reativa necessária para produzir o campo magnético não pode ser fornecida pela máquina primária, nem pelo rotor em gaiola de esquilo (não há terminais no rotor). A máquina de indução desta forma, somente pode operar fornecendo potência ativa se conectada a uma fonte externa de potência reativa.

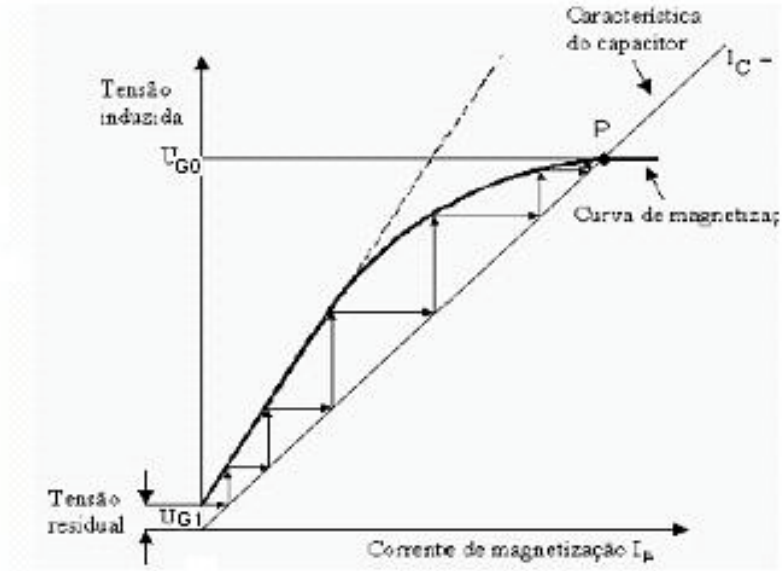

Figura 3.15 – Auto – Excitação do Gerador de Indução

A tensão U_{G1} agora carrega o capacitor, o qual por sua vez irá alimentar o enrolamento do estator com uma corrente de magnetização I_{G1}. Por sua vez, I_{G1} aumenta a magnetização da máquina, e a tensão correspondente U_{G2} é gerada. Este processo é repetido até que a corrente induzida I_{Gi} (produzida pela tensão U_{Gi}) e a corrente do capacitor Ic estiverem em equilíbrio. Este será o caso da intersecção das duas curvas no ponto P da figura 3.15. Este processo acontece na condição sem carga. Selecionado um capacitor C[μF] adequado, a tensão a vazio U_{G0} pode ser estabelecida conforme o desejado [15].

A figura 3.16 ilustra a máquina de indução usada como gerador neste trabalho.

Figura 3.16– Conjunto Motor de Corrente Contínua e Gerador de Indução

A tabela IV ilustra os dados de placa do gerador de indução:

Fabricante	ELETROMÁQUINAS
Tipo	A4-5^{A}-B3/4
Potência	1,86 [kW]
Tensão	220/380 [V] .Y
Rotação	1500 [RPM]
Corrente nominal	7,5/4,3 [A]
Freqüência	50 [Hz]
Fator de potência	0,82
Rotações por	1410
Rendimento	0,81
Data de	1965

Tabela IV – Dados de Placa do Gerador de Indução

3.5.3 Circuito de Carga do Gerador de Indução Trifásico

O gerador de indução foi utilizado como carga para o motor de corrente contínua, MCC, quando havia necessidade de uma análise da resposta da regulação de velocidade do MCC ao degrau de carga.

A carga do gerador de indução foi um banco de resistores ilustrado na figura 3.18 e o sistema montado entre o gerador e o banco de resistores para compensação de reativo junto ao gerador e controle de tensão no banco de resistores está ilustrado na figura 3.17 [15].

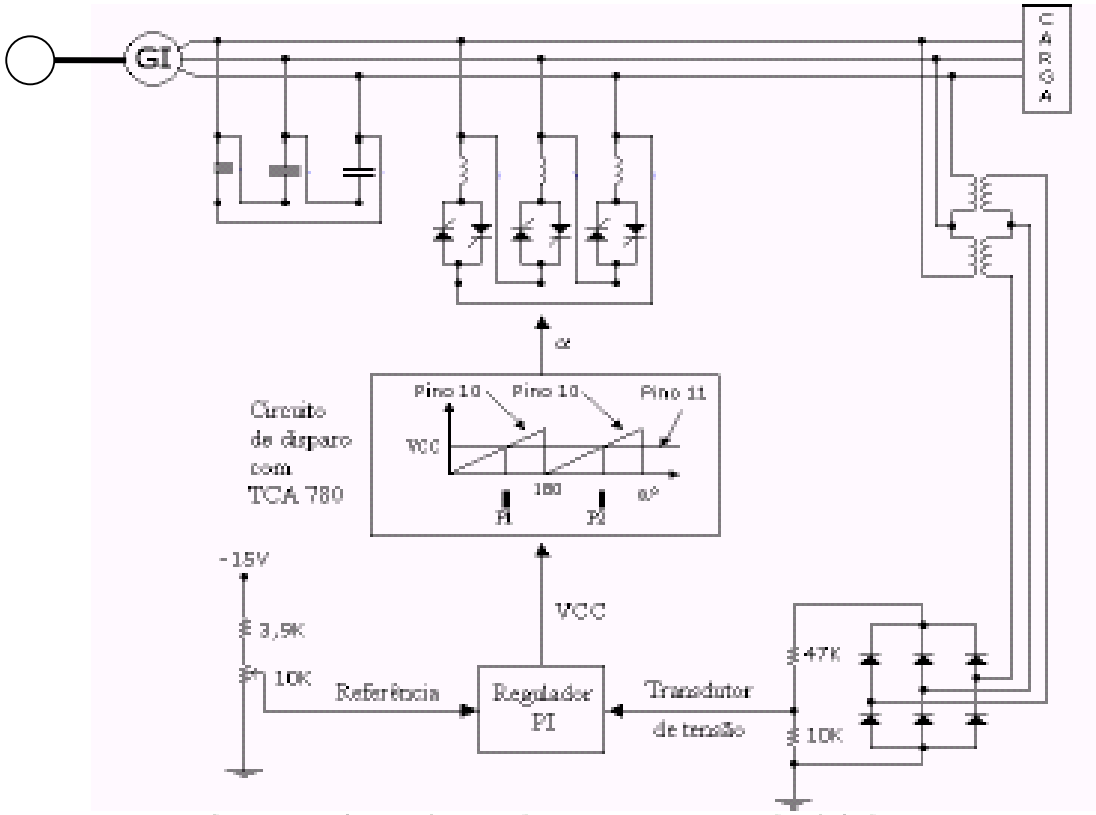

Figura 3.17 – Circuito de Controle da Tensão para Carga do Gerador

O objetivo deste circuito é controlar a tensão na carga do gerador de indução, independentemente da carga da máquina.

O método consiste basicamente de capacitores fixos associados a indutores controlados a tiristores para o controle da tensão. A fonte estática de compensação de reativos é composta de capacitores fixos ligados em triângulo, e indutores também ligados em triângulo. Em série com cada indutor estão dois tiristores em anti-paralelo. Como os

capacitores são fixos, a tensão gerada é regulada através da alteração da corrente dos indutores por meio da variação do ângulo de disparo dos tiristores.

Um regulador tipo PI (proporcional Integral) foi o escolhido para ser implementado na malha do controle de tensão do sistema proposto [16]. A otimização dos parâmetros dos reguladores foi feita em tempo real na tela de execução do software CHOPPV como será apresentado no capítulo 4.

Figura 3.18 – Banco de Resistores

3.5.4 Entrada de Carga

A entrada de carga no gerador de indução deverá ser estabelecida quando sua tensão atingir valor de tensão nominal, que é de 1 [pu].

A conexão de uma carga elétrica aos terminais do gerador de indução antes que este atinja sua tensão nominal, causará uma queda de tensão relativa nos seus terminais, fazendo com que funcione num valor de tensão inferior ao seu valor de tensão nominal [15].

3.6 Placa de Aquisição de Dados

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], ±2,5 [V], ±1,25 [V], ±0,625 [V], ±0,3125 [V], com tempo de conversão de até 25[μs]. A conversão D/A, tem a mesma resolução (12 bits), mas apenas com um canal de saída, com tempo de acomodação de 30 [μs], e faixas de saída de 0 a +5 [V] ou 0 a +10 [V].

Segundo a configuração do circuito utilizando Estimador de Velocidade foi utilizado um canal de entrada, para conversão A/D, correspondendo ao sinal de realimentação de corrente, e um canal de saída, para o sinal de controle V_{CC}.

Os passos necessários para instalação do programa gerenciador desta placa de aquisição são descritos no Apêndice B.

3.7 Transdutor de Corrente – Sensor Hall

O transdutor de corrente utilizado foi um sensor Hall, modelo LA100P do fabricante LEN. Este transdutor de corrente inclui divisor resistivo para limitar a tensão de entrada para o conversor A/D, sendo em seguida os sinais de tensão corrigidos por software para condições de carga proporcional à nominal.

3.8 Filtro do Sinal de Corrente

O filtro do sinal de corrente da armadura é usado para reduzir os valores de tensão proveniente do sensor Hall a fim de adequar aos valores especificados de entrada da placa de aquisição. Como pode ser visto na figura A1, o filtro do sinal de corrente está conectado ao sensor Hall e a placa de aquisição e composto por resistores e capacitores com o seguinte dimensionamento:

Ri (Resistência do Filtro de Corrente) = 6,8kohms

Ci (Capacitância do Filtro de Corrente) = 0,22microF

Ti (Cte. Tempo do Filtro de Corrente) = 1,5ms

3.9 Filtro do Sinal de Velocidade

O filtro do sinal de velocidade é usado para reduzir os valores de tensão proveniente do taco gerador a fim de adequar aos valores especificados de entrada da placa de aquisição. Como pode ser visto na figura A2, o filtro do sinal de velocidade está conectado ao taco gerador e a placa de aquisição e composto por resistores e capacitores com o seguinte dimensionamento:

Rn (Resistência do Filtro de Velocidade) = 47kohms

Cn (Capacitância do Filtro de Velocidade)= 2,2microF

Tn (Cte. Tempo do Filtro de Velocidade) = 100ms

4 Software em Linguagem C++ para Estimação e Controle de Velocidade do MCC

4.1 Introdução

Os programas ou softwares de controle utilizados, programa CHOPPV e programa gerenciador da placa de aquisição de dados PCL 711 (711CS.LIB), 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 (Vcc) para o circuito de disparo. Os programas incluem ainda temporização, diagnóstico de falha e ajuste de parâmetros on-line.

Para um melhor entendimento sobre a funcionalidade e operação do software CHOPPV, é apresentado a seguir, figura 4.1, o fluxograma representativo das etapas deste software como também uma descrição de cada etapa.

1 - Inicialização de variáveis: Esta etapa consiste na declaração das variáveis do programa assim como atribuição de valores iniciais a algumas variáveis como segue:

Lista das Principais Variáveis do Programa CHOPPV:

float d;

d: Parte da equação para o cálculo de Va, Tensão de Alimentação do MCC.

long float eff=218.0, Va, En=182.05, Ra=3.5;

eff: Tensão fase fase de rede de alimentação que alimenta a ponte retificadora;

Va: Tensão de alimentação do MCC;

En: Força Contra-Eletromotriz;

Ra: Resistência da Armadura.

float nRef=1.00, nReal=0.0, ne=nRef;

nRef.: Velocidade de Referência;

nReal: Velocidade de Realimentação;

ne: Erro da Velocidade

float iRef=1.2, iR=1.2, ie=-1.2, iReal=0.0, Ilim=1.2, iRefer=1.20;

iRef: Corrente de Referência;

IR: Corrente de Referência após Filtro;

Ie: Erro da corrente ;

iReal: Corrente de Realimentação;

ilim.: Corrente limite ajustada via software para fins de proteção do motor;

float Tn=0.550, Tgs2=0.022, Ti=0.020;

Tn: Constante de Tempo de Velocidade;

Tgs2: Constante de Tempo dd Filtro;

Ti: Constante de Tempo do Regulador de Corrente

float VRn=2.55, VRi=0.050, Vcc=0.1;

VRn: Ganho do Regulador de Velocidade;

VRi: Ganho do Regulador de Corrente;

Vcc: Sinal gerado pelo software CHOPPV para ajuste de ângulo de disparo dos tiristores;

float a1, a2, b1, b2, b3, b4, T=0.003;

a1, a2,: Representam partes das equações recursivas para desempenhar a função do filtro do sinal gerador pelo regulador de velocidade;

b1, b2, b3, b4: Representam partes das equações recursivas para desempenhar as funções do reguladores. de velocidade e corrente;

T: Tempo de amostragem do software.

float DataBuf[3];

DataBuf[3]: Variável para aquisição dos sinais de corrente e velocidade através da placa de aquisição

2 - Contagem do tempo de amostragem: O tempo de amostragem compreende o período de tempo entre a aquisição do sinal externo, que no caso é o sinal da corrente da armadura, e a emissão do sinal Vcc pelo software através da placa de aquisição com destino ao módulo controle de pulsos. Este tempo é definido em 0,003s;

3 - Conversão analógica digital: A aquisição do sinal externo da corrente da armadura pelo software CHOPPV é procedida através da conversão analógico digital do sinal de corrente proveniente do transdutor de corrente, que no caso é o sensor Hall, através da placa de aquisição PCL 711 descrita no item 3.5;

4 - Estimação de Velocidade: Com a aquisição do sinal de corrente pelo software, é procedida a Estimação de Velocidade através do software CHOPPV;

5 - Regulação de Velocidade e Corrente: Após a Estimação de Velocidade, é executada a regulação de velocidade e Regulação de Corrente para gerar o sinal Vcc de controle do ângulo de disparo dos tiristores, e conseqüente controle da tensão da armadura do MCC;

6 - Conversão Digital Analógica: A conversão digital analógica do sinal Vccd, Vcc digital, gerado pelo programa é realizada na placa PCL 711 descrita no item 3.5. Após a geração do Sinal Vccd pelo software CHOPPV, o tempo de amostragem termina.

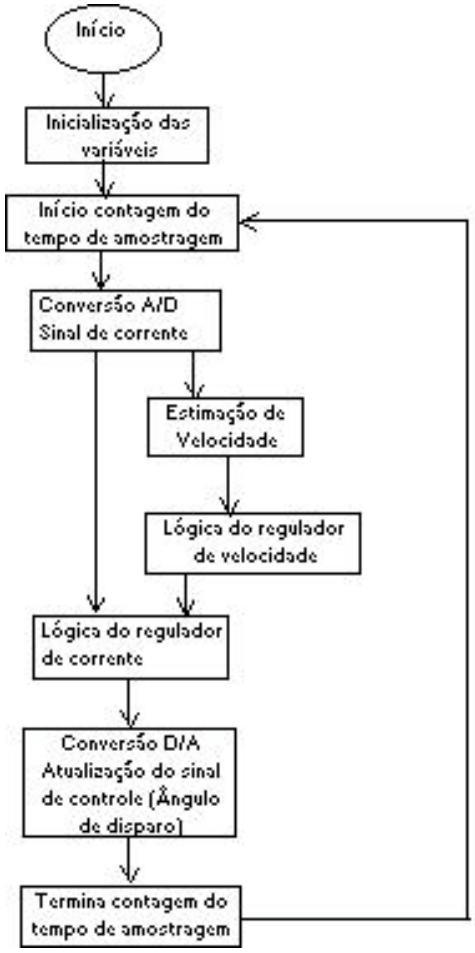

Figura 4.1 – Fluxograma do software para controle de velocidade do MCC

4.2 Reguladores

O regulador PI (proporciona - integral), [9], é o mais apropriado para esse tipo de controle, sendo muito utilizado em aplicações industriais onde se deseja um sistema um comportamento preestabelecido, atendendo a algumas especificações, como: overshoot, tempo de acomodação, erro em regime permanente, etc. Adota-se um critério de otimização, efetuando-se assim os ajustes do ganho e constante de tempo de ação integral do regulador, necessários para o controle e regulação de velocidade e corrente em malha fechada [16].

A equação idealizada de um regulador PI é dada por:

$$Gc(s) = k.\left[\frac{1+\tau_I \cdot s}{\tau_I \cdot s}\right] \qquad (4.1)$$

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 representar o comportamento desse sistema.

Quando se emprega modelos ou equações, para obter a expressão recursiva, basta substituir os operadores integrais 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 (1) e da integração trapezoidal, pode-se obter a equação recursiva capaz de representar o comportamento do regulador PI.

$$\frac{1}{s} = \frac{T}{2} \cdot \frac{z+1}{z-1} \qquad (4.2)$$

Onde

T - é o tempo de amostragem.

Substituindo-se a equação (4.2) em (4.1), tem-se a equação recursiva resultante:

$$y_K = y_{k-1} + b_1 \cdot \left(e_k - b_2 \cdot e_{k-1}\right) \qquad (4.3)$$

sendo:

y_k - Saída do regulador

e_k - Entrada do regulador

$$b_1 = k \cdot (1 + c); \quad b_2 = \frac{(1-c)}{(1+c)}; \quad c = \frac{T}{2 \cdot \tau_1}$$

Um melhor detalhamento sobre as equações recursivas está apresentado no Anexo A deste trabalho.

4.3 Código Fonte do Software para Controle de Velocidade do MCC

A seguir, o código fonte do programa CHOPPV com estimador de velocidade para controle de velocidade de motor de corrente contínua. A apresentação deste software está conforme à apresentada em ambiente Turbo C [7]

Ao longo deste programa foram colocados comentários a fim de detalhar todas as rotinas e cálculos executados.

```
/*
**********************************************************************
```

```
*  Programa    : CHOPPV                                    *
*  Descricao   : Controle digital com estimacao de velocidade para   *
*                acionamento e controle de velocidade de uma m quina *
*                de corrente continua utilizando cartao PCL-711B     *
*  Versao      : CHOPPV - VERSAO 0                                   *
*  Data        : 21/11/2003                          *         *

*********************************************************************
*/
/* Inclusao de Diretivas */
#include <stdio.h>
#include <conio.h>          /* Aceita diretivas, incluindo c¢digos */
#include <stdlib.h>              /* de fontes de outros */
#include <dos.h>                /* programas ou diret¢rios. */
#include <math.h>     /* Funcao Cosseno necessita desta biblioteca */
#include <timer.h>


/* Declara‡Æo de Vari veis Globais */


extern "C" pcl711(int, unsigned int *);   /* Inclui fun‡Æo "pcl711" definida
     inteiro, sem sinal       em um m¢dulo separado utilizando
                                        linguagem "C" */
```

```
unsigned int param[60];                    /* Defini‡Æo de um vetor de dados -
                                           array - que formam a tabela de
                                           par*f*metros inteiros e s/ sinal */
unsigned int datain[200], dataout[200];    /* Buffer de 10 dados inteiros +
                                           para conversÆo */
unsigned int far * datin, * datout;        /* Endere‡o do buffer de dados acima
                                           - pointer - tipo inteiro e longo:
                                           2 palavras c/ range de 1Mbyte */
int tecla,i;                               /* Vari veis de leitura do teclado
                                           e numero de canais */
/* Vari veis pontos flutuantes do controle */
float d;
long float eff=218.0, Va, En=182.05, Ra=3.5;
float nRef=1.00, nReal=0.0, ne=nRef;
float ne_1=0.1, nReal_1=0.0;
float iRef=1.2, iR=1.2, ie=-1.2, iReal=0.0, Ilim=1.2, 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=0.550, Tgs2=0.022, Ti=0.020;
float VRn=2.55, VRi=0.050, Vcc=0.1, Vcc_1=0.1, Vcontr=0.1;
float a1, a2, b1, b2, b3, b4, T=0.003;
float DataBuf[3];
float e,  v , dd ;
```

```
/* Declara‡Æo de vari veis void - significa que nÆo retorna um valor */

void conv_ad(void);

void conv_da(void);

void control(void);

void control1(void);

void control2(void);

void teclado(void);


/* ConversÆo AD - Tabela de par*f*metros */


void conv_ad()

{

      unsigned int i;

/* Pointer - Espa‡o de mem¢ria - Vari vel que cont,m um endere‡o que,

  normalmente , o endere‡o de outra vari vel."                  */

      datin = datain;                /* Atribui ao pointer datin o valor

                                        equivalente … vari vel datain   */

      param[0]  = 0;                 /* N£mero do cartÆo              */

      param[1]  = 0x220;               /* Endere‡o de Base I/O          */

      /* Frequ^ncia de amostragem = Frequ^ncia de base do cartÆo/(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 fun‡äes D/I          */

/* Offset do Buffer , o endere‡o de mem¢ria (buffer) onde os dados

serÆo 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;                /* Endere‡o do Buffer B (nÆo usado)*/

param[13] = 0;                /* Segmento- NÆo usado, setar em 0 */

/* A conversÆo 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;                /* N£mero de conversäes A/D        */

param[15] = 0;                /* Canal de in¡cio da conversÆo A/D*/

param[16] = 1;                /* Canal de parada da conversÆo A/D*/

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


      /* Indica‡Æo de falha na conversÆo A/D */


pcl711(3, param);       /* Func. 3 : Inicializa‡Æo do Hardware      */

if (param[45] != 0) {   /* Se par*f*metro 45 diferente de 0, fazer:   */
```

```
clrscr();          /* Limpar a tela                        */
printf("\n FALHA NA INICIALIZA€ÇO DO DRIVER !"); /* Imprimir   */
getch();           /* Mostrar a tela de sa¡da              */
exit(1);           /* Fecha o loop e sai com status 1 - Erro  */
        }

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

  pcl711(5, param);       /* Func 5 : Verifica‡Æo n£mero conversäes A/D*/
  if (param[45] != 0) {
clrscr();
printf("\n FALHA NO SOFTWARE DE TRANSFERÒNCIA DE DADOS A/D !");
getch();
exit(1);
        }
```

```
                    /* Conversäes A/D */


      for (i = 0; i < param[14]; i++)   /* Dados amostrados - canais 0 e 1 */
{
  DataBuf[i] = datain[i]  & 0xFFF;
        /* Coleta de dados para o buffer no endereco 0xFFF
(os trˆs primeiros d¡gitos hexadecimais podem ser zerados pois o
restante , suficiente p/ suportar 4096 d¡gitos bin rios ) */
  DataBuf[i] =((5.0 - (-5)) * DataBuf[i] / 4096) + (-5);
  /* ConversÆo para que o sinal de tensÆo seja dispon¡vel para
     aplica‡Æo nas equa‡äes 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)       : Inicio da escala do A/D "-5" V
  */
 }
/* Leitura da tensÆo de realimenta‡Æo para a malha de velocidade
  e de corrente, sob condi‡äes de velocidade e carga nominal /*
   /* Calculo Estimado da Velocidade */
 //  nReal=(DataBuf[0]/2.700);
     //Calculo da Tensao de Alimentacao do MCC;
```

```
//d=1.1814 - 0.0131*Vcc*180;   //cosseno linearizado- opcao 1;

d=cos(Vcc*3.14159274*1.089);          //cosseno real - opcao 2;

Va=1.35*eff*d;

// if ((Vcc<0.42))

                                    // Equacao Polinomial - opcao 3

//                                                                    Va=-
395142.22*Vcc*Vcc*Vcc*Vcc*Vcc*Vcc+379270.35*Vcc*Vcc*Vcc*Vcc*Vcc-
89505.65*Vcc*Vcc*Vcc*Vcc-17674.43*Vcc*Vcc*Vcc+9644.75*Vcc*Vcc-
1753.71*Vcc+366.92;

// if ((Vcc>=0.393)) Va=700000000.00*Vcc*Vcc*Vcc-
828900000.00*Vcc*Vcc+327165400.00*Vcc-43042101.00;

// if (Va<=0.0) (Va=0.0);


// Estimacao da velocidade;


iReal=(DataBuf[1]/1.493);

nReal=Va/En-((Ra*iReal*7.72)/En); //Calculo da Velocidade Real do MCC;

}


                    /* Controle */


/* Equa‡äes Recursivas para efetuar fun‡äes 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()
{
ne=nRef-nReal;
iRef=(b1*ne)+(b2*ne_1)+iRef_1; /* nreal= Realimenta‡Æo de velocidade*/

     //if(iRef>1.2) ne=1.0*ne;
```

```
    //if(iRef<-1.2) ne=1.0*ne;
iRef=(b1*ne)+(b2*ne_1)+iRef_1;
iRef_1=iRef;
ne_1=ne;


                /* Filtro da Corrente de Referˆncia */
iRefer=iRef;
iR=a1*(iRefer+iRefer_1)+a2*iR_1; /*  Sendo:                    */
 /* iR= Corrente refer. ap¢s filtro */
    /* iRef= Corrente referˆncia       */
if(iR>=Ilim) iR=Ilim;
if(iR<=-Ilim) iR=-Ilim; /* Limita‡Æo da corrente de referˆncia*/
iR_1=iR;
iRefer_1=iRefer;
}
                    /* Regulador de Corrente */
void control2()
{                           /*  Sendo:                          */
ie=iReal-iR;
Vcc=(b3*ie)+(b4*ie_1)+Vcc_1; /* ireal= realimenta‡Æo de corrente  */


          /* iR= Corrente ref. ap¢s filtro     */
```

```
            /* ie= Erro corrente- Ent. regulador */
if(Vcc>0.9) ie=1.0*ie;


if(Vcc<0.1) ie=1.0*ie;


Vcc=(b3*ie)+(b4*ie_1)+Vcc_1;
Vcc_1=Vcc;
ie_1=ie;
if(Vcc>0.9) Vcontr=0.9;
if(Vcc<.10) Vcontr=.10;    /* Limita‡Æo da tensÆo de controle   */
if(Vcc<=0.90 && Vcc>=.10) Vcontr=Vcc;


}
            /* Altera‡Æo nos par ƒmetros do sistema */
void teclado()
 {
    //     if (kbhit()) tecla=getch();  /* Se for pressionada alguma tecla, */
                                  /*    abrir a tela de saida */
   /* O acionamento destas teclas de subrotina, permitem ajuste on-line de
      par ƒmetros do sistema, com o ajuste sendo mostrado na tela de sa¡da */


  /* Teclas "s" e "d" atuando na valor da velocidade de referˆncia */
```

```
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 referˆncia */

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 referˆncia */

if (tecla==112 && nRef<2.0) nRef=nRef+0.20;  /* Ajuste limitado ao */

if (tecla==111 && nRef>-2.0) nRef=nRef-0.20;  /* 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<5.000) 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;
}

          /*  ConversÆo D/A - Tabela de Par ƒmetros */

void conv_da()
{
      datout=dataout;          /* Atribui ao pointer datout o valor
                               equivalente … vari vel dataout       */
      param[0]=0;             /* N£mero do cartÆo                  */
      param[1]=0x220;          /* Endere‡o de base I/O              */
   /* Offset do Buffer , o endere‡o de mem¢ria (buffer) onde os dados
     serÆo guardados. Segmento , o comprimento do buffer de dados */
      param[20] = FP_OFF(datout); /* Offset do buffer A dados  sa¡da D/A  */
      param[21] = FP_SEG(datout); /* Segmento do buffer A dados sa¡da D/A */
      param[22] = 0;           /* Endere‡o do Buffer B sa¡da(nÆo usado) */
      param[23] = 0;           /* Segmento sa¡da- NÆo usado, setar em 0 */
      param[24] = 1;           /* N£mero de conversäes D/A             */
      param[25] = 0;           /* Canal de in¡cio da conversÆo D/A      */
```

```
   param[26] = 0;            /* Canal de parada da conversÆo D/A      */


                  /* Indica‡Æo de falha na conversÆo D/A */


   pcl711(3, param);       /* Func 3 : Inicializa‡Æo do hardware       */
   if (param[45] != 0)  {  /* Se par*f*metro 45 diferente de 0, fazer:   */
clrscr();          /* Limpar a tela                          */
printf("\n FALHA NA INICIALIZA€ÇO DO DRIVER !"); /* Imprimir   */
getch();           /* Mostrar a tela de saida                */
exit(1);           /* Fecha o loop e sai com status 1 - Erro   */
          }
   pcl711(12, param);      /* Func 12: Inicializa‡Æo do conversor D/A  */
   if (param[45] != 0) {
clrscr();
printf("\n FALHA NA INICIALIZA€ÇO DO D/A !");
getch();
exit(1);
          }
   pcl711(13, param);     /* Func 13: Verifica‡Æo n£mero conversäes D/A*/
   if (param[45] != 0) {
clrscr();
printf("\n FALHA NO SOFTWARE DE TRANSFERÒNCIA DE DADOS D/A !");
```

```
   getch();
   exit(1);
            }

  /* ConversÆo para que o sinal de tensÆo de sa¡da das equa‡äes
     recursivas de controle em pu ocupe um espaco de endere‡o do
     buffer de dados de sa¡da.                          */
      dataout[0]=(4095*Vcontr);

 }
                  /* Programa principal */
void main(void)   /* Garante que as vari veis globais nÆo retornam valores */
{              /* Declara‡Æo de vari veis */
int xx=0;      /* Inicializa o n£mero de intera‡äes em 0 */
Timer t;         /* Contador de tempo "t" */

clrscr();

textbackground(4);            /* Define Fundo Vermelho (4) */
gotoxy(1,1);
cprintf(" PROGRAMA DE CONTROLE DE VELOCIDADE DE UM MCC COM
ESTIMADOR DE VELOCIDADE :  ");
```

```
gotoxy(1,3);

cprintf(" Programa com Estimador de Velocidade                    ");

gotoxy(1,4);

cprintf(" Aluno: Vinicius Zimmermann Silva                        ");

gotoxy(1,5);

cprintf(" Programa Utilizando Taco gerador                        ");

gotoxy(1,6);

cprintf(" Alunos: Fabricio de Lima Ribeiro  e  Otavio H. S. Vicentini      ");

gotoxy(1,7);

cprintf(" Orientador  :  professor  Angelo  J.  J.  Rezek                 ");


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


gotoxy(1,16);  cprintf(" [O] - Diminui a Vel. [0.05] ");

gotoxy(40,16); cprintf(" [P] - Aumenta a Vel. [0.05] ");

gotoxy(1,17);  cprintf(" [K] - Diminui a Vel. [0.01] ");

gotoxy(40,17); cprintf(" [L] - Aumenta a Vel. [0.01] ");

gotoxy(1,19);  cprintf(" [F] - Aumenta o VRn  [0.05] ");

gotoxy(40,19); cprintf(" [V] - Diminui o VRn  [0.05] ");

gotoxy(1,20);  cprintf(" [G] - Aumenta a Tn   [0.05] ");

gotoxy(40,20); cprintf(" [B] - Diminui a Tn   [0.05] ");

gotoxy(1,21);  cprintf(" [H] - Aumenta o VRi  [0.05] ");
```

```
gotoxy(40,21); cprintf(" [N] - Diminui o VRi  [0.05] ");

gotoxy(1,22);  cprintf(" [J] - Aumenta o Ti   [0.05] ");

gotoxy(40,22); cprintf(" [M] - Diminui o Ti   [0.05] ");


do{


  tecla=0;

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


  xx++;           /* Incrementa 1 no n£mero de intera‡äes */

  t.reset();      /* Reseta o temporizador "t" */

  t.start();      /* Inicia a temporiza‡Æo */


  asm cli;

  conv_ad();      /* Efetua a subrotina de conversÆo A/D */

  asm sti;

                  /* Condi‡Æo para efetuar as equa‡äes de controle -

                  sinal de realimenta‡Æo maior que o ru¡do normal */


    control();

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

```
ne=nRef-nReal;


if(iR<Ilim && iR>-Ilim) control1();

if(iR==Ilim && ne<0.0) control1();

if(iR==-Ilim && ne>0.0) control1();


 ie=iReal-iR;


 if(Vcc<0.90 && Vcc>.10)control2();

 if(Vcc>=0.9 && ie<0.0 ) control2();

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

 asm cli;

 conv_da();    /* Efetua a subrotina de conversÆo D/A */

 asm sti;

 if (xx>=100) {/* Imprime na tela resultados instant‚neos cada 100 intera‡äes */

   xx=0;


   gotoxy(1,9);

   textbackground(1);

cprintf(" VRn= %1.3f  Tn= %1.4f  VRi= %1.3f  Ti= %1.4f                ", VRn, Tn,
VRi, Ti);

   gotoxy(1,10);
```

```
      cprintf(" Tempo gasto: %1.6f                                    ", T);
      gotoxy(1,11);
      cprintf(" Canal [0] - Canal de Velocidade ;  Canal [1] - Canal de Corrente  ");
      gotoxy(1,12);
cprintf(" Canal [0] = % 1.3f (pu)          Canal [1] = % 1.3f (pu)       ", nReal, iReal);
      gotoxy(1,13);
cprintf(" nRef= %1.2f          Vcontr= %1.2f          iR= %1.4f ie= %1.2f       ", nRef,
Vcontr, iR, ie);
      gotoxy(1,14);
      cprintf(" Said.R.Veloc =  %1.3f    Said.R.Corr =  %1.3f                ", iRef, Vcc);
    }
    t.stop();       /* Termina a temporiza‡Æo */
    T=t.time();     /* Consideramos o tempo de amostragem c/ valor inicial

    /*T=0.003 s. Se o tempo de execucao do programa "t" , maior que "T", necessario
    tentar reduzi-lo. Se "t" e menor fazemos T=t.*/
   T=.0030;
} while (tecla!=27);
textbackground(0);
clrscr();
}
```

4.4 Procedimento para Execução do Software CHOPPV

Para estimação e controle de Velocidade do MCC, o software CHOPPV deve ser executado em conjunto com a biblioteca 711CS.LIB, porque o software CHOPPV faz a estimação e o controle da velocidade do MCC, e a biblioteca 711CS.LIB é responsável pela aquisição dos dados de entrada e a disponibilização do sinal de saída através das conversões A/D e D/A respectivamente. Em linguagem C++ para dois ou mais softwares serem executados em conjunto, é necessário a criação de um projeto. Os passos para criação de projeto estão descritos no apêndice C.

Neste item, são apresentados os passos necessários para execução do projeto que integra o software CHOPPV e a biblioteca 711CS.LIB. Para instalação do software gerenciador da placa de aquisição PCL 711, que contém a biblioteca 711CS.LIB, deve-se seguir os passos apresentados no Apêndice B.

Passo 1: Selecionar a opção *Open Project* arrastando-se o mouse sobre a comando Open Project, conforme ilustrado na figura 4.2.

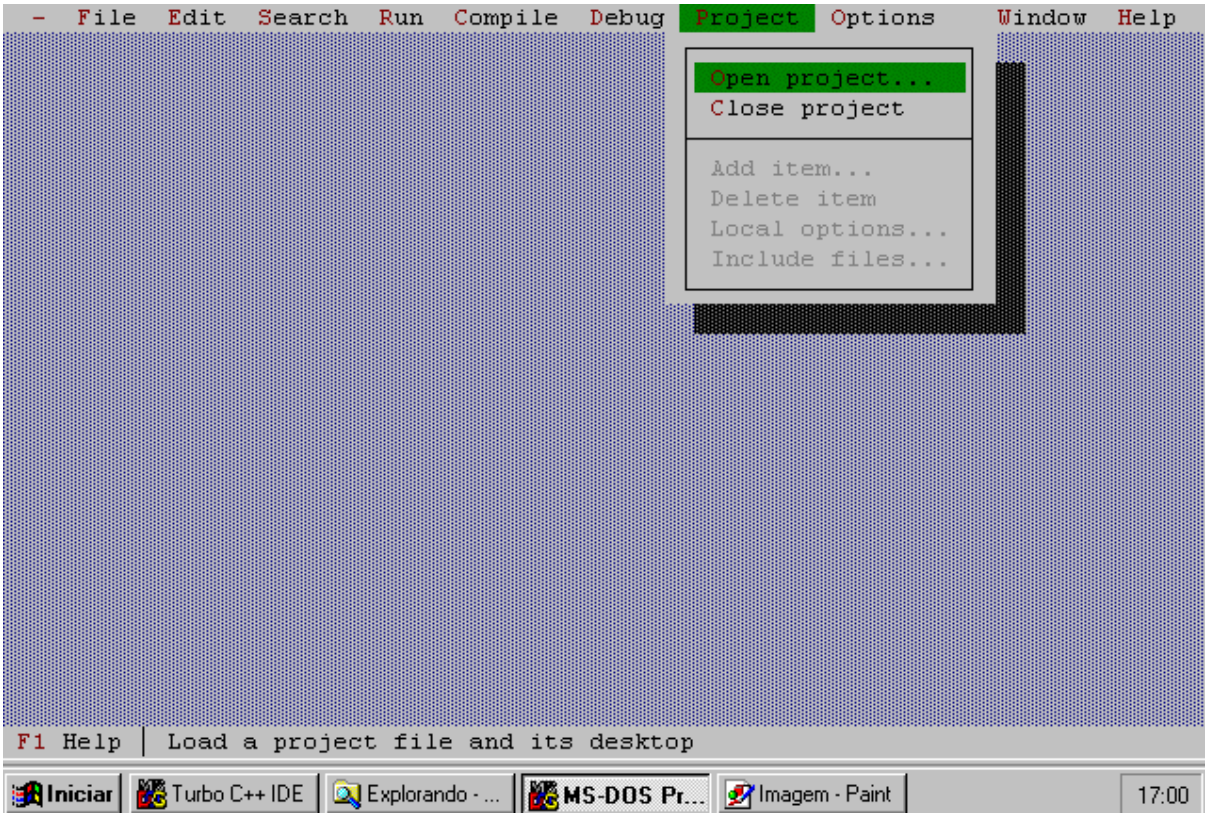

Figura 4.2: Primeira Tela para Execução do Software CHOPPV

Para se abrir a janela apresentada na cor cinza na figura 4.2 deve-se clicar uma vez com o botão direito do mouse sobre a palavra Project apresentada na parte superior da tela.

Passo 2: Selecionar o projeto que se deseja abrir dando um clique com o botão direito do mouse sobre o nome do projeto a que se deseja abrir e clicar uma vez com o botão direito do mouse na palavra *OK,* conforme ilustrado na figura 4.3.

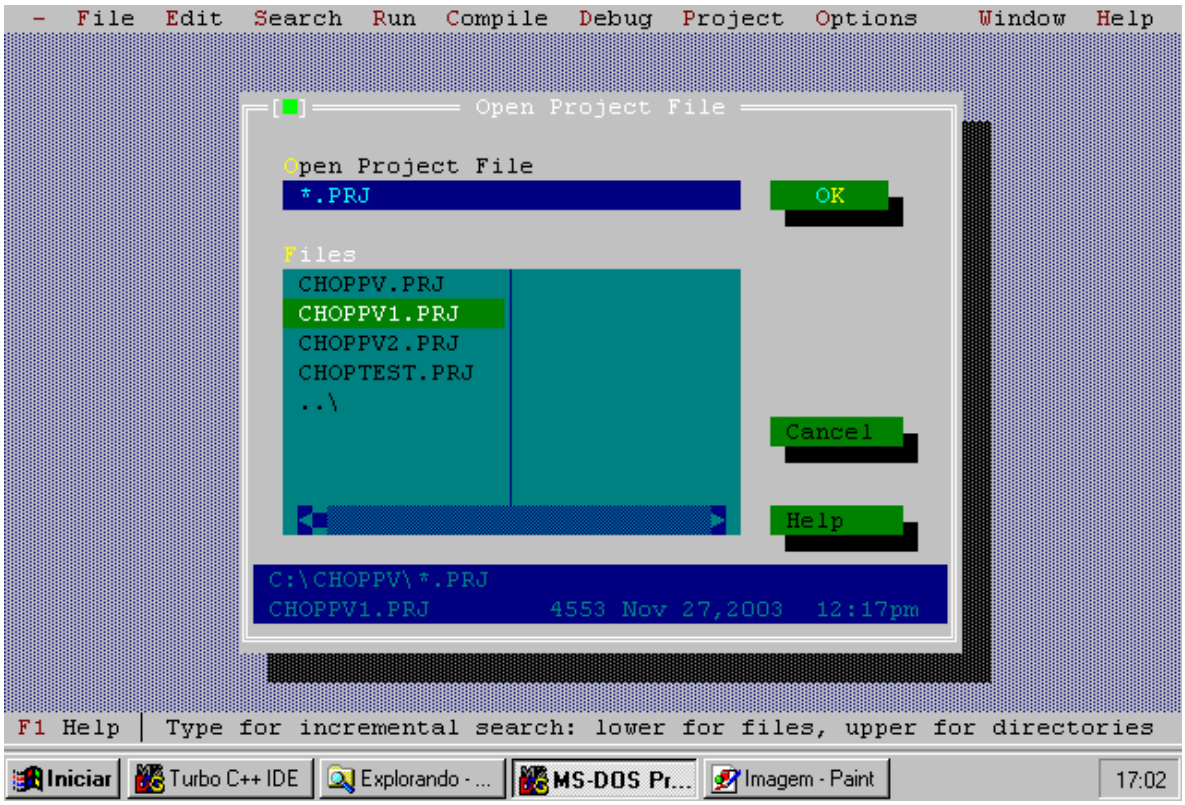

Figura 4.3: Segunda Tela para Execução do Software CHOPPV

A janela na cor cinza aparece na tela, ilustrada na figura 4.3, ao final das ações descritas no passo 1. Para se fazer a procura do projeto desejado deve-se usar as setas, como também o sinal "..\" ilustrados na parte inferior da tela verde da figura 4.3.

Passo 3: Se o programa não aparecer na tela ao fim do passo 2, selecionar o item Project no menu Window, conforme ilustrado na figura 4.4.

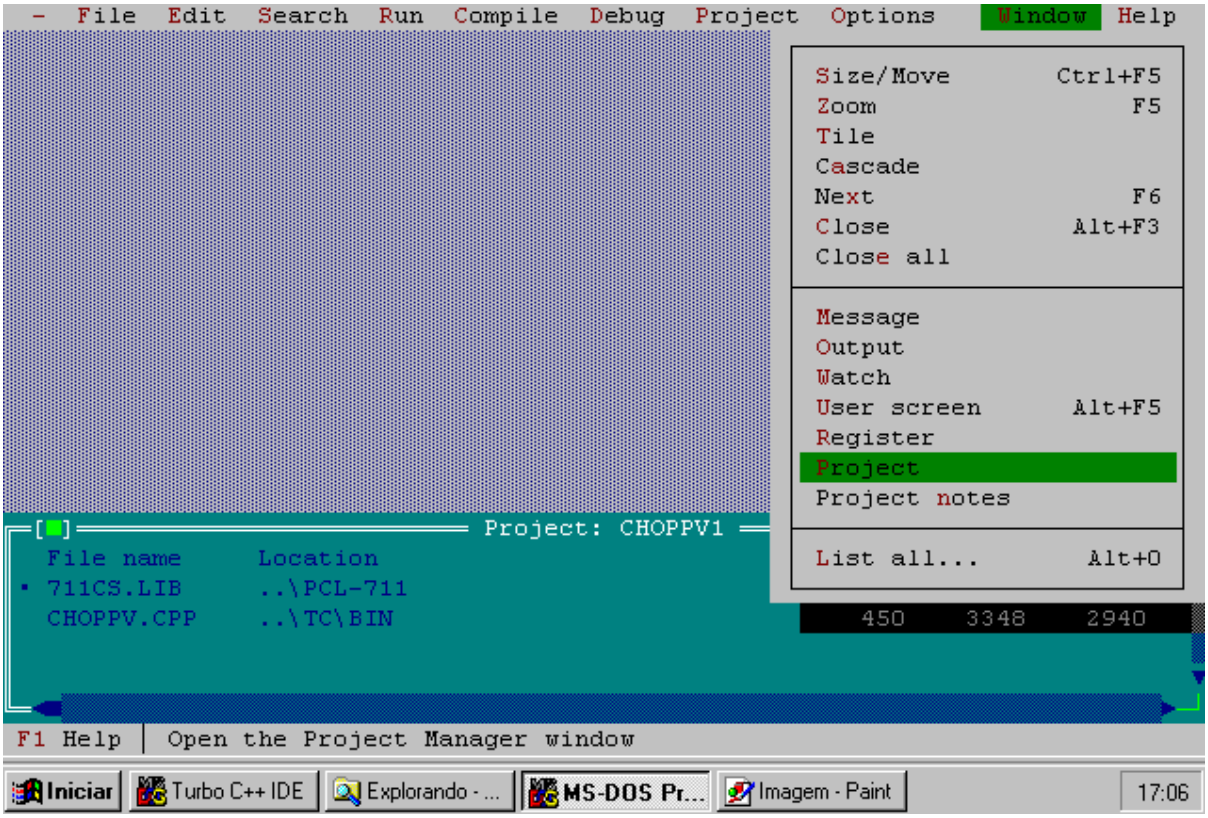

Figura 4.4: Terceira Tela de Execução do Software CHOPPV

A seleção do item Project do menu Window é feita arrastando-se o mouse sobre o item Project.

Passo 4: Conforme ilustrado na figura 4.5, selecionar o programa que se deseja abrir na janela verde apresentada na parte inferior da tela e clicar duas vezes com o botão direito do mouse sobre o nome do programa desejado.

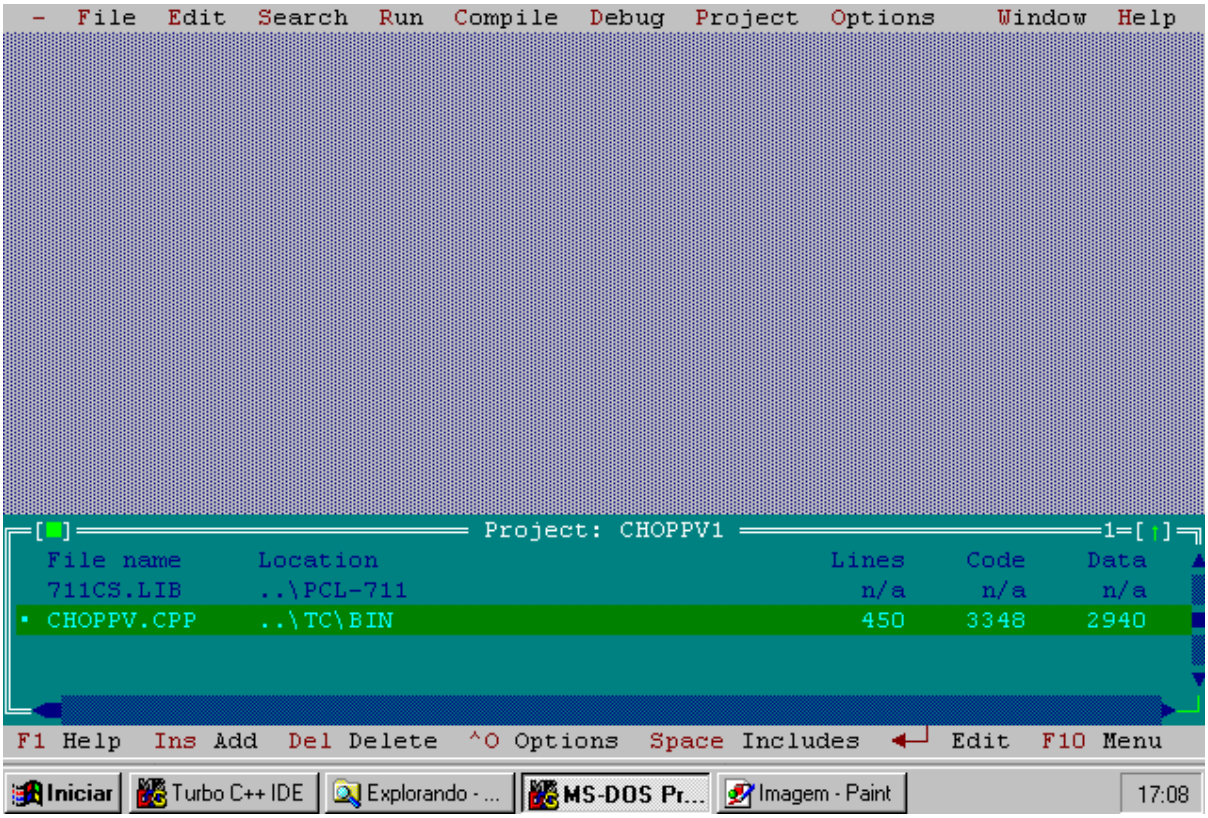

Figura 4.5: Quarta Tela de Execução do Software CHOPPV

Para seleção do programa desejado, arrasta-se o mouse sobre o nome do programa desejado clicando uma vez com o botão direito do mouse.

Ao fim do passo 2 ou ao fim do passo 4, o projeto se abrirá na tela conforme demonstrado na figura 4.6 abaixo:

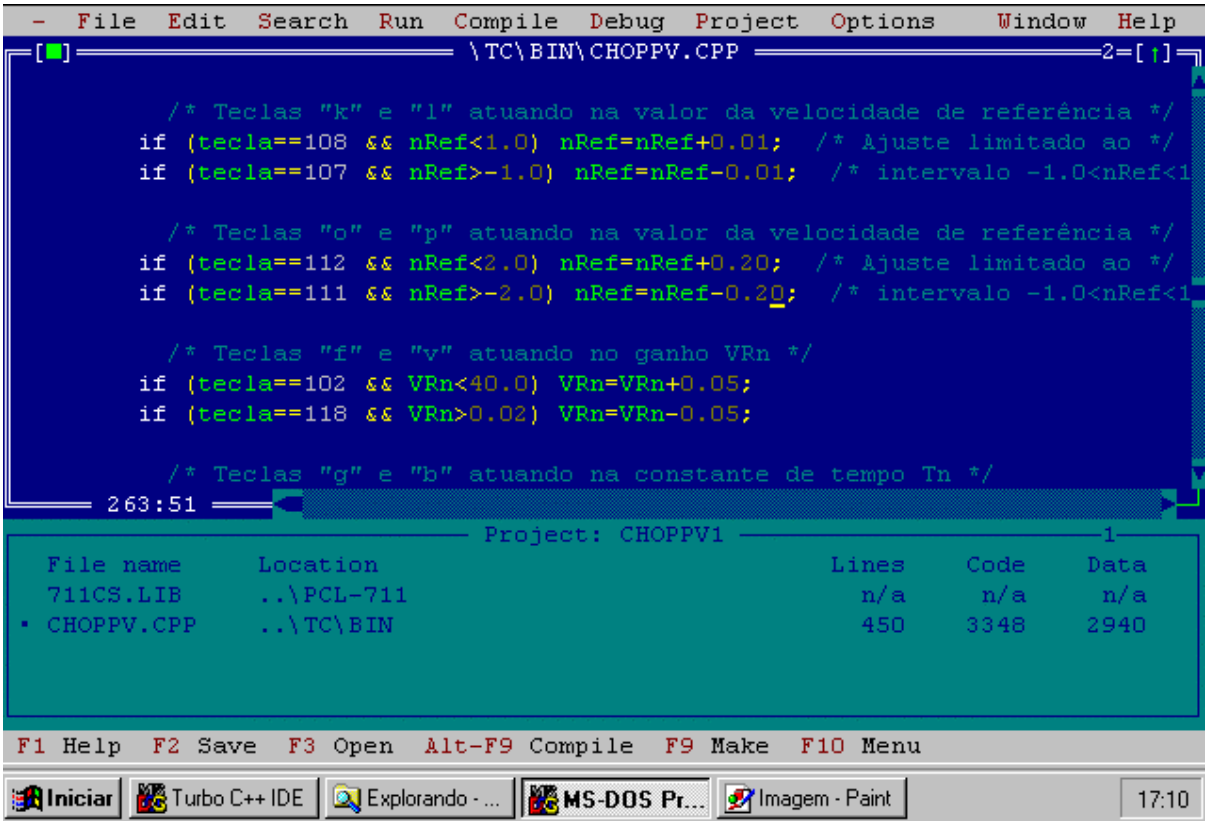

Figura 4.6: Quinta Tela de Execução do Software CHOPPV

Para maximizar a tela do programa em C e eliminar a tela verde, pressionar a tecla F5.

Passo 5: Para executar o software, seleciona-se o comando RUN ou pressiona-se as teclas CTRL e F9 simultaneamente, conforme demonstrado na figura 4.7 a seguir.

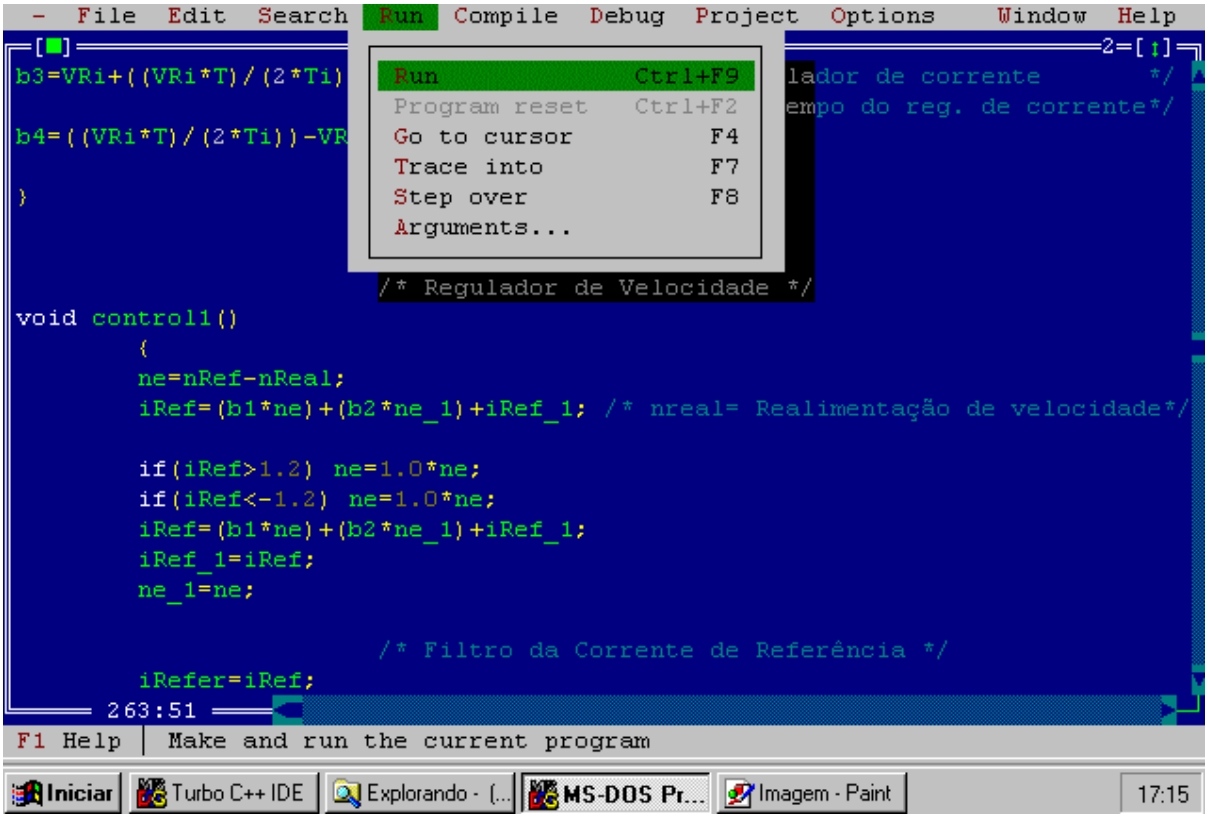

Figura 4.7: Sexta Tela de Execução do Software CHOPPV

Para selecionar o comando RUN deve-se arrastar o mouse sobre comando RUN e apertar o botão direito do mouse.

A seguir a Tela de Execução do Programa CHOPPV conforme ilustrado na figura 4.8.

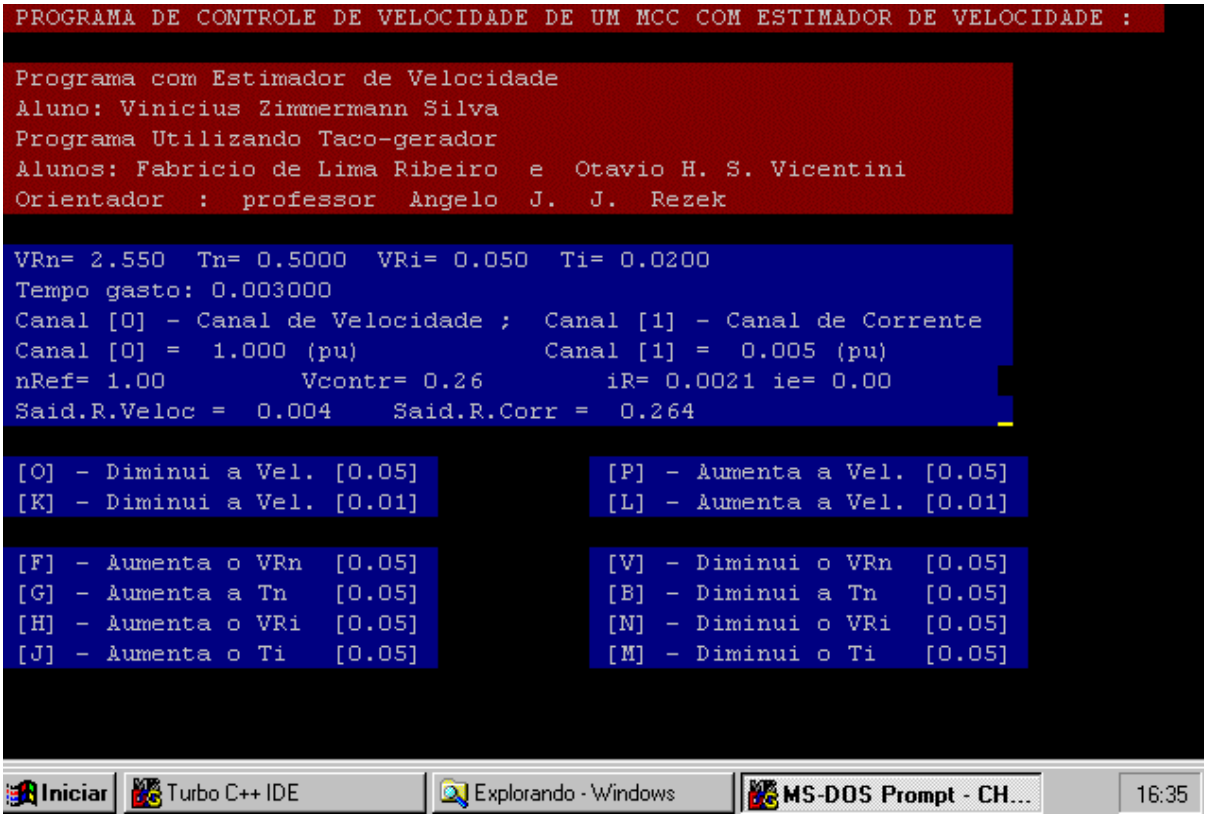

Figura 4.8: Tela de Execução do Software CHOPPV

VRn = Ganho do Regulador para o Sinal da Velocidade

Tn = Constante de tempo para a velocidade

Vri = Ganho do Regulador para o Sinal da Corrente

Ti = Constante de tempo da corrente

nRef. = Referência da Velocidade

Vcontr = Valor em PU(por unidade) da tensão de controle

IR = Referência de corrente

Ie = Erro de corrente

Said R. Veloc. =Saída resposta do regulador de velocidade

Said R. Corr. = Saída resposta do regulador de corrente

4.5 Energização do Circuito e Partida do Motor de Corrente Contínua

Para energização do circuito da armadura e partida do MCC devem ser executados os seguintes passos:

1 - Ligar o disjuntor geral da bancada e apertar o botão ligar;

2 - Ligar o disjuntor de alimentação do campo de excitação independente do motor de corrente contínua;

3 - Ligar o micro-computador;

4 - Ligar o sensor hall;

5 - Ligar o módulo de controle de pulsos e verificar se a chave manual/automático está na posição automático;

6 - Ligar a fonte de tensão do circuito de controle de tensão na carga;

7 - Ligar o módulo de controle de pulsos do circuito de controle de tensão na carga;

8 - Verificar se a chave de fechamento para entrada de carga está na posição aberto, mantê-la na posição aberta;

9 - Executar o software CHOPPV conforme instruções apresentadas neste capítulo 4;

10 - Para fazer a Partida do MCC, girar o reostato demarrador.

5 Sistemática Utilizada para Validação das Equações e Parâmetros Adotados para Estimação de Velocidade via Software

5.1 Introdução

Após a retirada do taco gerador, e inserção das equações para cálculo da velocidade real do motor (MCC) no software CHOPPV, ainda tornou-se necessário ajuste na constante de tempo da velocidade. Esta alteração foi devido a retirada de filtro para o sinal proveniente do taco gerador, destinado a placa de aquisição PCL 711. A referida constante de tempo foi alterada de Tn = 2.000, com taco, para Tn = 0.55, sem taco.

Esse ajuste foi feito através do software em C++, CHOPPV, em tempo real com o motor em funcionamento. Esses parâmetros são ajustados de forma a obter resposta dinâmica ótima do motor, ou seja, respostas com redução das oscilações de velocidade diante dos degraus de carga e degraus de velocidade referência.

Para obtenção da equivalência entre a velocidade referência ajustada na tela do programa e a velocidade real do eixo do motor fez-se pequenos ajustes na corrente de campo do motor e o valor fixado foi:

Iexc. = 0.25A = 250mA.

A maneira como se faz o ajuste em tempo real dos parâmetros acima mencionados está demonstrada na tela de execução do software CHOPPV ilustrada na figura 4.8.

5.2 Equações para Cálculo da Velocidade Real do Motor

A partir do equacionamento já demonstrado do MCC, será explicitado abaixo, a complementação das equações já demonstradas enfocando os ajustes que se tornaram necessários e demonstrando as opções de cálculo testadas e a que forneceu melhores resultados.

5.2.1 Cálculo da Força Contra-Eletromotriz

A força -contra –eletromotriz, fcem, do motor de corrente contínua, MCC, foi calculada como a seguir demonstrado. A fcem calculada abaixo, foi inserida no software CHOPP V como uma variável de entrada constante.

$$Va = En + Ra * Ia; \quad (5.1)$$

Onde:

Va = Tensão de alimentação do MCC medida na saída do retificador com carga nominal aplicada ao MCC e na velocidade nominal, 1PU;

Ra = Resistência da Armadura;

Ia = Corrente da Armadura.

Cujos valores medidos foram:

Va=209V;

Ra=3,5Ω;

Ia_n=7,7A. – Corrente da Armadura Nominal

Assim temos:

$$En = 209 - 3{,}5 * 7{,}7 \quad (5.2)$$

$$En = 182{,}05$$

Na tela do software CHOPPV temos:

long float eff=218.0, Va, En=182.05, Ra=3.5;

Onde:

eff = Tensão fase-fase medido na rede de alimentação.

5.2.2 Cálculo da Velocidade Real do Motor

5.2.2.1 Introdução

Nesta seção, o equacionamento para cálculo da velocidade real do motor será apresentado na linguagem matemática simples e na linguagem matemática do software CHOPP V. A seguir, é reapresentada a figura 3.10 ilustrativa do circuito implementado em laboratório, que será usado como referência para desenvolvimento das equações para estimação da velocidade.

5.2.2.2 Equacionamento Teórico

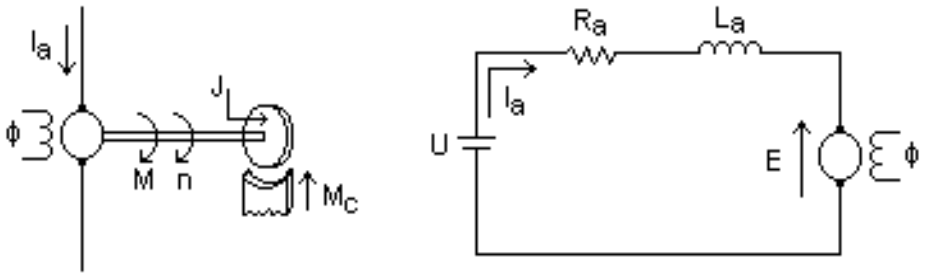

Figura 3.10 - Representação da Parte Mecânica e do Circuito Elétrico da Armadura do Motor CC com Excitação Independente

Va = Tensão de Alimentação do MCC

E = Força Contra Eletromotriz

Ra = Resistência da Armadura

Ia = Corrente da Armadura

En=Força Contra Eletromotriz Nominal

n_{pu}=Velocidade Real do MCC em PU

n= Velocidade Real do MCC

ϕ=Fluxo de Campo

M=Conjugado Motor

Mc=Conjugado de Carga

J=Momento de Inércia da Carga

$$Va = E + RaIa \quad (5.3) \therefore$$

$$\frac{Va}{En} = \frac{E}{En} + \frac{RaIa}{En} \quad (5.4)$$

Considerando que:

$E = k\phi n$ (5.5)

$En = k\phi n_n$ (5.6)

Substituindo (5.6) *e* (5.5) *em* (5.4) *temos*:

$$\frac{n}{n_n} = \frac{Va}{En} - \frac{RaIa}{En} \quad (5.7) \therefore$$

$$n_{PU} = \frac{Va}{En} - \frac{RaIa}{En} \quad (5.8)$$

Cálculo da Tensão de Alimentação do MCC – Va

$$Va = 1{,}35 * eff * \cos(\alpha) \quad (5.9)$$

Conforme filosofia do circuito de disparo apresentado na figura 3.2, temos:

$\alpha = Vcc * 180º$ (5.10) *ou*

$\alpha = Vcc * 3{,}14159274$ (5.11) $\therefore$

$$Va = 1{,}35 * eff * \cos(Vcc * 3{,}14159274) \quad (5.12)$$

eff = Tensão fase fase de alimentação da ponte retificadora

α = Ângulo de Disparo dos Tiristores

Vcc = Sinal de tensão gerado pelo software para controle do ângulo de disparo dos tiristores

Substituindo (5.12) em (5.8) temos:

Velocidade Real do MCC

$$n_{PU} = \frac{1{,}35 * eff * \cos(Vcc * 3{,}1415972)}{En} - \frac{RaIa}{En} \quad (5.13)$$

eff =Constante

En =Constante

Ra =Constante

5.2.2.3 Cálculo da Velocidade Real do Motor no Software CHOPPV

As equações a seguir serão descritas conforme apresentadas no software CHOPPV:

// Estimação da velocidade;

iReal=(DataBuf[1]/1.493); (5.14)

nReal=Va/En-((Ra*iReal*7.72)/En); //Calculo da Velocidade Real do MCC; (5.15)

Note que primeiramente é obtida da placa de aquisição a corrente real do motor, iReal, para, em seguida, calcular a velocidade real do motor nReal em PU(por unidade).

Sendo :

Va = Tensão de alimentação do MCC calculada via software CHOPPV;

En = Força contra eletro motriz calculada e colocada no software CHOPP V como variável de entrada constante;

Ia_n = 7,72A =Corrente da Armadura Nominal da Máquina.

Cálculo da Tensão de Alimentação do MCC– Va

No software CHOPPV, a equação para cálculo da tensão de alimentação Va foi dividida em duas expressões como a seguir demonstrada para facilitar o emprego e testes de outras expressões alternativas para o cálculo da Va.

d=cos(Vcc*3.14159274*1.089); //cosseno real - opcao 2; (5.16)

Va=1.35*eff*d; (5.17)

As equações 5.16 e 5.17 estão apresentadas conforme software CHOPPV.

Conforme figura 3.2, temos:

$$\alpha = Vcc*3.14159274, \qquad (5.18)$$

$$\text{pois } Vcc / 1PU = \alpha / \pi \qquad (5.19)$$

$$\therefore \alpha = Vcc*\pi \qquad (5.20)$$

Substituindo (5.20) em (5.16) tem-se

$$Va=1.35*eff*\cos(\alpha*1{,}089); \qquad (5.21)$$

A constante 1.089 foi utilizada para ajuste na equação 5.21. Este ajuste foi feito empiricamente mediante análise das respostas do motor aos comandos de aumento e diminuição da velocidade do motor via software. O aspecto da linearidade também foi analisado para ajuste desta constante, ou seja, a igualdade entre a velocidade de referência do software e a velocidade real para toda faixa de valores entre 0 e 1PU.

5.3 Equações Verificadas para o Cálculo de Va – Tensão de Alimentação do MCC

5.3.1 Introdução

Diante das dificuldades em se conseguir uma boa regulação da velocidade do motor frente aos degraus de velocidade referência e degraus de carga e, também, pela dificuldade em obter boa linearidade entre a velocidade referência e a velocidade real do motor, iniciaram-se os procedimentos de ajustes de parâmetros da equação para o cálculo de Va e ajustes dos parâmetros dos reguladores como Ganhos e constantes de tempo. Em não se obtendo um bom resultado inicial, partiu-se para outras alternativas para cálculo de Va, sendo elas: cálculo de Va por cosseno linearizado, cálculo de Va por equação polinomial a partir de medições em laboratório de Vcc e Va. A seguir estas alternativas serão apresentadas detalhadamente, assim como os respectivos resultados e o processo para obtenção de cada uma delas.

5.3.2 Alternativas para Cálculo de Va

5.3.2.1 Cálculo de Va utilizando cosseno linearizado

Com o objetivo de se conseguir resultados de Va mais adequados para uma boa regulação foi feita a tentativa da linearização do cosseno para obtenção de Va, tensão de alimentação do MCC.

A figura 5.1 ilustra o gráfico do cosseno real e do cosseno linearizado como também a equação da linearização utilizada no software CHOPPV.

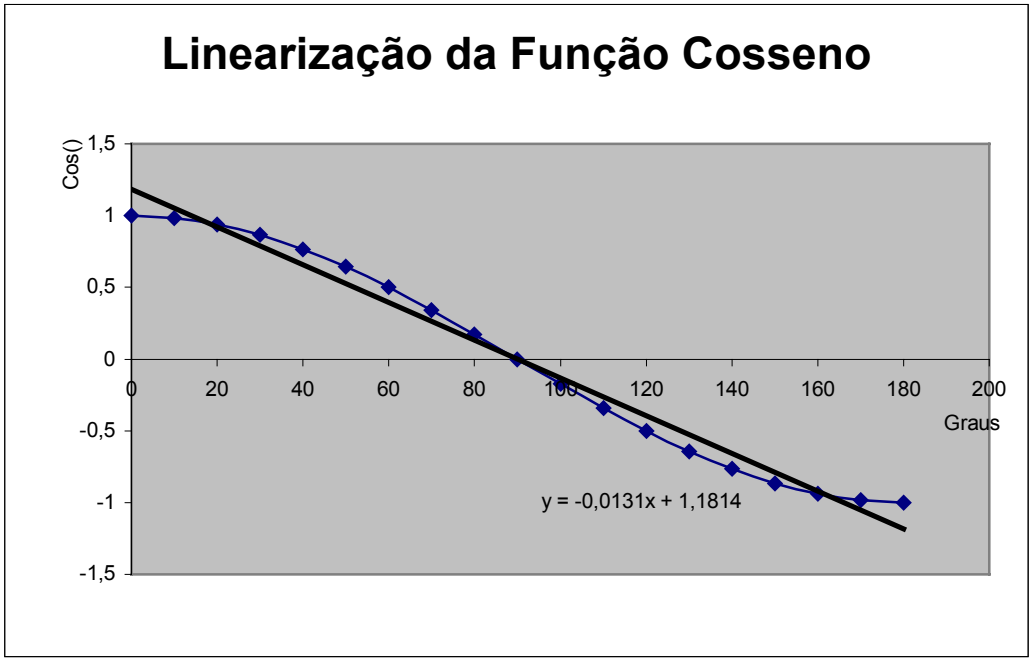

Figura 5.1 –Função Cosseno Linearizada

Como x = Vcc * 180 (5.22)

e y = cos (α) (5.23) tem –se:

cos (α) = 1.1814 – 0.0131*Vcc *180 (5.24)

Va = 1,35*eff*cos(α)linearizado; (5.25)

Va=1.35*eff*(1.1814-0.0131*Vcc*180); (5.26)

5.3.2.2 Cálculo de Va utilizando cosseno real

Esta foi a alternativa usada inicialmente, deixada de lado para teste de outras alternativas frente a dificuldade de ajuste dos parâmetros e, finalmente adotada como a melhor das alternativas como poderá ser constatado no próximo item, Análise das Alternativas.

Va = 1,35*eff*cos(α); (5.27)

Va = 1,35*eff*cos(Vcc*3.14159274); (5.28)

5.3.2.3 Equação polinomial para cálculo de Va

A partir do sinal de controle do ângulo de disparo dos tiristores, Vcc, e Tensão de Alimentação do MCC – Va, obteve-se a equação polinomial demonstrada na figura 5.2 a seguir, que foi inserida no software CHOPP V conforme equacionamento demonstrado a seguir. Os sinais de Vcc e Va foram obtidos a partir do ajuste manual do Vcc através do módulo de controle de pulsos, após a inabilitação do software CHOPPV.

Na figura 5.2 está ilustrado o gráfico demonstrativo da equação polinomial obtida a partir da aquisição do sinal Vcc, proveniente do módulo de controle de pulsos, e do sinal da tensão de alimentação do MCC, Va.

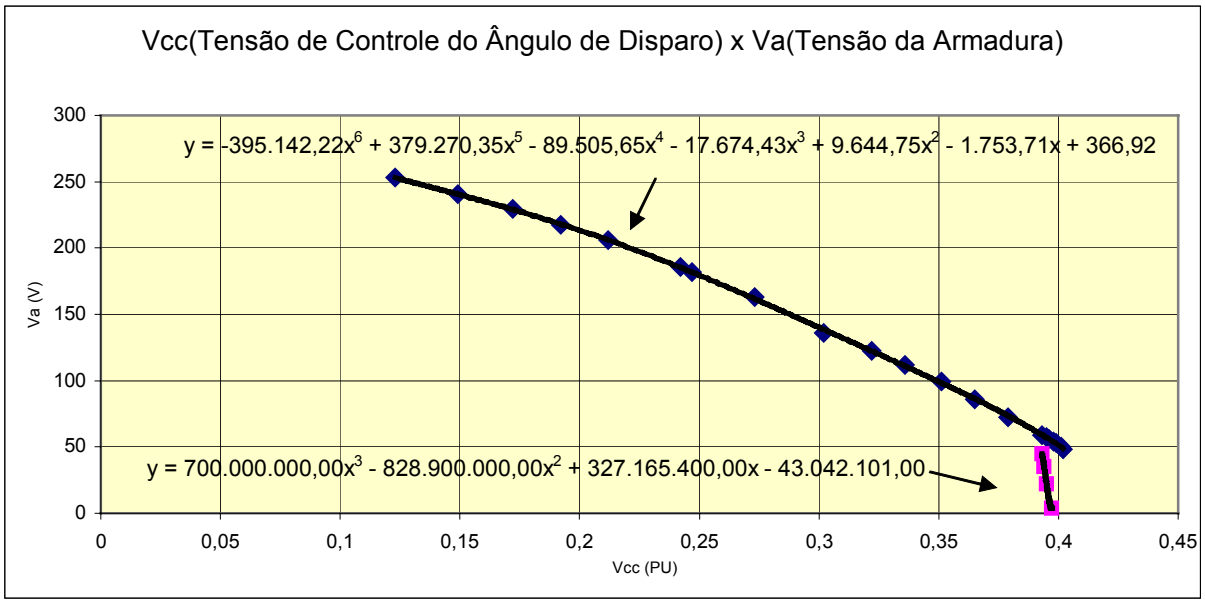

Figura 5.2: - Equação Polinomial Va=f(Vcc)

Conforme programação em linguagem C++:

Va=-395142.22*Vcc*Vcc*Vcc*Vcc*Vcc*Vcc+379270.35*Vcc*Vcc*Vcc*Vcc*Vcc-89505.65*Vcc*Vcc*Vcc*Vcc-17674.43*Vcc*Vcc*Vcc+9644.75*Vcc*Vcc-1753.71*Vcc+366.92; (5.29)

// if ((Vcc>=0.393)) (5.30)

Va=700000000.00*Vcc*Vcc*Vcc-828900000.00*Vcc*Vcc+327165400.00*Vcc-43042101.00; (5.31)

if (Va<=0.0) (Va=0.0); (5.32)

5.3.3 Análise das Alternativas para Cálculo de Va

As três alternativas descritas anteriormente foram testadas e seus resultados foram analisados objetivando verificar o desempenho do motor frente aos degraus de carga, degraus de velocidade e linearidade da velocidade referência frente a velocidade real do motor. A seguir, serão apresentadas as análises feitas dos resultados de desempenho do motor frente a cada uma delas.

Para a alternativa 1 – Cálculo de Va utilizando cosseno linearizado - verificou-se que o motor não apresentou boa dinâmica frente aos degraus de velocidade e carga, e não houve boa linearidade frente as mudanças na velocidade de referência, ou seja, a velocidade real do motor não correspondia a velocidade de referência ajustada no programa para toda a faixa de velocidade analisada (0 a 1PU).

Para a alternativa 2 – Cálculo de Va utilizando cosseno real – esta foi a melhor das alternativas analisadas, o controle de velocidade do motor apresentou boa dinâmica e boa linearidade. Se inicialmente fosse feito o correto ajuste na constante multiplicadora do ângulo alfa, 1,089, e o ajuste correto somente na constante de tempo da velocidade, 0,50, Tn, não seria necessário o teste de outras alternativas como foi feito.

Para a alternativa 3 – Cálculo de Va utilizando equação polinomial – esta apresentou melhores resultados que a alternativa 1, porém para alguns valores de velocidade de referência, não houve equivalência com a velocidade real.

Abaixo, é apresentado gráfico demonstrativo comparativo das alternativas 2 e 3 sob a ótica de qual delas se aproximou mais dos valores teóricos ideais de velocidade:

A figura 5.3 ilustra o gráfico comparativo para a Alternativa 2:

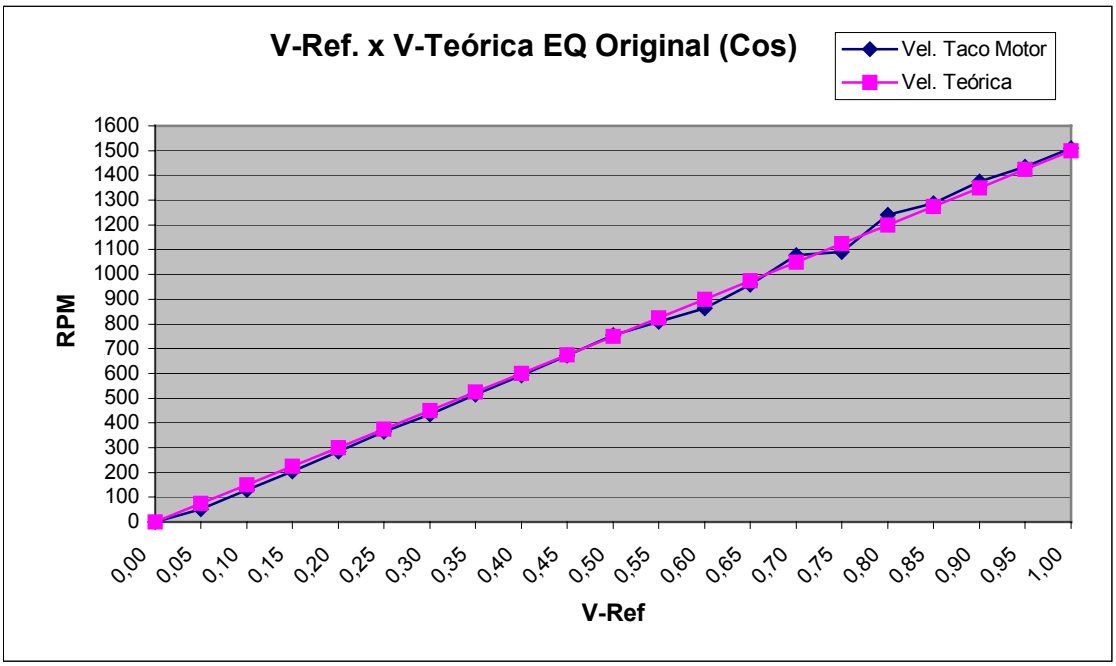

Figura 5.3 – Equação com cosseno real para Cálculo de Va - Alternativa 2

A figura 5.4 ilustra o gráfico comparativo para alternativa 3:

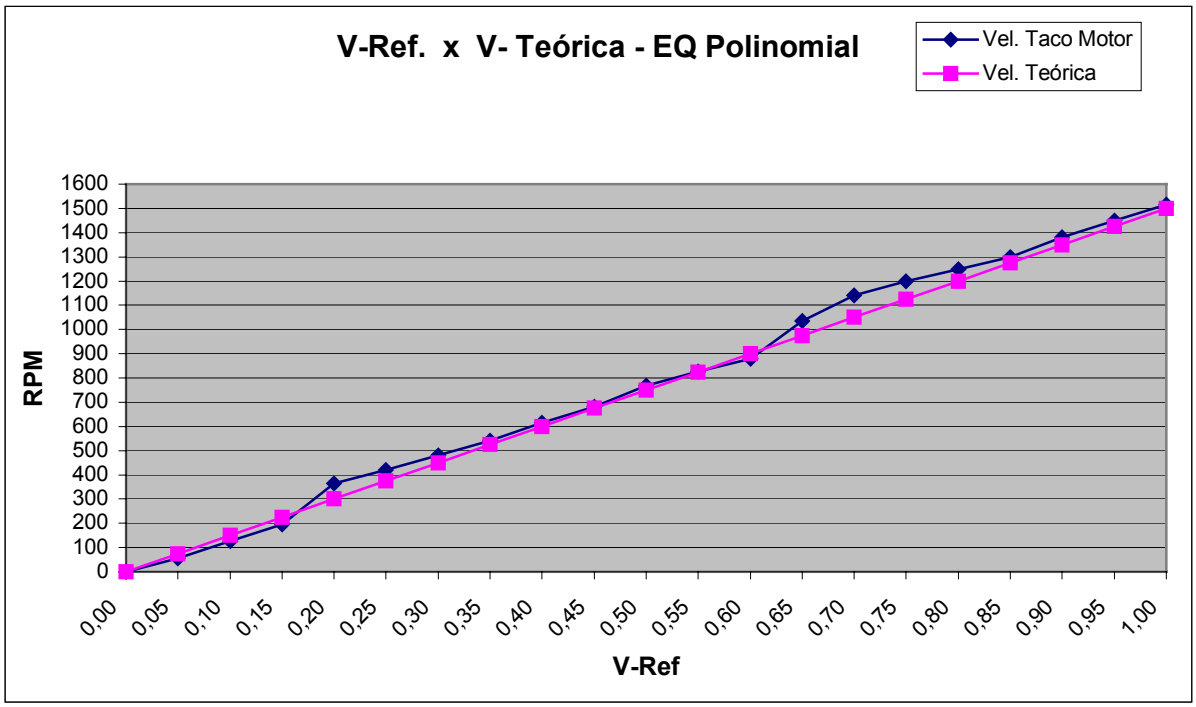

Figura 5.4 – Cálculo de Va por Equação Polinomial - Alternativa 3

Como pode ser verificado nestes dois gráficos acima, a alternativa 2 se apresentou com os melhores resultados, ou seja, obtenção de Va por cosseno real. .

6 Análise dos Resultados Obtidos

A seguir, serão apresentados os resultados obtidos pelo software de controle de velocidade com estimador de velocidade, CHOPPV, frente aos degraus de carga e velocidade referência, como também o comportamento do motor diante da partida. Para facilitar a análise serão apresentados conjuntamente os resultados obtidos com estimador, e os resultados obtidos com o taco para cada evento.

Pôde-se observar a rápida resposta do sistema em termos de velocidade para variações na carga e na referência (nref). Para entrada brusca de carga nominal (7,7A) a estabilização da velocidade se deu em aproximadamente 2,5[s](Fig. 6.2) para sistema de controle de velocidade utilizando taco e, em 2,0 [s], quando foi utilizado estimador de velocidade (figura 6.1). No caso da aplicação de degrau negativo de 0,2 [pu] na referência de velocidade, a estabilização da velocidade ocorreu em aproximadamente 2,5[s] quando da utilização de taco gerador (figura 6.9) e em 2,0[s], quando da utilização de Estimador de Velocidade (figura 6.10).

Para análise dos dados abaixo, considerar:

Corrente:

1 Divisão vertical = 4,6 [A]

Velocidade:

1 Divisão vertical = 500 [rpm].

6.1 Partida do Motor

A figura 6.1 ilustra as formas de onda de tensão e corrente para Partida sem taco gerador:

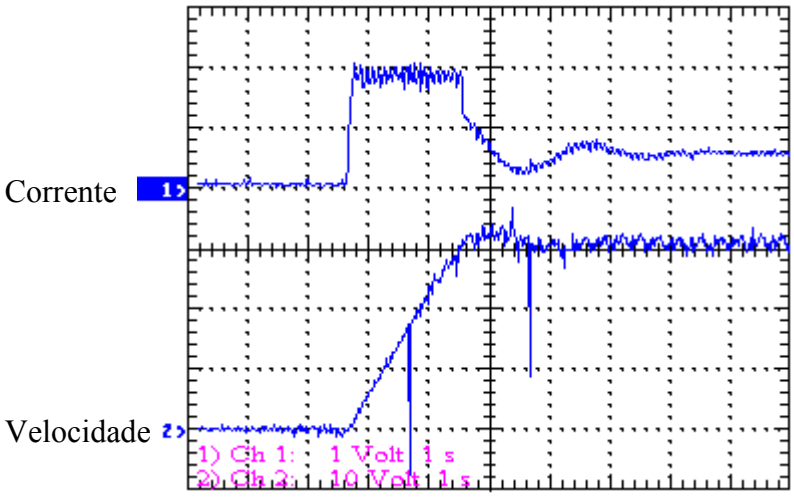

Figura 6.1 – Partida com Estimador de Velocidade

A figura 6.2 ilustra as formas de onda de tensão e corrente para Partida com o taco:

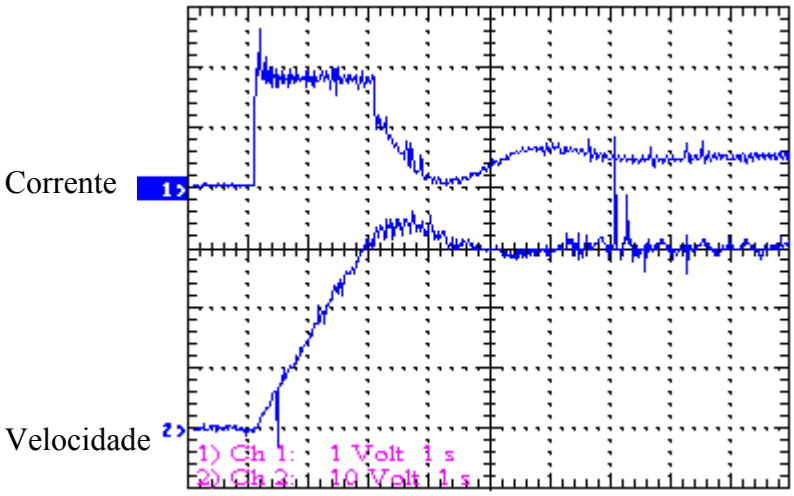

Figura 6.2: Partida com Taco

6.2 Degrau de carga (6A)

A – Degrau de carga (6 A) sem taco gerador:

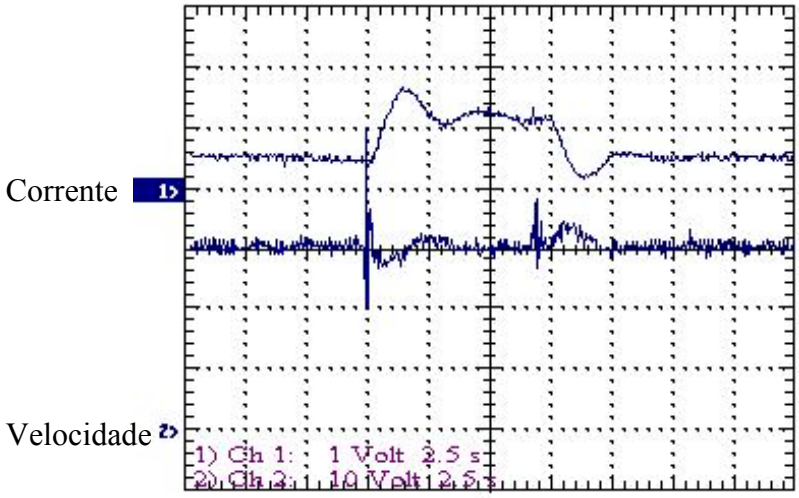

Figura 6.3 Degrau de Carga (6A) com Estimador de Velocidade

B- Degrau de carga (6A) com o taco gerador:

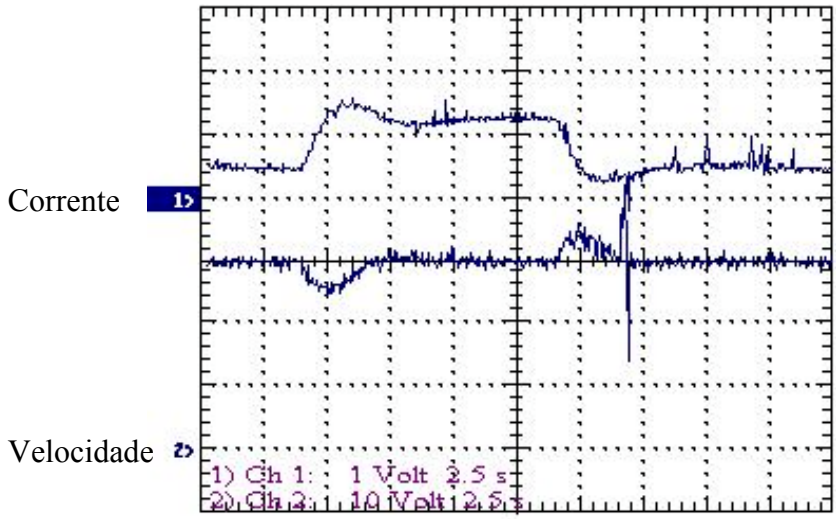

Figura 6.4: Degrau de Carga (6A) com Taco

Para ambos os casos, estando o motor operando a vazio, aplicou-se carga parcial de 6A e após alguns segundos retirou-se essa carga bruscamente e verificou-se a regulação de velocidade tanto com a utilização de taco gerador quanto com a utilização de estimador de velocidade.

6.3 Degrau de carga (7,7A)

A-Degrau de carga nominal (7,7 A) sem o taco

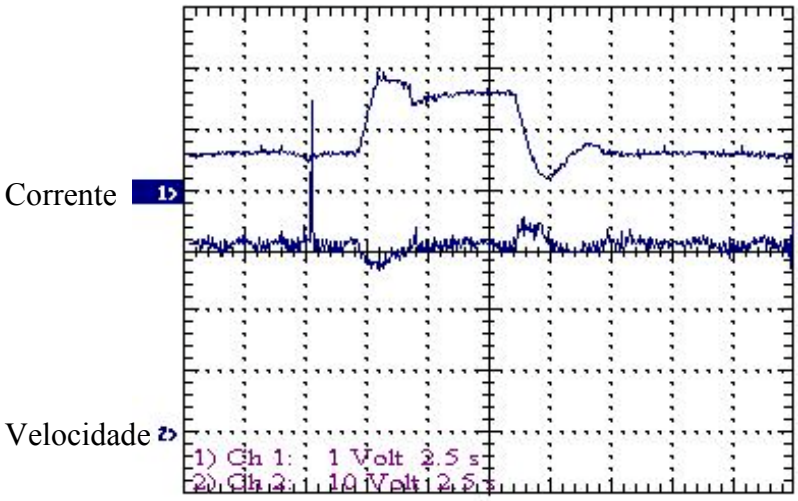

Figura 6.5: Degrau de Carga Nominal com Estimador de Velocidade

B- Degrau de carga nominal (7,7 A) com o taco gerador:

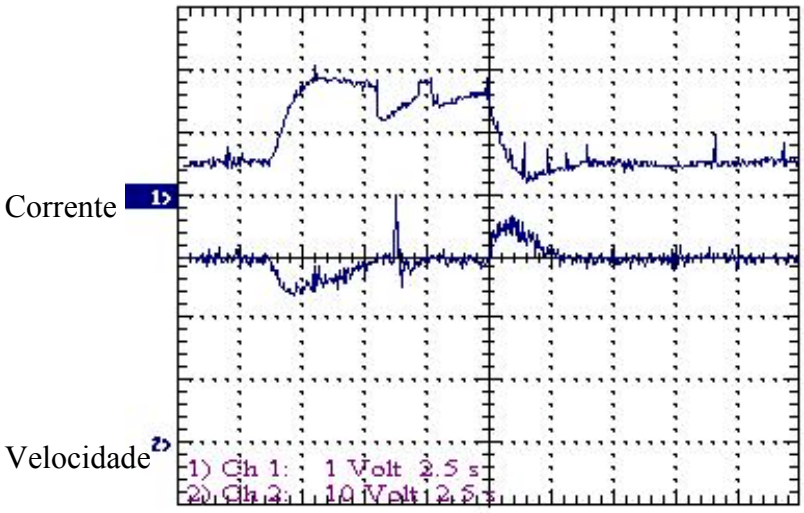

Figura 6.6: Degrau de Carga Nominal com Taco

Para ambos os casos, estando o motor operando a vazio, aplicou-se carga nominal de 7,7A e após alguns segundos retirou-se essa carga bruscamente e verificou-se a regulação de velocidade tanto com a utilização de taco gerador quanto com a utilização de estimador de velocidade.

6.4 Degrau de Velocidade de Referência de 0,2PU a Carga de 6A

A- Degrau de Nref. de 0,2 PU a carga de 6A sem o taco

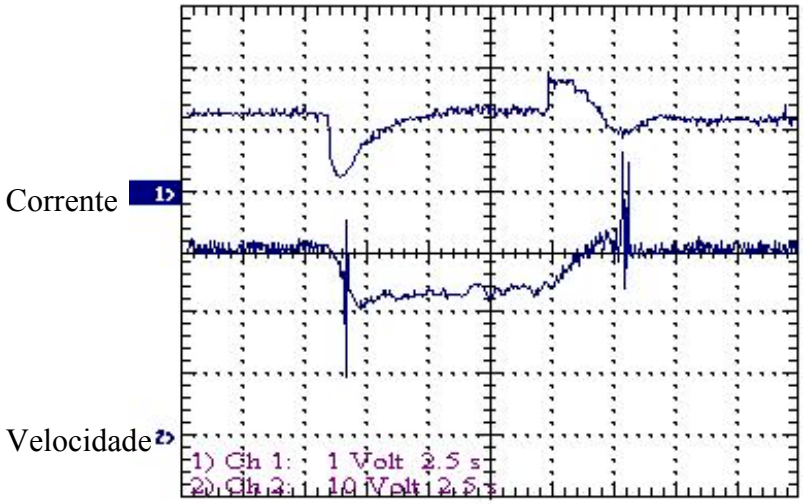

Figura 6.7: Degrau de Nref. de 0,2PU a Carga de 6A com Estimador

B- Degrau de Nref. de 0,2 PU a carga de 6A com o taco gerador

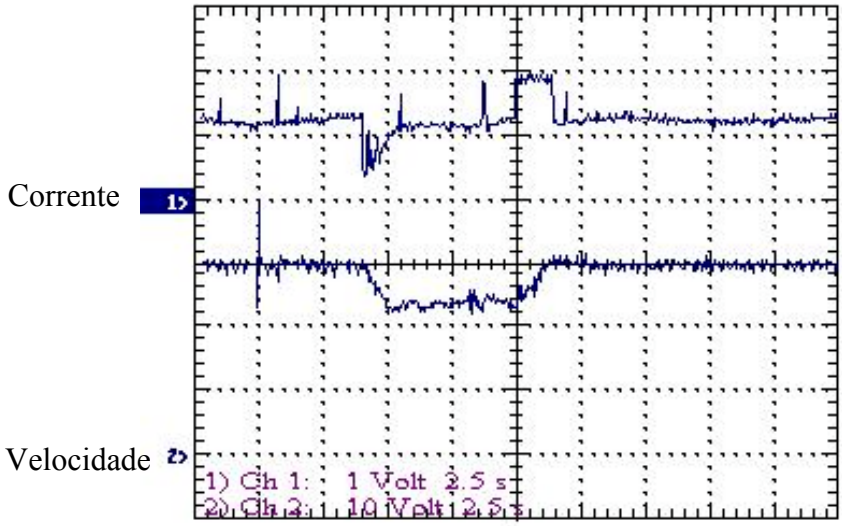

Figura 6.8: Degrau de Nref. de 0,2PU a Carga de 6A com Taco

Para ambos os casos, aplicou-se um degrau negativo de 0,2PU na velocidade de referência, e após alguns segundos aplicou-se um degrau positivo de 0,2PU quando a velocidade retornou ao valor original. Assim, verificou-se a regulação de velocidade tanto com a utilização de taco gerador quanto com a utilização de estimador de velocidade.

6.5 Degrau de Velocidade de Referência de 0,2PU a Carga de 7,7A

A- Degrau de Nref de 0,2PU a carga de 7,7A sem o taco

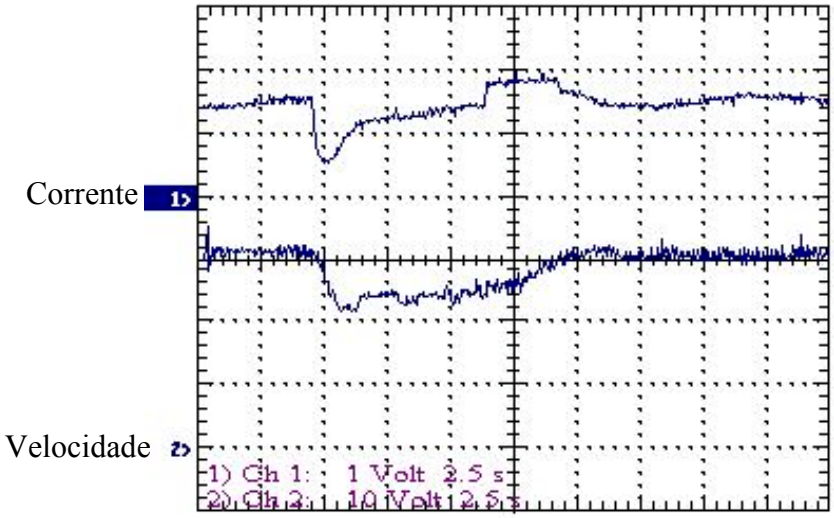

Figura 6.9: Degrau de Nref. de 0,2PU a Carga de 7,7A com Estimador de Velocidade

B- Degrau de Nref de 0,2PU a carga de 7,7A com o taco gerador

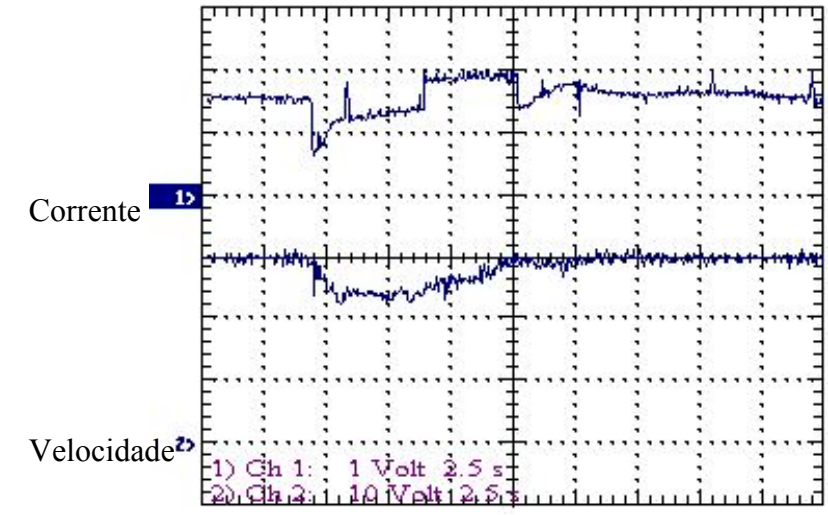

Figura 6.10: Degrau de Nref. de 0,2PU a Carga de 7,7A com Taco

Para ambos os casos, estando o motor operando com carga nominal de 7A, aplicou-se um degrau negativo de 0,2PU na velocidade de referência, e após alguns segundos aplicou-se um degrau positivo de 0,2PU quando a velocidade retornou ao valor original. Assim, verificou-se a regulação de velocidade tanto com a utilização de taco gerador quanto com a utilização de estimador de velocidade.

7 Conclusão

Os resultados experimentais obtidos em laboratório foram satisfatórios, pois o controle digital com Estimador de Velocidade para o acionamento de uma máquina de corrente contínua com excitação independente se mostrou estável e eficiente, tendo uma resposta mais rápida do que aquela apresentada pelo controle digital com Taco Gerador.

Ao se aplicar carga no motor sendo esta ajustada tanto para carga nominal de 7,7A, quanto para carga parcial de 6A, verificou-se que a velocidade, em um primeiro momento caiu, para posteriormente voltar ao patamar original.

Verificou-se, portanto, uma boa regulação da velocidade quando da aplicação de carga.

Do mesmo modo, estando o motor com a carga nominal de 7,7A, ou a carga parcial de 6A, aplicando-se um degrau negativo de 0,2PU na velocidade referência, verificou-se uma queda de 20% em relação a velocidade original, mantendo-se neste valor até a aplicação do degrau positivo de 0,2PU quando retornou a velocidade original. Estes resultados, portanto, representam um sistema de controle em malha fechada com bom sinal de velocidade de realimentação que, neste caso, foi estimada.

Para o desenvolvimento do estimador de velocidade do MCC foi necessário, além das equações provenientes do modelamento matemático do MCC e Retificador, ajuste na constante de tempo do regulador de velocidade, Tn, de 2,0 com taco para 0,50 sem o taco. Este ajuste foi conseqüência da retirada do filtro do sinal de velocidade proveniente do taco gerador.

Conforme análise laboratorial do erro intrínseco do estimador, cuja representação está ilustrado no gráfico da figura 5.3, e associada às informações obtidas das notas de aulas do 2ºsemestre de 2000 do curso de Acionamentos Elétricos Controlados, elaborada pelo Professor Nery de Oliveira Junior da Universidade Federal de Itajubá, afirmou-se que as três formas de aquisição de velocidade real do eixo do motor são: aquisição de velocidade por taco digital, aquisição de velocidade por taco analógico e aquisição de velocidade por estimador de velocidade. Os erros intrínsecos a cada uma destas formas de aquisição são: 0,01% para taco digital, 0,1% para taco analógico e 1% para estimador de velocidade.

CONCLUSÃO

Diante dos bons resultados apresentados pelo estimador de velocidade, apresenta-se também mais uma alternativa para controle de velocidade real de motor de corrente contínua. Como também, abre-se espaço para a melhoria deste estimador, visando obter maiores níveis de precisão a fim de possibilitar maior viabilidade de aplicação.

Apêndice A – Montagem em Laboratório

A seguir serão apresentados figuras e esquemas representativos dos circuitos elétricos montados em laboratório.

A figura A.1 ilustra o esquema de Ligação Geral com Estimador de Velocidade:

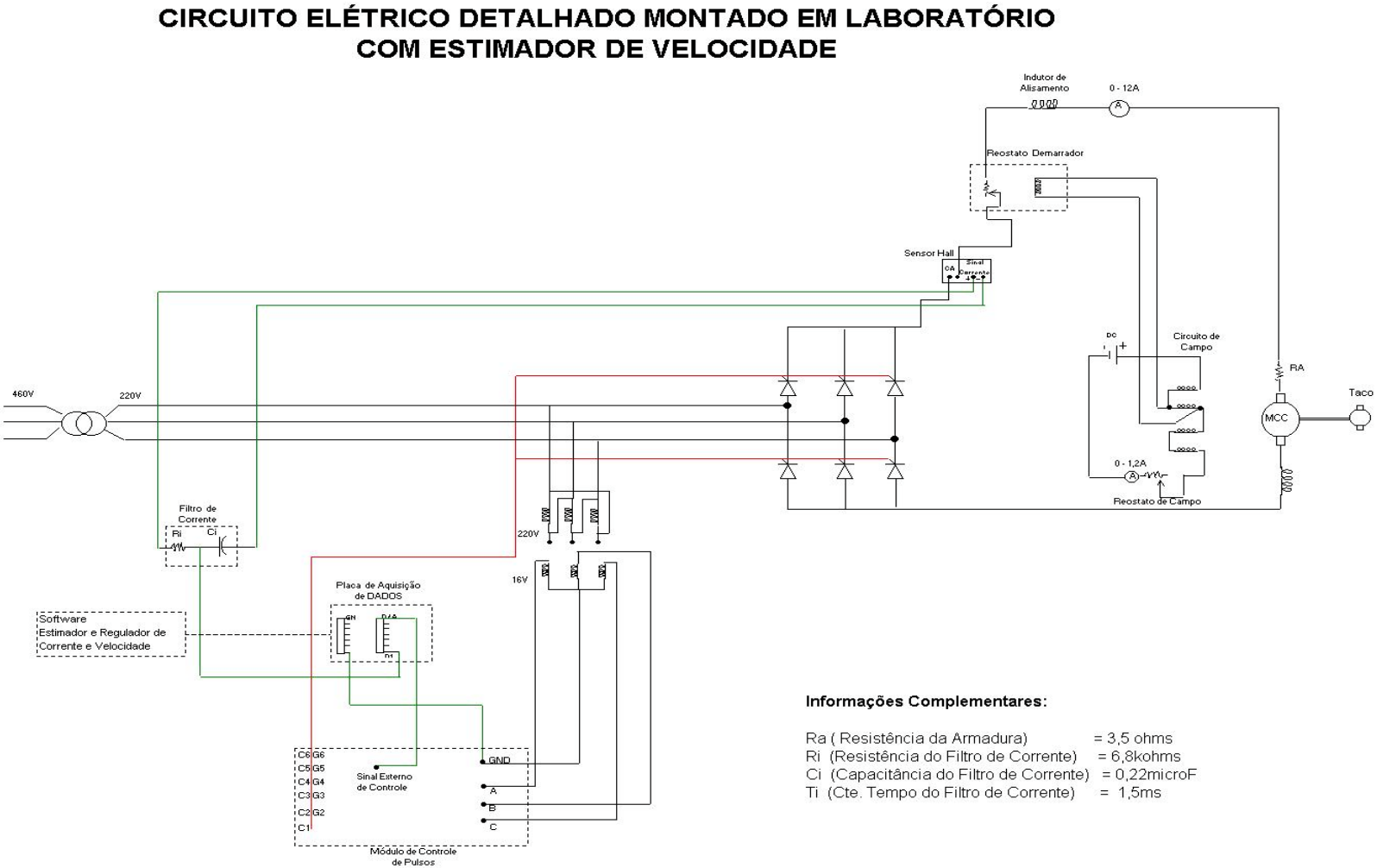

Figura A.1 – Esquema Elétrico de Ligação Geral com Estimador de Velocidade

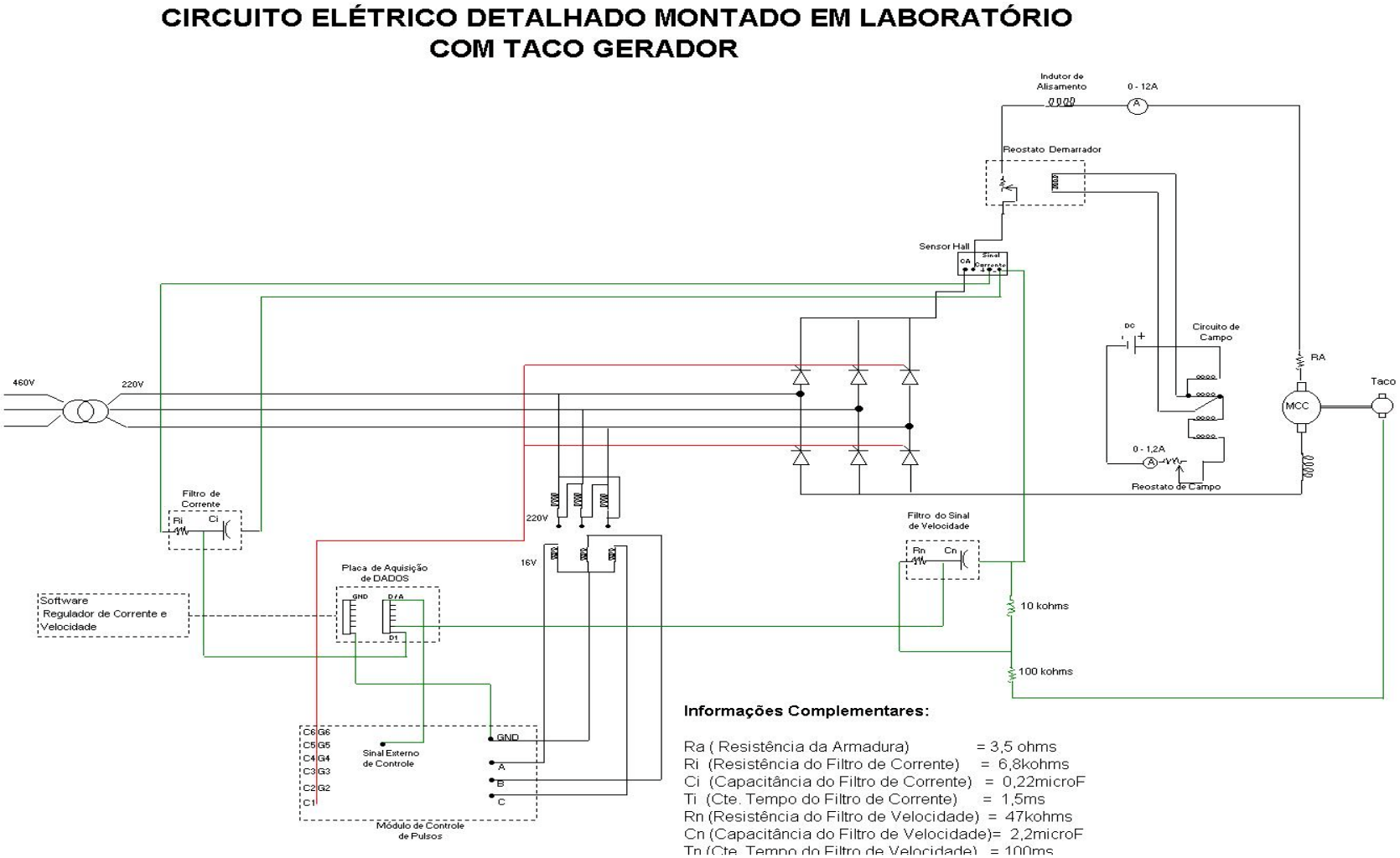

Figura A.2 – Esquema de Ligação Geral com Taco Gerador-

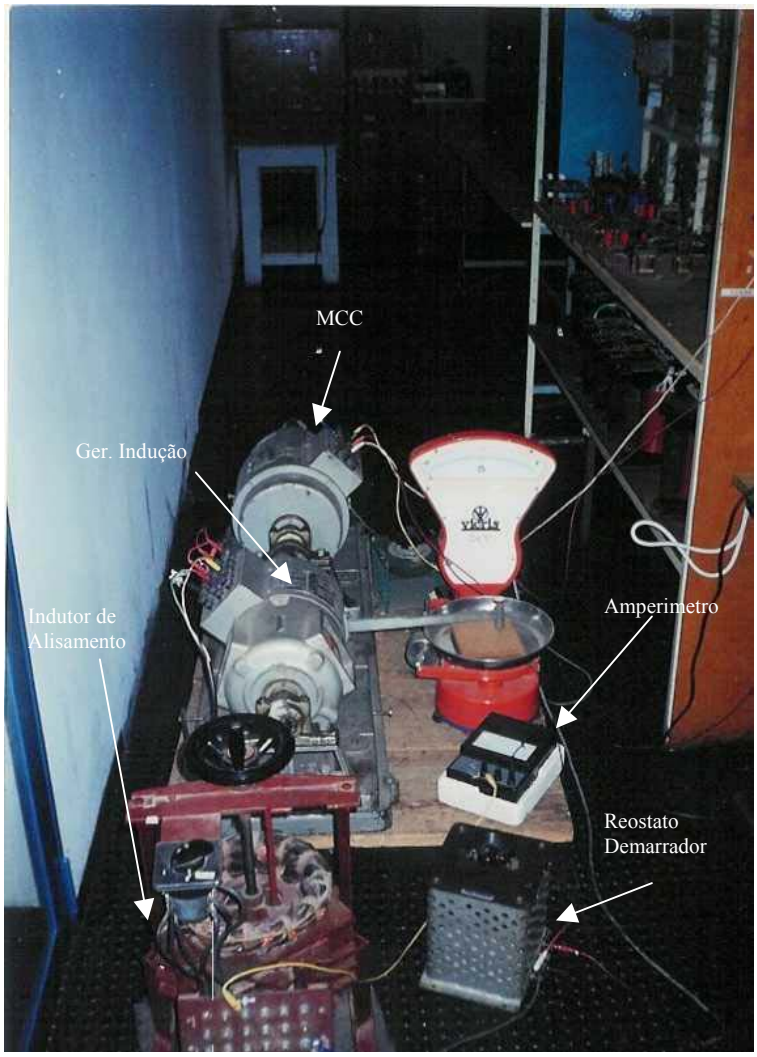

Figura A.3 – MCC – Ger. Indução – Indutor e Instrumentos de medida

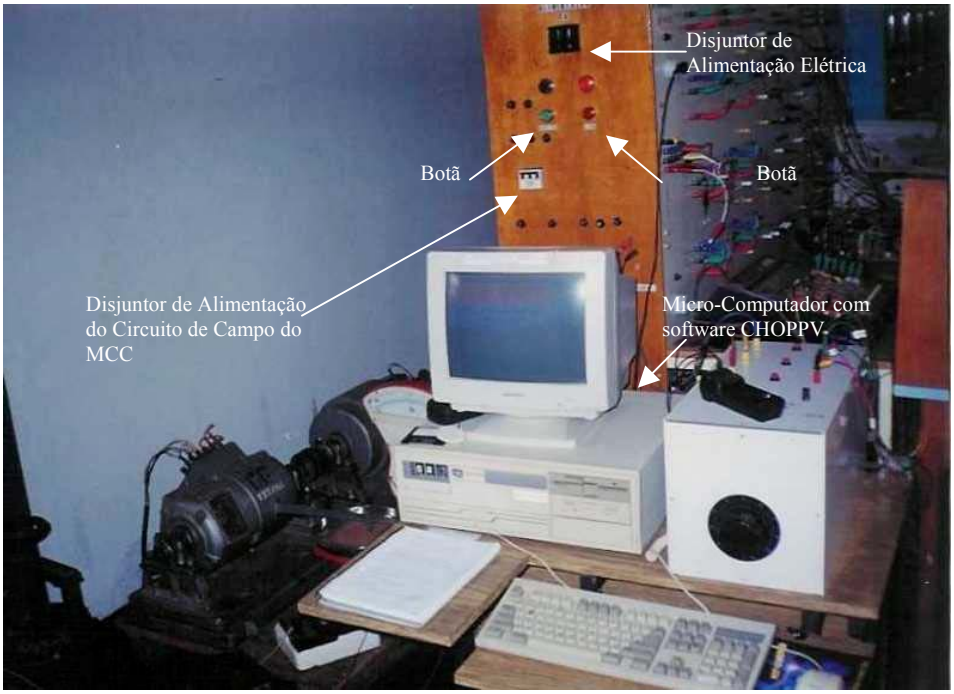

Figura A.4 – Painel de Energia e Micro-computador

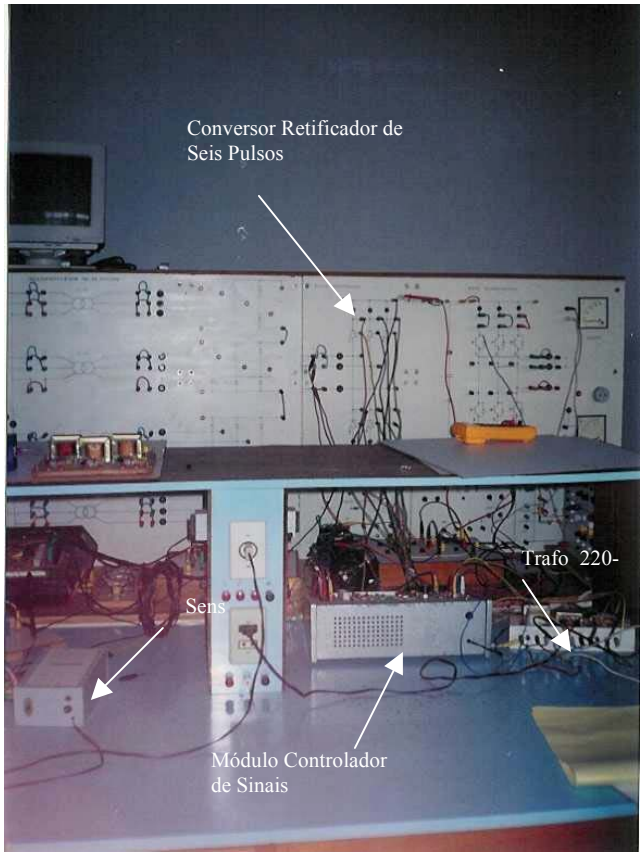

Figura A.5- Vista da Bancada

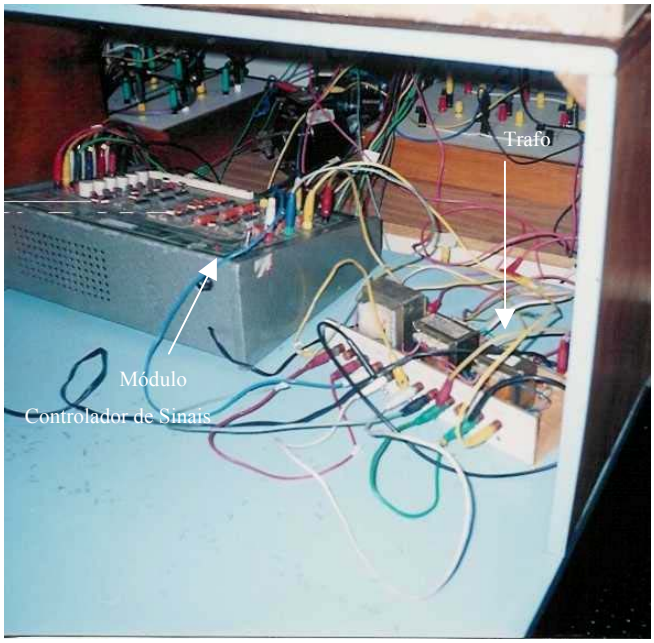

Figura A.6 – Detalhe do Módulo de Controle de Sinais do Conversor

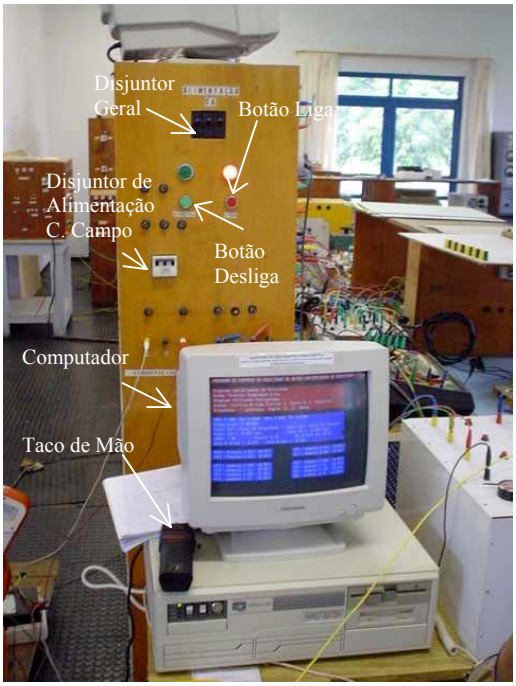

Figura A.7 – Micro computador e Painel de Alimentação do Circuito

Figura A.8 – MCC, Taco Motor, Gerador, Reostato e Amperimetro

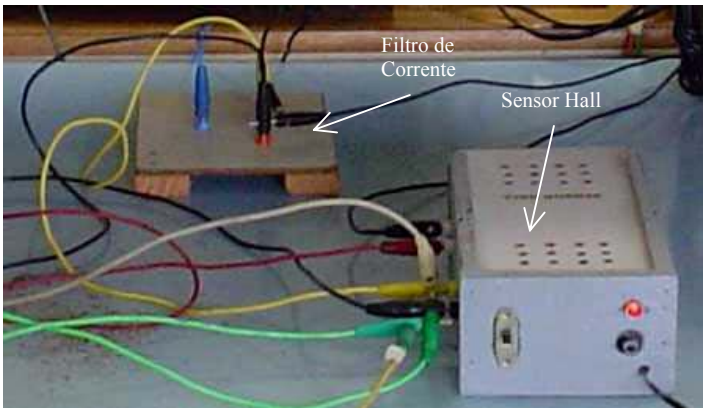

Figura A.9 – Sensor Hall e Filtro de Corrente

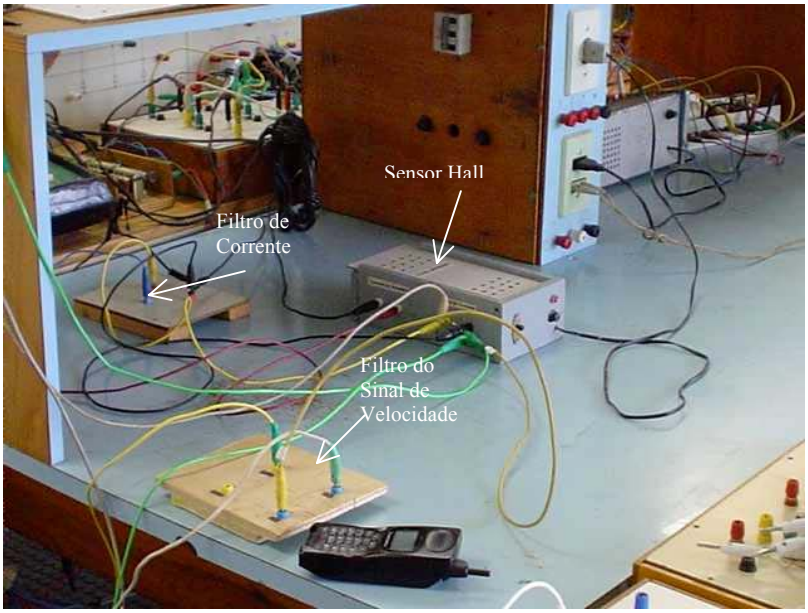

Figura A.10 – Configuração com utilização do Taco

Figura A.11 – Aquisição de Formas de Onda

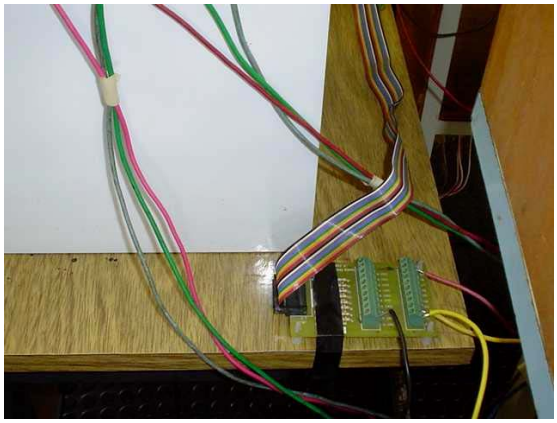

Figura A.12 – Placa de Aquisição de Dados PCL 711

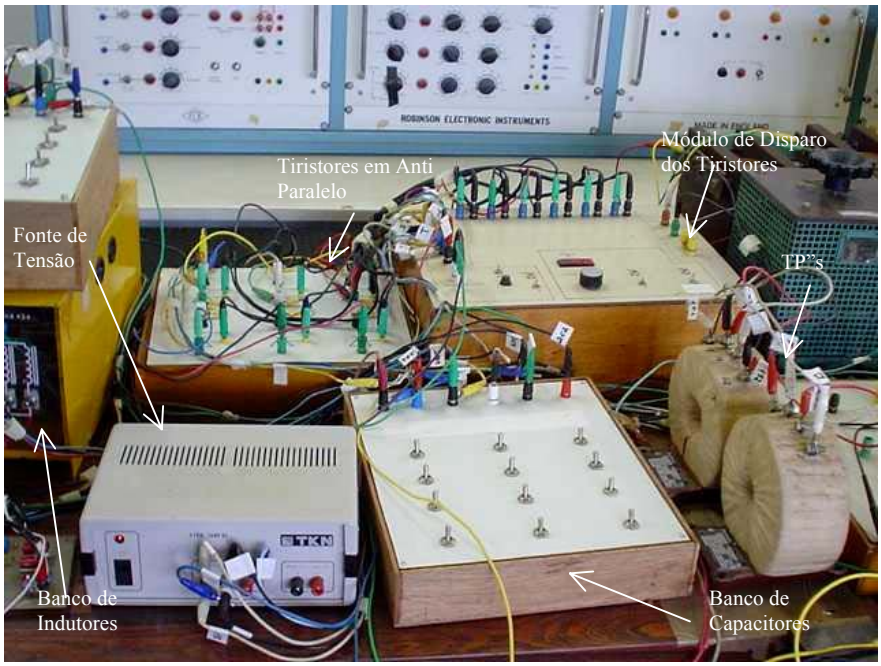

Figura A.13 – Circuito de Controle de Tensão da Carga do Gerador de Indução

Apêndice B – Instalação do Software Gerenciador da Placa de Aquisição PCL 711

Como foi informado no capítulo 4, o software gerenciador da placa de aquisição PCL-711 é executado em conjunto com o software CHOPPV para contribuir com as funções de aquisição dos dados de entrada e geração dos sinais de saída através das conversões A/D e D/A. Nesta seção são descritos os passos necessários para instalação do software gerenciador da placa de aquisição PCL 711.

A seguir, os passos necessários para instalação do software gerenciador da placa de aquisição PCL 711:

1)Copiar os arquivos dos discos de instalação da placa de aquisição PC-Card 711 para o drive C:\ do computador

(SUGESTAO: C:\TC\PROJ\ )

2) Copiar os arquivos de biblioteca TIMER.H e TIME.H, que estão nos discos de instalação da placa de aquisição, para o diretório

C:\TC\INCLUDE E C:\TC\LIB

3)Copiar o arquivo 711 CS LIB para o diretório

C:\TC\LIB

4) Antes de executar os arquivos (*.EXE)

Deve-se executar o arquivo PCL-711.EXE

Obs.: O programa CHOPPV não é executado sem que a placa de aquisição de dados esteja devidamente instalada.

Apêndice C - Procedimento para Criação de Projeto em Linguagem C++

A criação de um projeto em linguagem C++ está ligado a idéia de mais de um programa sendo executados simultaneamente para atendimento a uma finalidade específica. Neste trabalho foi necessária a criação de um projeto porque foi preciso associar o programa para Estimação e Controle de Velocidade de Motor de Corrente Contínua com o programa gerenciador da placa de aquisição PCL711 cujo nome é CS711.LIB cuja função é aquisição de dados de entrada, pelas conversões A/D e D/A. A seguir serão apresentados os passos para criação de um projeto ".PRJ" em linguagem C++, assim como foi feito para criação do projeto CHOPPV, como segue:

1) Conforme ilustrado na figura C1, selecionar a opção OPEN PROJECT no menu PROJECT:

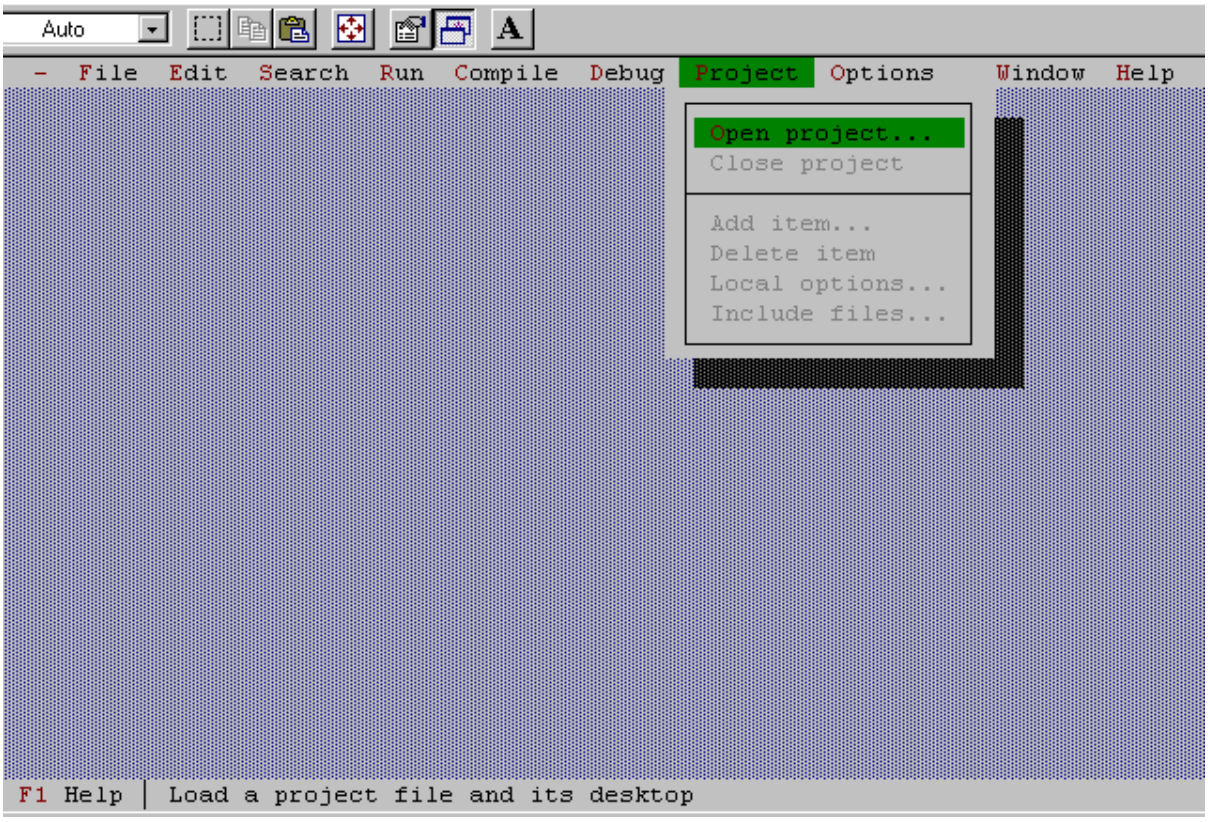

Figura C1: Primeira Tela para Criação de Projeto

Para selecionar a opção Open project deve-se arrastar o mouse sobre o item de menu Project e em seguida clicar uma vez com o botão esquerdo do mouse e então arrastar o mouse para o comando Open project.

2) Nomear o projeto no campo Open Project File e selecionar OK clicando uma vez com o botão esquerdo do mouse sobre a palavra OK:

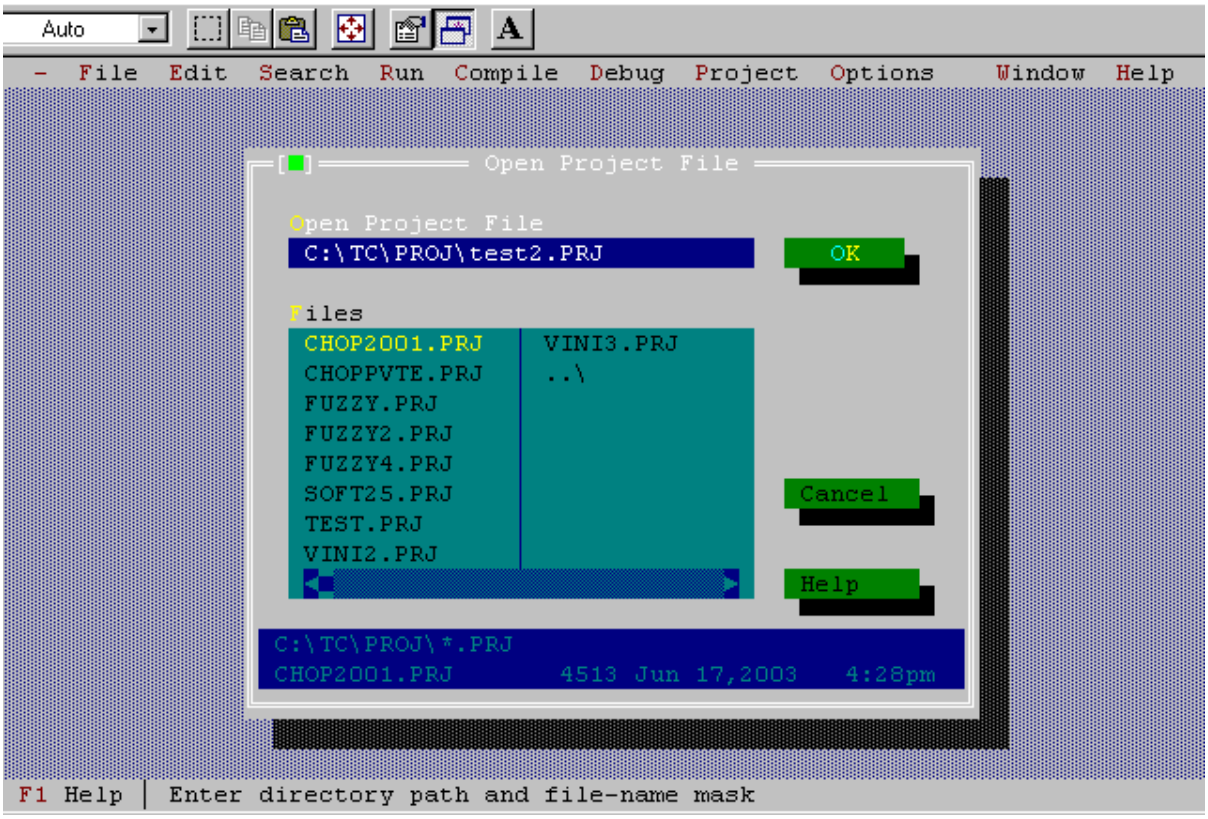

Figura C2: Segunda Tela para Criação de Projeto

O projeto deve ser nomeado com a extensão “.PRJ” como ilustrado na figura C2.

3) Selecionar a opção ADD ITEM arrastando o mouse sobre este comando e clicar uma vez com o botão esquerdo do mouse., como ilustrado na figura C3,:

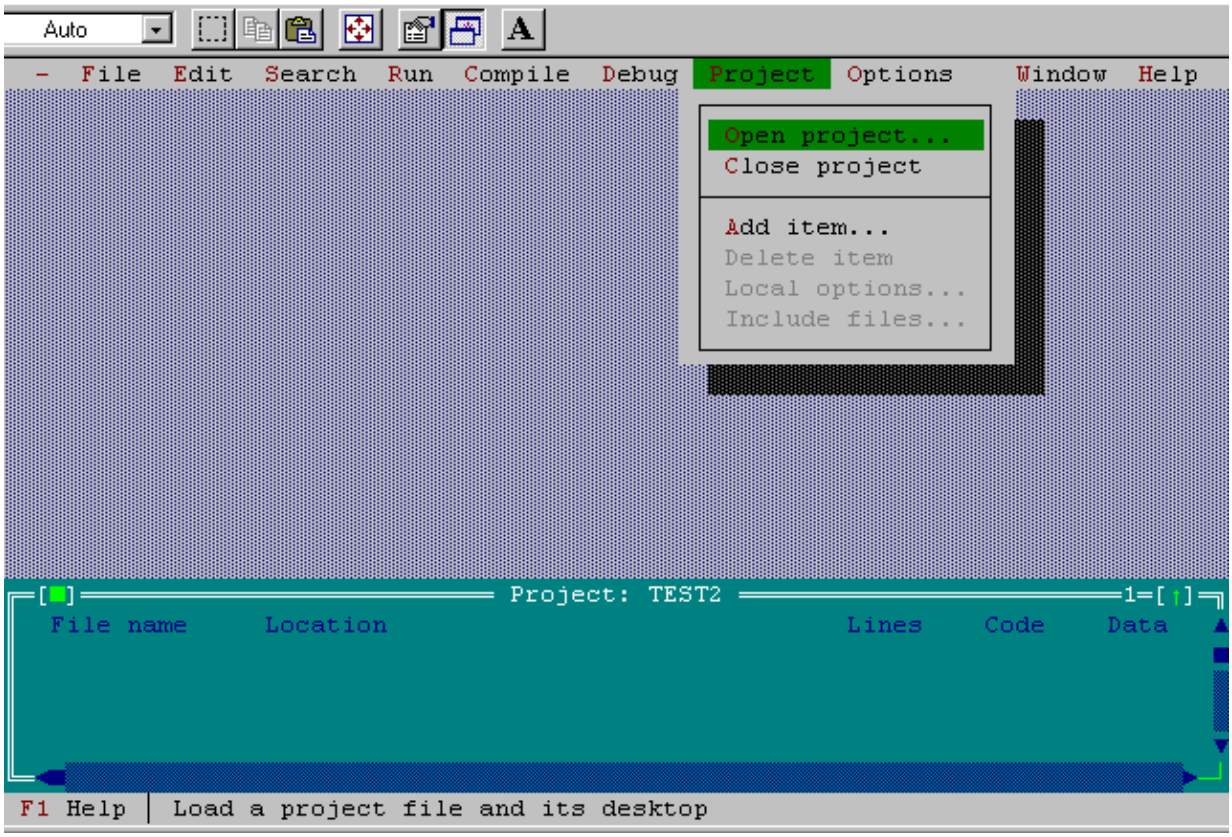

Figura C3: Terceira Tela para Criação de Projeto

Nesta etapa inicia a integração do programa CHOPPV.CPP com o programa CS711.LIB para compor um projeto.

4) Procurar o arquivo 711CS.LIB e selecionar a opção ADD clicando uma vez com o botão esquerdo do mouse sobre este comando:

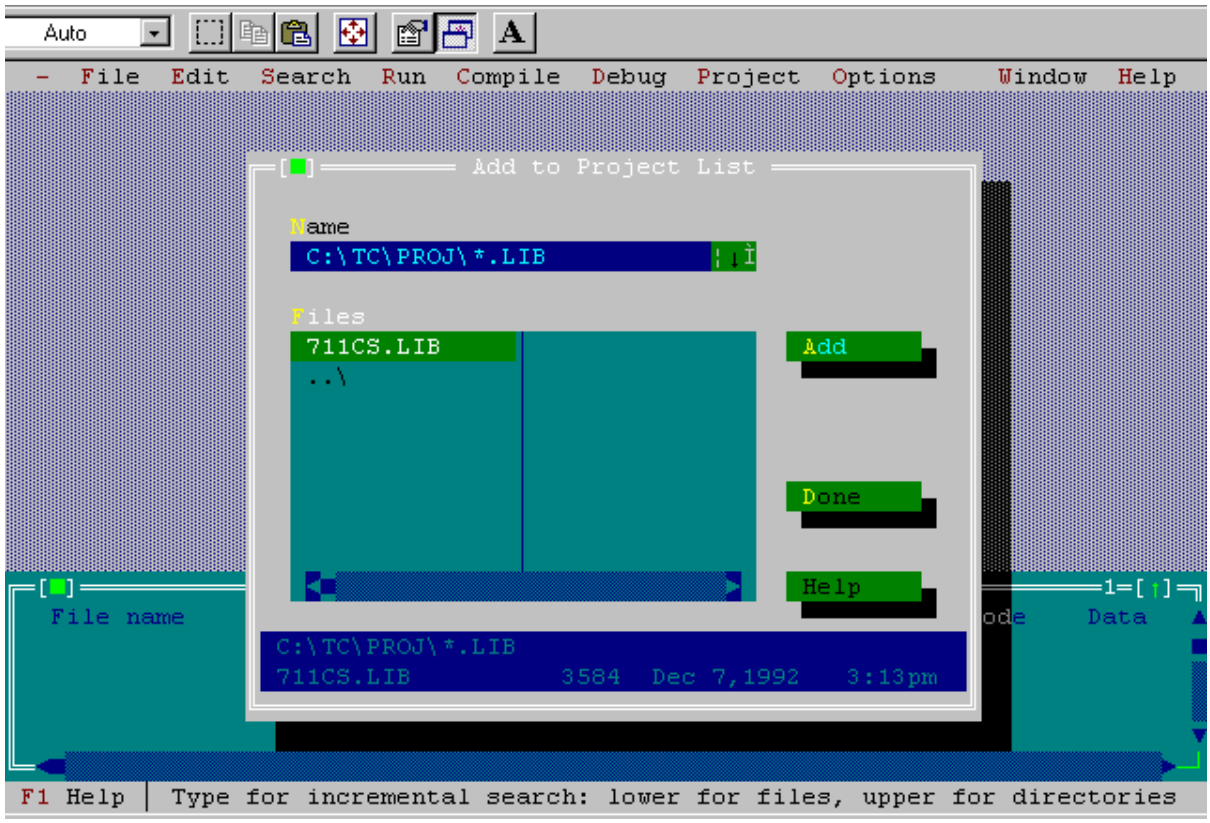

Figura C4: Quarta Tela para Criação de Projeto

Ao término do passo 3, abre-se a janela *Add to Project List* ilustrada na figura C4. Nesta janela deve-se procurar o arquivo 711CS.LIB, e clicar uma vez com o botão esquerdo do mouse para selecioná-lo.

5) Procurar o arquivo código fonte do programa *.CPP, que neste caso é o programa CHOPPV1, e selecionar a opção ADD clicando uma vez com o botão esquerdo do mouse sobre esta opção, como ilustrado na figura C5:

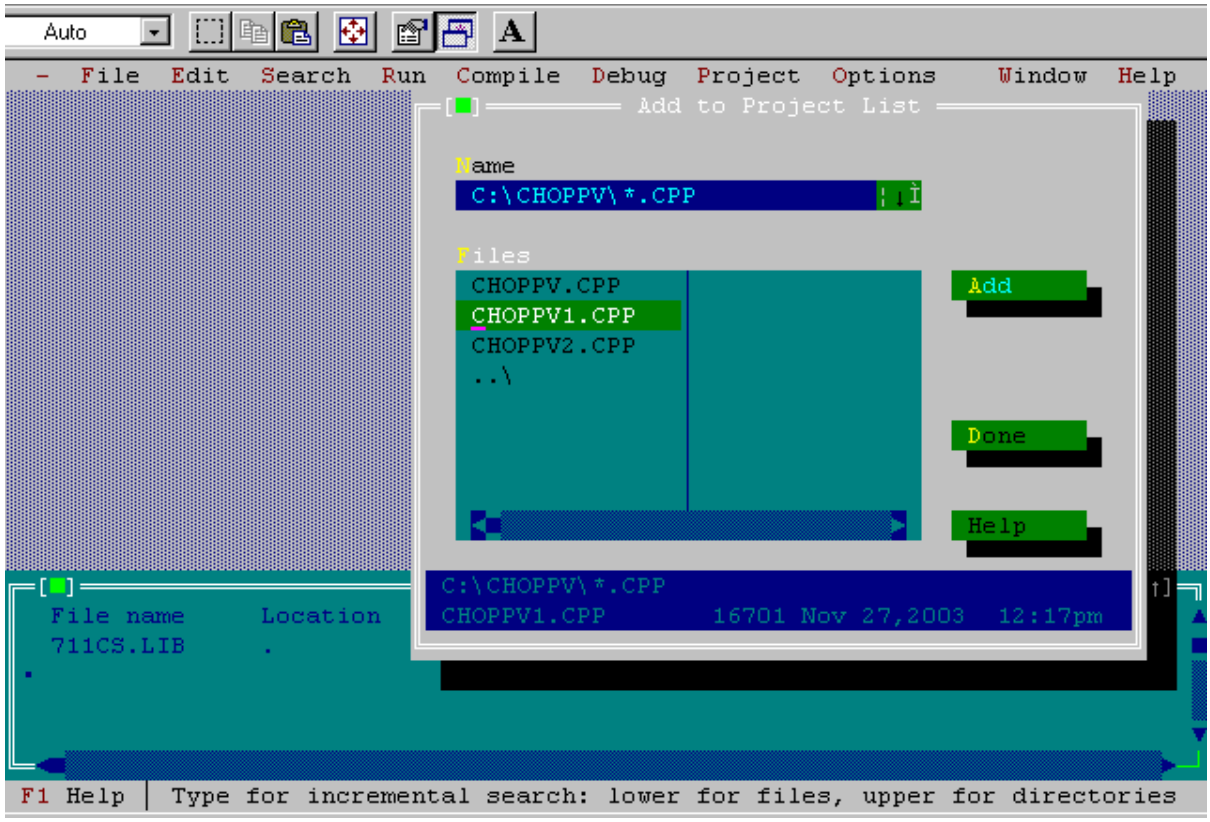

Figura C5: Quinta Tela para Criação de Projeto

Ao selecionar o programa CHOPPV1 e adicioná-lo a lista do projeto termina-se a seleção de programa que fará parte do projeto.

6)Selecionar a opção DONE clicando uma vez com o botão esquerdo do mouse sobre esta opção, conforme ilustrado na figura C6:

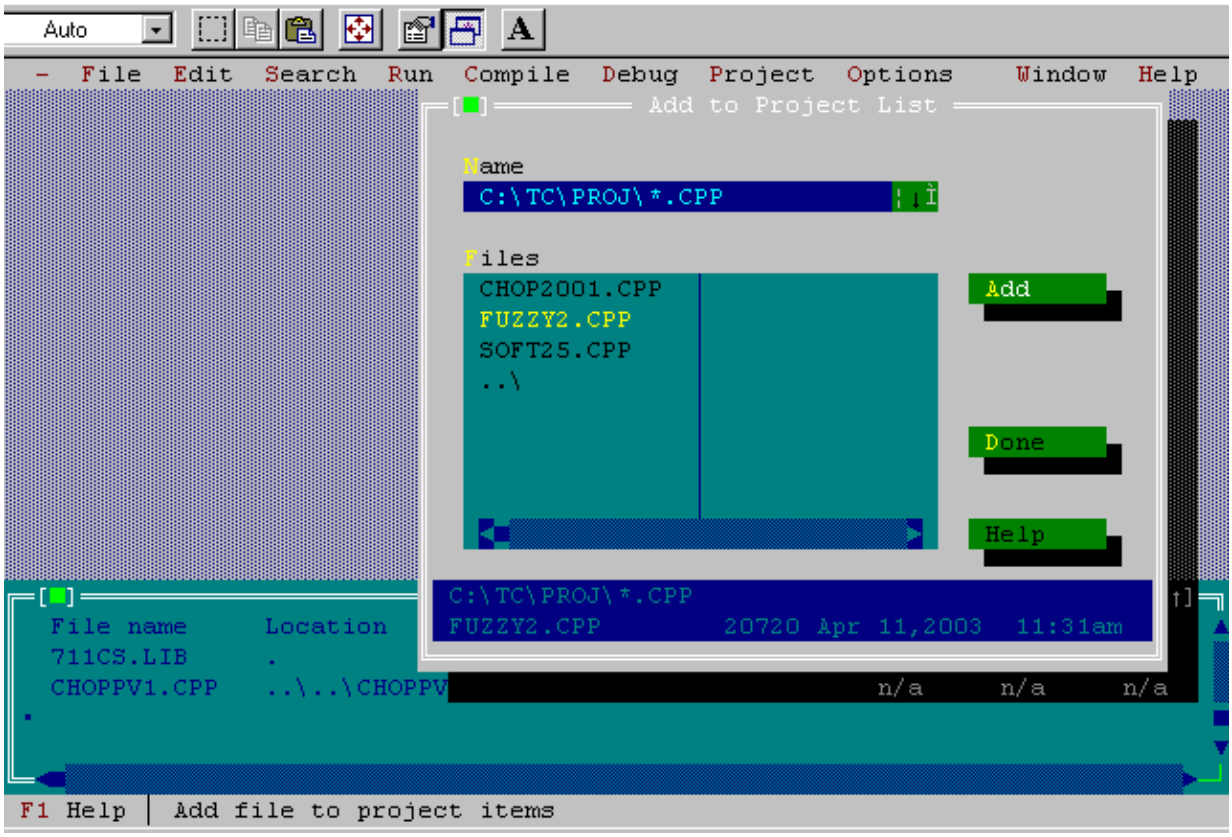

Figura C6: Sexta Tela para Criação de Projeto

Nesta etapa ilustrada na figura C6 finaliza-se a integração dos programas CHOPPV1.CPP e 711CS.LIB para formar o projeto.

7) Selecionar a opção LINKER – LIBRARIES no menu OPTIONS, clicando uma vez com o botão esquerdo do mouse sobre esta opção, conforme ilustrado na figura C7:

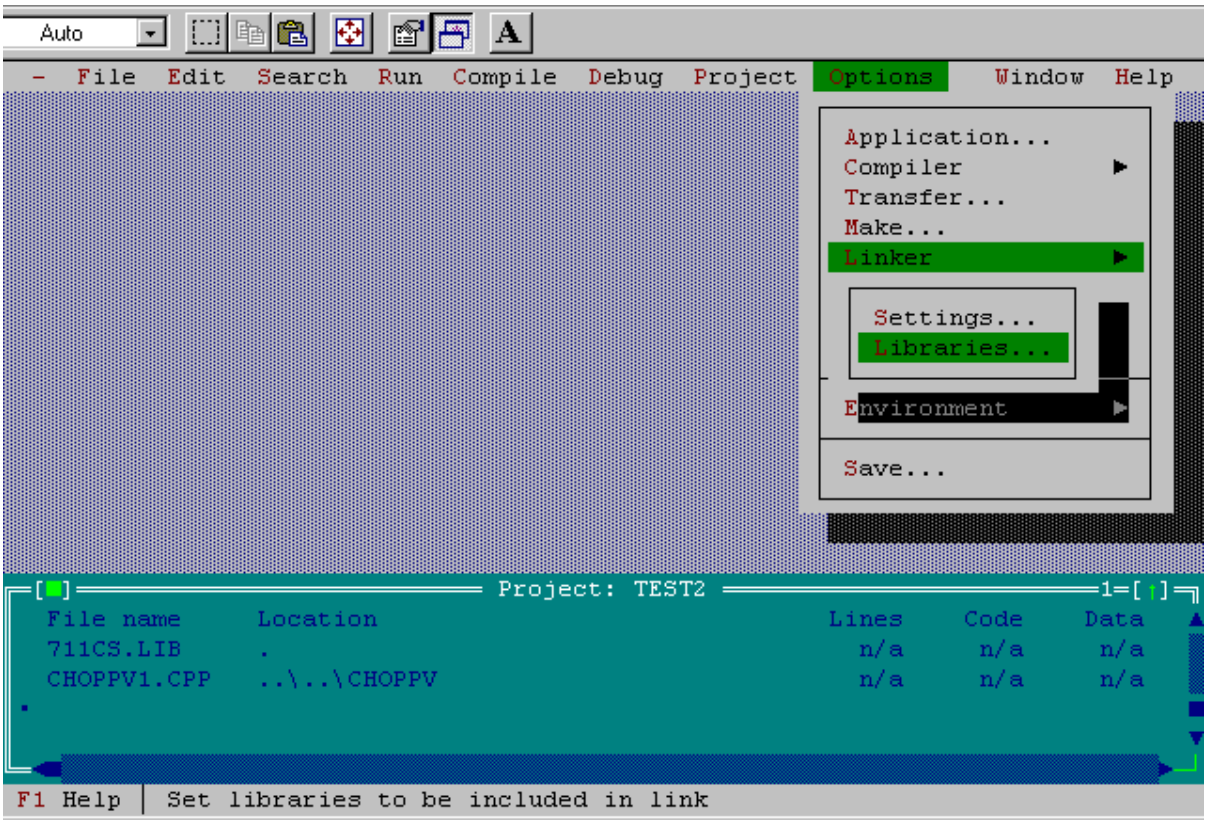

Figura C7: Sétima Tela para Criação de Projeto

8) Marcar as opções CONTAINER CLASS, GRAPHICS LIBRARY E STANDARD RUN TIME, conforme ilustrado na figura C8 a seguir:

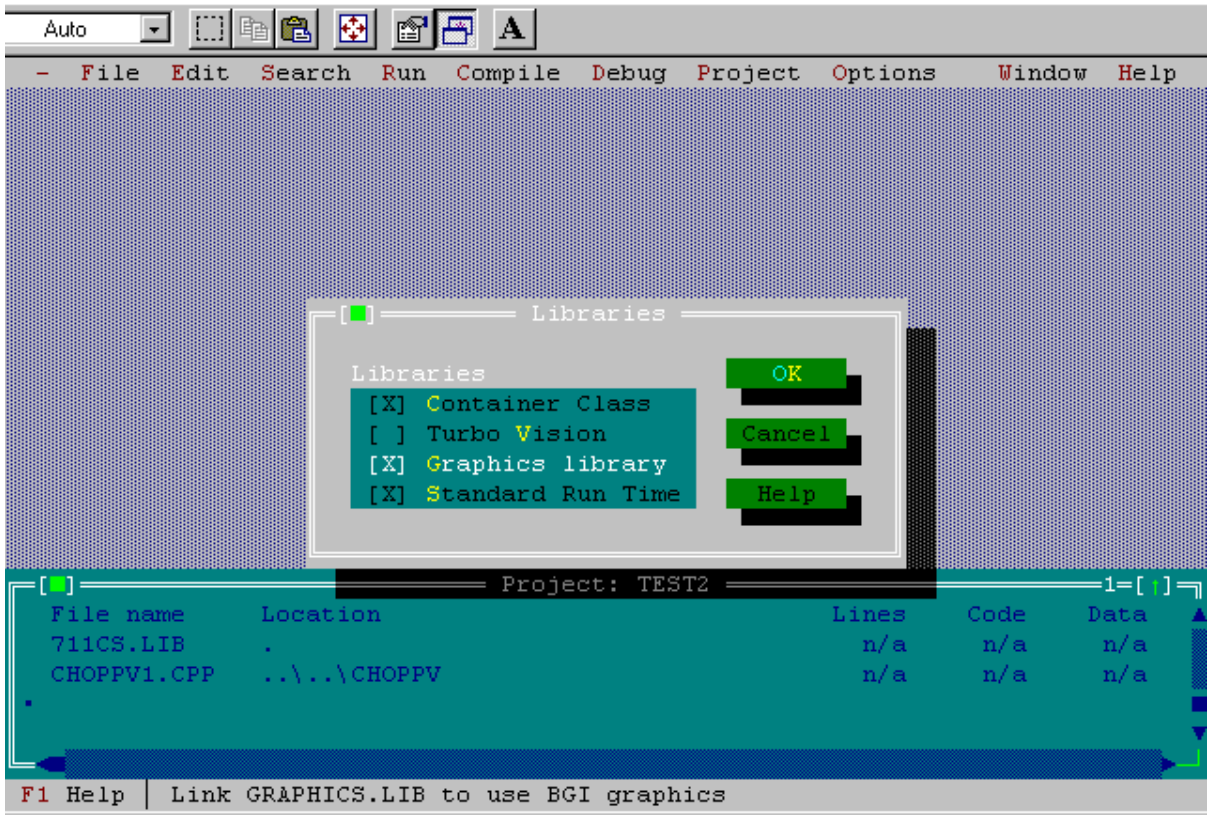

Figura C8: Oitava Tela para Criação de Projeto

A figura C8 ilustra as bibliotecas escolhidas para o projeto definido no passo 6.

9) Conforme ilustrado na figura C9, executar o projeto selecionando a opção RUN ou as teclas CTRL e F9:

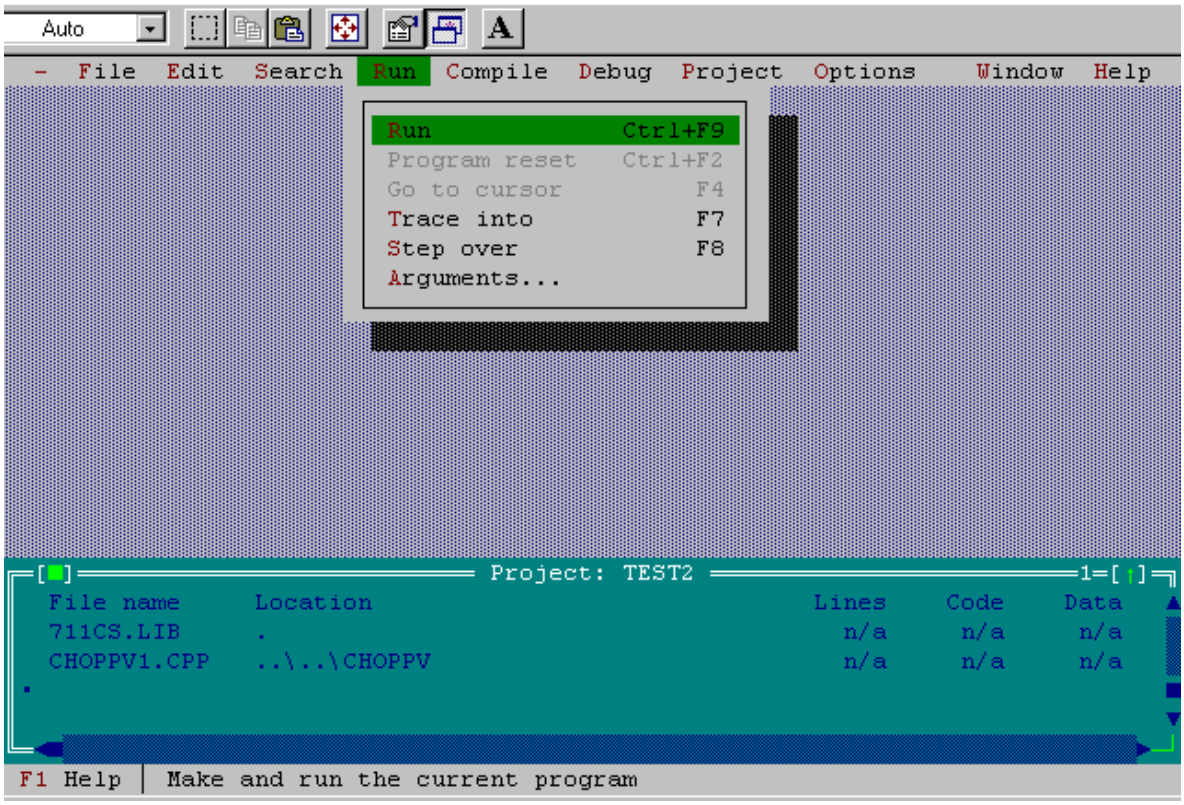

Figura C9: Nona Tela para Criação de Projeto

A figura C9 ilustra o procedimento para executar o projeto definido e configurado nos passos anteriores.

10) Reapresentação da Tela de Execução do software CHOPPV ilustrada na figura 4.8 a seguir:

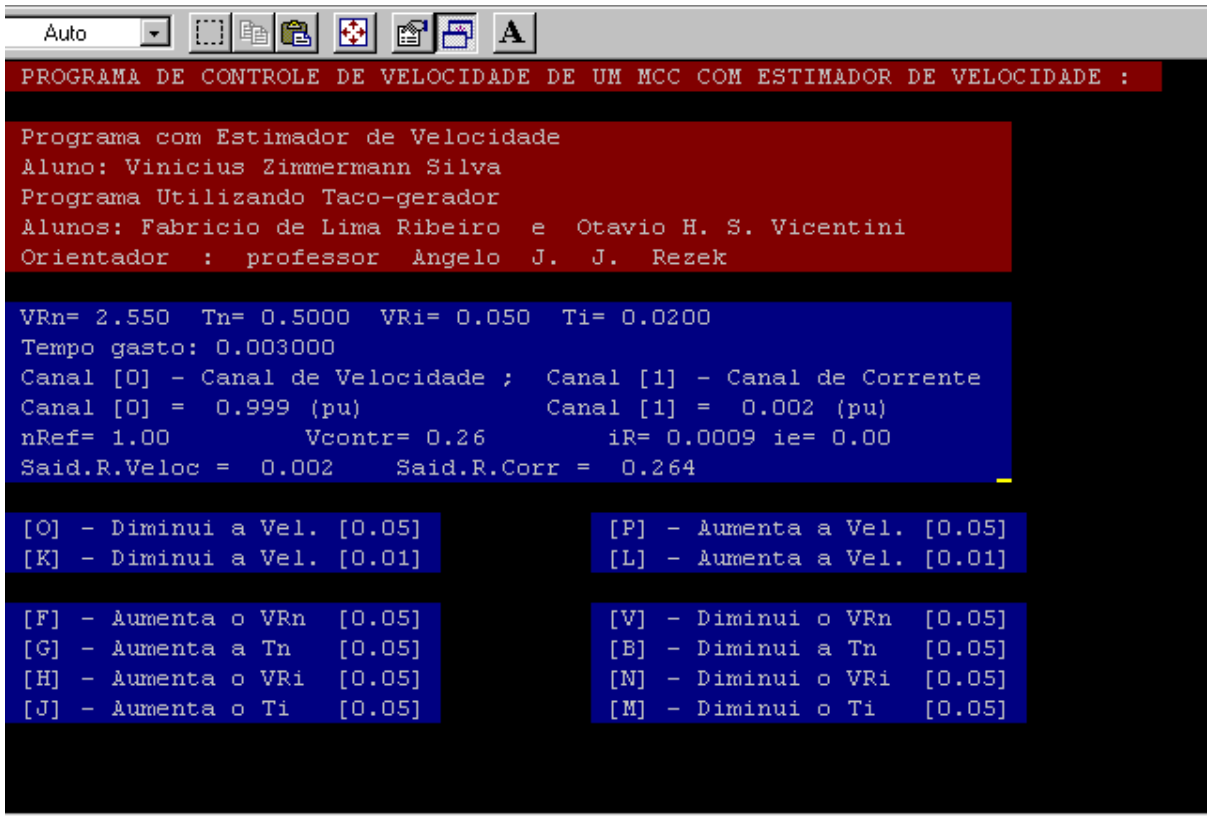

Figura 4.8: Tela de Execução do Software CHOPPV

Todos os parâmetros e informações contidas nesta tela estão detalhadamente descritos no capítulo 4.

11) Antes de abrir um novo projeto verificar se o projeto anterior foi fechado. Para fechar um projeto deve-se selecionar a opção CLOSE Project conforme ilustrado na figura C10:

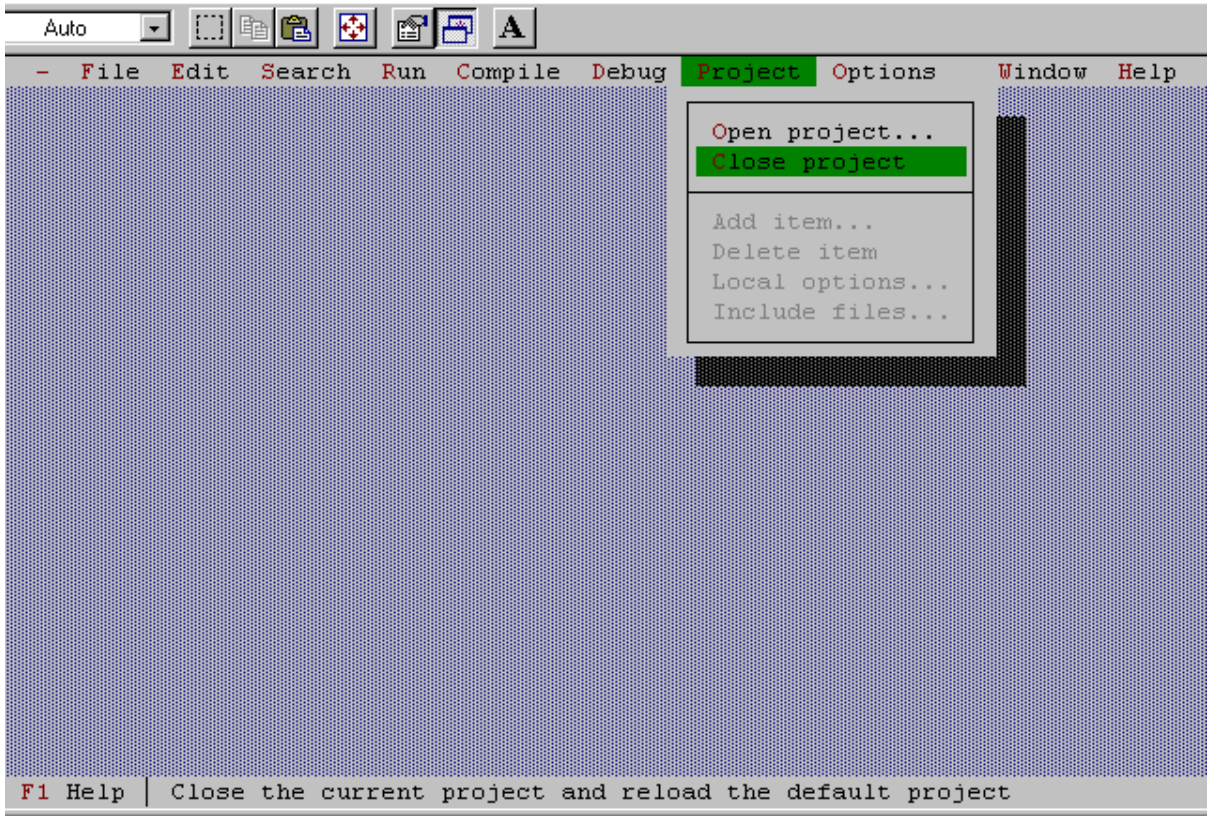

Figura C10: Instrução para Fechamento de Projeto

Caso um projeto tenha sido aberto e não fechado, não se conseguirá abrir um novo projeto até que este seja fechado. Portanto esta etapa é apenas uma ação que deve ser seguida antes da abertura de um novo projeto.

Anexo A – Memorial de Cálculo para Obtenção das Equações Recursivas

O regulador PI (proporciona - integral) é o mais apropriado para esse tipo de controle, sendo muito utilizado em aplicações industriais onde se deseja um sistema um comportamento preestabelecido, atendendo a algumas especificações, como: overshoot, tempo de acomodação, erro em regime permanente, etc. Adota-se um critério de otimização, efetuando-se assim os ajustes do ganho e constante de tempo de ação integral do regulador, necessários para o controle e regulação de velocidade e corrente em malha fechada [17].

Os reguladores PI foram dimensionados pelo método de otimização em função da simetria e têm a seguinte forma:

$$G_C(s) = V_R\left(1 + \frac{1}{T_I s}\right) \qquad (1)$$

Sendo:

G_C - Função de transferência do regulador

V_R - Ganho proporcional do regulador

T_I - Constante de tempo de integração

O filtro de alisamento de sinal é dado por:

$$G_F(s) = \frac{1}{1 + T_F s} \qquad (2)$$

Sendo: G_F - Função de transferência do filtro

T_F - Constante de tempo do filtro

Os compensadores e filtros digitais são introduzidos computacionalmente no sistema através de equações recursivas aplicando o método de aproximação por integração trapezoidal, conforme demonstrado na Figura AA1:

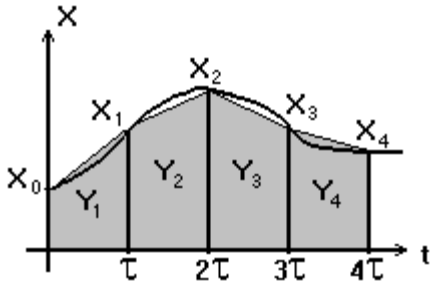

Figura AA1 - Método de aproximação por integração trapezoidal.

Pelo método da integração trapezoidal, temos:

$$Y_1 = Y_0 + \frac{\tau(X_0 + X_1)}{2}$$
$$Y_2 = Y_1 + \frac{\tau(X_1 + X_2)}{2}$$
$$Y_n = Y_{n-1} + \frac{\tau(X_{n-1} + X_n)}{2} \qquad (3)$$

Sendo:

τ- Tempo de amostragem

Aplicando transformada Z, chegamos a:

$$\frac{Y(z)}{X(z)} = \frac{\tau(z+1)}{2(z-1)} \qquad (4)$$

Podemos então fazer a seguinte aproximação:

$$\Im\left[\frac{1}{s}\right] = \frac{\tau(z+1)}{2(z-1)} \qquad (5)$$

Então, os reguladores PI podem ter a seguinte representação discreta no tempo:

$$\frac{Y(z)}{X(z)} = \frac{V_R}{2T_I}\left[\frac{(\tau + 2T_I)z + (\tau - 2T_I)}{(z-1)}\right] \qquad (6)$$

E, na forma de equação recursiva:

$$Y(k) = Y(k-1) + \left(\frac{V_R\tau}{2T_I} + V_R\right)X(k) + \\ + \left(\frac{V_R\tau}{2T_I} - V_R\right)X(k-1) \qquad (7)$$

Para os filtros digitais:

$$\frac{Y(z)}{X(z)} = \frac{\tau(z+1)}{(\tau + 2T_F)z + (\tau - 2T_F)} \qquad (8)$$

E, na forma de equação recursiva:

ANEXO A – MEMORIAL DE CÁLCULO PARA OBTENÇÃO DAS EQUAÇÕES RECURSIVAS

$$Y(k) = \frac{\tau}{(2T_F + \tau)}\left[X(k) + X(k-1)\right] + \\ + \frac{(2\mathrm{T_F} - \tau)}{(\tau + 2T_F)}Y(k-1) \qquad (9)$$

A implementação prática dos reguladores PI e filtros digitais foi efetuada com eficiência, com regulação de velocidade e limitação de corrente, independente da carga do motor..

Anexo B – Código Fonte para Acionamento e Controle de Velocidade de Motor de Corrente Contínua Utilizando Taco Gerador

A seguir será apresentado o código fonte do programa em linguagem C++ que fará o controle da velocidade do MCC com a utilização do taco gerador.

Para um melhor entendimento sobre a funcionalidade e operação do software, é apresentado na figura AB1 o correspondente fluxograma.

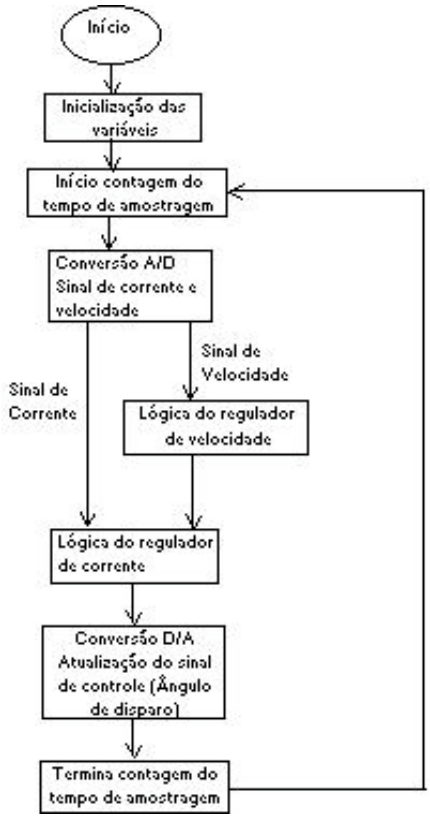

Figura AB1 – Fluxograma do Software com Taco Gerador

ANEXO B – CÓDIGO FONTE EM LINGUAGEM C++ PARA CONTROLE DE VELOCIDADE DE MCC UTILIZANDO TACO GERADOR

```
/*

**********************************************************************

    *  Programa    : CHOPPV - COM TACO                          *

    *  Descricao   : Controle digital para acionamento e controle de    *

    *                velocidade de uma m quina de corrente continua     *

    *                utilizando cartao PCL-711B                  *

    *  Versao      : CHOPPV - VERSAO 0                                   *

    *  Data        : 21/11/2003                          *        *

**********************************************************************

*/


/* Inclusao de Diretivas */


#include <stdio.h>

#include <conio.h>          /* Aceita diretivas, incluindo c¢digos */

#include <stdlib.h>              /* de fontes de outros */

#include <dos.h>               /* programas ou diret¢rios. */

#include <math.h>    /* Funcao Cosseno necessita desta biblioteca */

#include <timer.h>
```

ANEXO B – CÓDIGO FONTE EM LINGUAGEM C++ PARA CONTROLE DE VELOCIDADE DE MCC UTILIZANDO TACO GERADOR

```
/* Declara‡Æo de Vari veis Globais */


extern "C" pcl711(int, unsigned int *);   /* Inclui fun‡Æo "pcl711" definida
     inteiro, sem sinal       em um m¢dulo separado utilizando
                                        linguagem "C" */
unsigned int param[60];                          /* Defini‡Æo de um vetor de dados -
                                        array - que formam a tabela de
                                        par‚metros inteiros e s/ sinal */
unsigned int datain[200], dataout[200];     /* Buffer de 10 dados inteiros +
                                        para conversÆo */
unsigned int far * datin, * datout;       /* Endere‡o do buffer de dados acima
                                        - pointer - tipo inteiro e longo:
                                        2 palavras c/ range de 1Mbyte */
int tecla,i;                             /* Vari veis de leitura do teclado
                                        e numero de canais */
/* Vari veis pontos flutuantes do controle */
float d;
long float eff=218.0, Va, En=182.05, Ra=3.5;
float nRef=1.00, nReal=0.0, ne=nRef;
float ne_1=0.1, nReal_1=0.0;
float iRef=1.2, iR=1.2, ie=-1.2, iReal=0.0, Ilim=1.2, 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=2.0, Tgs2=0.022, Ti=0.020;

float VRn=2.55, VRi=0.050, Vcc=0.1, Vcc_1=0.1, Vcontr=0.1;

float a1, a2, b1, b2, b3, b4, T=0.003;

float DataBuf[3];

float e,  v , dd ;


/* Declara‡Æo de vari veis void - significa que nÆo retorna um valor */

void conv_ad(void);

void conv_da(void);

void control(void);

void control1(void);

void control2(void);

void teclado(void);


/* ConversÆo AD - Tabela de par*f*metros */


void conv_ad()

{

      unsigned int i;

/* Pointer - Espa‡o de mem¢ria - Vari vel que cont,m um endere‡o que,

  normalmente , o endere‡o de outra vari vel."                  */

      datin = datain;                /* Atribui ao pointer datin o valor
```

```
                                    equivalente … vari vel datain   */
    param[0]  = 0;                  /* N£mero do cartÆo             */
    param[1]  = 0x220;              /* Endere‡o de Base I/O         */
    /* Frequ^ncia de amostragem = Frequ^ncia de base do cartÆo/(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 fun‡äes D/I          */
    /* Offset do Buffer , o endere‡o de mem¢ria (buffer) onde os dados
    serÆo 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;                  /* Endere‡o do Buffer B (nÆo usado)*/
    param[13] = 0;                  /* Segmento- NÆo usado, setar em 0 */
    /* A conversÆo 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;                  /* N£mero de conversäes A/D     */
    param[15] = 0;                  /* Canal de in¡cio da conversÆo A/D*/
    param[16] = 1;                  /* Canal de parada da conversÆo A/D*/
    param[17] = 0;                  /* Ganho dos canais, 0 : +/- 5V  */
```

```
        /* Indica‡Æo de falha na conversÆo A/D */


  pcl711(3, param);       /* Func. 3 : Inicializa‡Æo do Hardware      */
  if (param[45] != 0) {   /* Se par*f*metro 45 diferente de 0, fazer:   */
clrscr();          /* Limpar a tela                          */
printf("\n FALHA NA INICIALIZA€ÇO DO DRIVER !"); /* Imprimir   */
getch();           /* Mostrar a tela de sa¡da                */
exit(1);           /* Fecha o loop e sai com status 1 - Erro   */
          }


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


  pcl711(5, param);       /* Func 5 : Verifica‡Æo n£mero conversäes A/D*/
```

```
    if (param[45] != 0) {
  clrscr();
printf("\n FALHA NO SOFTWARE DE TRANSFERÒNCIA DE DADOS A/D !");
  getch();
  exit(1);
          }
                /* Conversäes A/D */
    for (i = 0; i < param[14]; i++)   /* Dados amostrados - canais 0 e 1 */
{
  DataBuf[i] = datain[i]  & 0xFFF;
       /* Coleta de dados para o buffer no endereco 0xFFF
(os trˆs primeiros d¡gitos hexadecimais podem ser zerados pois o
restante , suficiente p/ suportar 4096 d¡gitos bin rios ) */
  DataBuf[i] =((5.0 - (-5)) * DataBuf[i] / 4096) + (-5);
  /* ConversÆo para que o sinal de tensÆo seja dispon¡vel para
    aplica‡Æo nas equa‡äes 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)       : Inicio da escala do A/D "-5" V
  */
 }
```

```
/* Leitura da tensÆo de realimenta‡Æo para a malha de velocidade
   e de corrente /*
    /* Sinal de Velocidade Aquisitada do Taco Gerador */
     nReal=(DataBuf[0]/2.700);

 // Sinal de Corrente Obtida do Transdutor de Corrente, Sensor Hall;
iReal=(DataBuf[1]/1.493);
}

                    /* Controle */
/* Equa‡äes Recursivas para efetuar fun‡äes 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()

{

ne=nRef-nReal;

iRef=(b1*ne)+(b2*ne_1)+iRef_1; /* nreal= Realimenta‡Æo de velocidade*/


if(iRef>1.2) ne=1.0*ne;

if(iRef<-1.2) ne=1.0*ne;

iRef=(b1*ne)+(b2*ne_1)+iRef_1;

iRef_1=iRef;

ne_1=ne;


                    /* Filtro da Corrente de Referˆncia */

iRefer=iRef;

iR=a1*(iRefer+iRefer_1)+a2*iR_1; /*  Sendo:                    */

 /* iR= Corrente refer. ap¢s filtro */
```

```
    /* iRef= Corrente referˆncia       */
if(iR>=Ilim) iR=Ilim;
if(iR<=-Ilim) iR=-Ilim; /* Limita‡Æo da corrente de referˆncia*/
iR_1=iR;
iRefer_1=iRefer;
}
                    /* Regulador de Corrente */
void control2()
{                           /*  Sendo:                          */
ie=iReal-iR;
Vcc=(b3*ie)+(b4*ie_1)+Vcc_1; /* ireal= realimenta‡Æo de corrente  */

        /* iR= Corrente ref. ap¢s filtro     */
        /* ie= Erro corrente- Ent. regulador */
if(Vcc>0.9) ie=1.0*ie;

if(Vcc<0.1) ie=1.0*ie;

Vcc=(b3*ie)+(b4*ie_1)+Vcc_1;
Vcc_1=Vcc;
ie_1=ie;
if(Vcc>0.9) Vcontr=0.9;
```

```
if(Vcc<.10) Vcontr=.10;    /* Limita‡Æo da tensÆo de controle   */

if(Vcc<=0.90 && Vcc>=.10) Vcontr=Vcc;



}

          /* Altera‡Æo nos par*f*metros do sistema */

void teclado()

 {

    //     if (kbhit()) tecla=getch();  /* Se for pressionada alguma tecla, */

                                  /*    abrir a tela de saida */

   /* O acionamento destas teclas de subrotina, permitem ajuste on-line de

     par*f*metros do sistema, com o ajuste sendo mostrado na tela de sa¡da */


  /* Teclas "s" e "d" atuando na valor da velocidade de referˆncia */

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 referˆncia */

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 referˆncia */

if (tecla==112 && nRef<2.0) nRef=nRef+0.20;  /* Ajuste limitado ao */
```

```
if (tecla==111 && nRef>-2.0) nRef=nRef-0.20;  /* 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<5.000) 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;

}


        /*  ConversÆo D/A - Tabela de Par*f*metros */


void conv_da()
```

```
{
    datout=dataout;          /* Atribui ao pointer datout o valor
                             equivalente … vari vel dataout       */
    param[0]=0;              /* N£mero do cartÆo                  */
    param[1]=0x220;          /* Endere‡o de base I/O              */
/* Offset do Buffer , o endere‡o de mem¢ria (buffer) onde os dados
   serÆo guardados. Segmento , o comprimento do buffer de dados */
    param[20] = FP_OFF(datout); /* Offset do buffer A dados  sa¡da D/A  */
    param[21] = FP_SEG(datout); /* Segmento do buffer A dados sa¡da D/A */
    param[22] = 0;           /* Endere‡o do Buffer B sa¡da(nÆo usado) */
    param[23] = 0;           /* Segmento sa¡da- NÆo usado, setar em 0 */
    param[24] = 1;           /* N£mero de conversäes D/A          */
    param[25] = 0;           /* Canal de in¡cio da conversÆo D/A      */
    param[26] = 0;           /* Canal de parada da conversÆo D/A      */

                /* Indica‡Æo de falha na conversÆo D/A */
    pcl711(3, param);       /* Func 3 : Inicializa‡Æo do hardware      */
    if (param[45] != 0)  {  /* Se par*f*metro 45 diferente de 0, fazer:  */
  clrscr();          /* Limpar a tela                    */
  printf("\n FALHA NA INICIALIZA€ÇO DO DRIVER !"); /* Imprimir   */
  getch();           /* Mostrar a tela de saida          */
  exit(1);           /* Fecha o loop e sai com status 1 - Erro   */
```

```
            }
      pcl711(12, param);      /* Func 12: Inicializa‡Æo do conversor D/A  */
      if (param[45] != 0) {
   clrscr();
   printf("\n FALHA NA INICIALIZA€ÇO DO D/A !");
   getch();
   exit(1);
            }
      pcl711(13, param);     /* Func 13: Verifica‡Æo n£mero conversäes D/A*/
      if (param[45] != 0) {
   clrscr();
printf("\n FALHA NO SOFTWARE DE TRANSFERÒNCIA DE DADOS D/A !");
   getch();
   exit(1);
            }

  /* ConversÆo para que o sinal de tensÆo de sa¡da das equa‡äes
     recursivas de controle em pu ocupe um espaco de endere‡o do
     buffer de dados de sa¡da.                          */
      dataout[0]=(4095*Vcontr);
}
```

ANEXO B – CÓDIGO FONTE EM LINGUAGEM C++ PARA CONTROLE DE VELOCIDADE DE MCC UTILIZANDO TACO GERADOR

```
                    /* Programa principal */


void main(void)   /* Garante que as vari veis globais nÆo retornam valores */

{                 /* Declara‡Æo de vari veis */

int xx=0;      /* Inicializa o n£mero de intera‡äes em 0 */

Timer t;          /* Contador de tempo "t" */


clrscr();


textbackground(4);            /* Define Fundo Vermelho (4) */

gotoxy(1,1);

cprintf(" PROGRAMA DE CONTROLE DE VELOCIDADE DE UM MCC COM
TACO GERADOR       :  ");

gotoxy(1,3);

cprintf(" Programa com Estimador de Velocidade                          ");

gotoxy(1,4);

cprintf(" Aluno: Vinicius Zimmermann Silva                              ");

gotoxy(1,5);

cprintf(" Programa Utilizando Taco gerador                              ");

gotoxy(1,6);

cprintf(" Alunos: Fabricio de Lima Ribeiro  e  Otavio H. S. Vicentini       ");

gotoxy(1,7);
```

```
cprintf(" Orientador  :  professor  Angelo  J.  J.  Rezek                ");


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


gotoxy(1,16);  cprintf(" [O] - Diminui a Vel. [0.05] ");

gotoxy(40,16); cprintf(" [P] - Aumenta a Vel. [0.05] ");

gotoxy(1,17);  cprintf(" [K] - Diminui a Vel. [0.01] ");

gotoxy(40,17); cprintf(" [L] - Aumenta a Vel. [0.01] ");

gotoxy(1,19);  cprintf(" [F] - Aumenta o VRn  [0.05] ");

gotoxy(40,19); cprintf(" [V] - Diminui o VRn  [0.05] ");

gotoxy(1,20);  cprintf(" [G] - Aumenta a Tn   [0.05] ");

gotoxy(40,20); cprintf(" [B] - Diminui a Tn   [0.05] ");

gotoxy(1,21);  cprintf(" [H] - Aumenta o VRi  [0.05] ");

gotoxy(40,21); cprintf(" [N] - Diminui o VRi  [0.05] ");

gotoxy(1,22);  cprintf(" [J] - Aumenta o Ti   [0.05] ");

gotoxy(40,22); cprintf(" [M] - Diminui o Ti   [0.05] ");


do{


  tecla=0;

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

```
xx++;           /* Incrementa 1 no n£mero de intera‡äes */

t.reset();      /* Reseta o temporizador "t" */

t.start();      /* Inicia a temporiza‡Æo */


asm cli;

conv_ad();      /* Efetua a subrotina de conversÆo A/D */

asm sti;

                /* Condi‡Æo para efetuar as equa‡äes de controle -

                sinal de realimenta‡Æo maior que o ru¡do normal */


  control();

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


ne=nRef-nReal;


if(iR<Ilim && iR>-Ilim) control1();

if(iR==Ilim && ne<0.0) control1();

if(iR==-Ilim && ne>0.0) control1();

 ie=iReal-iR;


 if(Vcc<0.90 && Vcc>.10)control2();

 if(Vcc>=0.9 && ie<0.0 ) control2();
```

```
if(Vcc<=.10 && ie>0.0 ) control2();


asm cli;

conv_da();    /* Efetua a subrotina de conversÆo D/A */

asm sti;


if (xx>=100) {/* Imprime na tela resultados instant*neos cada 100 intera‡äes */

  xx=0;

  gotoxy(1,9);

  textbackground(1);

cprintf(" VRn= %1.3f  Tn= %1.4f  VRi= %1.3f  Ti= %1.4f                ", VRn, Tn,
VRi, Ti);

  gotoxy(1,10);

  cprintf(" Tempo gasto: %1.6f                                ", T);

  gotoxy(1,11);

  cprintf(" Canal [0] - Canal de Velocidade ;  Canal [1] - Canal de Corrente  ");

  gotoxy(1,12);

cprintf(" Canal [0] = % 1.3f (pu)          Canal [1] = % 1.3f (pu)       ", nReal, iReal);

  gotoxy(1,13);

 cprintf(" nRef= %1.2f          Vcontr= %1.2f         iR= %1.4f ie= %1.2f       ", nRef,
Vcontr, iR, ie);

  gotoxy(1,14);

  cprintf(" Said.R.Veloc =  %1.3f    Said.R.Corr =  %1.3f                ", iRef, Vcc);
```

```
    }

    t.stop();       /* Termina a temporiza‡Æo */

    T=t.time();     /* Consideramos o tempo de amostragem c/ valor inicial

    /*T=0.003 s. Se o tempo de execucao do programa "t" , maior que "T", necessario

    tentar reduzi-lo. Se "t" e menor fazemos T=t.*/

   T=.0030;

} while (tecla!=27);

textbackground(0);

clrscr();



}
```

Referências Bibliográficas

[1] REZEK, A.J.J.; RODRIGUES, M. S.; MIRANDA, V. A. M.; OLIVEIRA, V.A .; CASSULA, A.M.; COSTA JR., R.A; TORRES, A.Z. Design and Simulation of a Controlled DC Drive (in portuguese), In: II SIMEAR, ABINNE TEC 91, 1991, EPUSP São Paulo. Proceedings of 2nd International Seminar on Electrical Machines and Controlled Drives. São Paulo: USP, 1991, vol 3, p. 141-160.

[2] REZEK, A.J. J. Análise em Regime Permanente e Transitório de um Sistema de Conversão de Energia Elétrica AC/DC. Dissertação de mestrado em Engenharia Elétrica, EFEI, Itajubá, 1986.

[3] ASSIS, W.O. Projeto e Implementação do Acionamento Controlado para Máquina de Corrente Contínua Através de Chopper. Dissertação de Mestrado em Engenharia Elétrica. EFEI, Itajubá. 1997.

[4] LANDER, C.W. Eletrônica Industrial, Teoria e Aplicações. São Paulo: Editora McGraw-Hill Ltda, 1988.

[5] ALMEIDA, J.L.A. Eletrônica de Potência. São Paulo: Ed. Érica, 1986.

[6] OGATA, K. Engenharia de Controle Moderno. Rio de Janeiro: Editora Prentice / Hall do Brasil Ltda., 1982.

[7] SCHILDT, H.. C Completo e Total, 3.ed. São Paulo: Makron Books do Brasil Editora Ltda; 1996.

[8] KOSOW, I.L.. Máquinas Elétricas e Transformadores, 2.ed. Porto Alegre: Globo; 1979.

[9] ROSA, P.C., Implementação de Reguladores PID – Digitais em Sistemas com Microprocessadores. Itajubá:, 1989. 87p (Tese de Mestrado em Engenharia Elétrica. Escola Federal de Engenharia de Itajubá).

[10] ARANDA, G. E. I., Projeto e Implementação de um Acionamento Controlado Analógico e Digital para Máquina de Corrente Contínua, Utilizando CHOPPER de Quatro Quadrantes, Itajubá-MG:, 2000. (Tese de Mestrado em Engenharia Elétrica. Escola Federal de Engenharia de Itajubá).

[11] FRÖHR, F. & ORTTENBURGER, F., Introductión al Control Electrónico, Siemens, Marcombo S.A., Barcelona, España, 1986.

[12] BUXBAUM, A., SCHIERAU, K., STRAUGHEN, A. Design of Control Systems for DC Drives; Alemanha: Spring-Verlag Berlin Heidelberg, pp162-177, 1990.

[13] AHMED, A., Eletrônica de Potência; tradução Bazán Tecnologia e Lingüística, revisão técnica João Antonio Martino. São Paulo: Prentice Hall, 2000.

[14] PEREIRA, C.A.G., Otimização de Reguladores para Acionamento Controlado de Motores de Indução Alimentados por Intermédio de Inversor de Corrente Tiristorizado com

Comutação Natural, Itajubá, 2003. 115p. (Tese de Mestrado em Engenharia Elétrica. Universidade Federal de Itajubá).

[15] BRAGA, A.V., Modelagem, Ajuste e Implementação de um Sistema de Controle de Tensão para o Gerador de Indução, Itajubá, 2002. 79p. (Tese de Mestrado em Engenharia da Energia. Universidade Federal de Itajubá).

[16] REZEK, A.J.J., VICENTINI, O.H.S., CORTEZ, J.A., SILVA, V.F., ASSIS, W.O., MAGALHÃES, C., Análise Comparativa de Desempenho de Reguladores Digitais em Acionamento Controlado para Motor Série Corrente Contínua, Itajubá, [2001?]. 8p. (Artigo. Universidade Federal de Itajubá)

[17] ASSIS, W.O., REZEK, A.J.J., SILVA, L.E.B., Projeto e Implementação de Controle Digital para Acionamento de uma Máquina CC através de Chopper, [2000?]. 6p. (Artigo. Universidade Federal de Itajubá).

[18] REZEK, A.J.J., VICENTINI, O.H.S., CORTEZ, J.A., SILVA, V.F., Projeto de Reguladores para Acionamento de Máquinas Elétricas: Modelagem, Ajuste, Simulação e Verificação Experimental, Itajubá, [2002?]. 11p. (Artigo. Universidade Federal de Itajubá).

[19] ABNT – Associação Brasileira de Normas Técnicas, NBR 14724 – Informação e Documentação – Trabalhos Acadêmicos – Apresentação, [2002]. 6p.

[20] SEVERINO, A. J., Metodologia do Trabalho Científico, 21. ed. rev. e ampl., São Paulo: Cortez; 1941.