

RAIMUNDO NAZARENO CUNHA ALVES

+ CONTROLE DE VELOCIDADE DE MOTOR DC
POR MICROCOMPUTADOR

Dissertação apresentada à Coordenação dos Cursos de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Paraíba, em cumprimento parcial às exigências para obtenção do grau de Mestre em Engenharia Elétrica.

ÁREA DE CONCENTRAÇÃO : Processamento da Informação

ORIENTADORES : Prof. Gurdip Singh Deep e
Prof. José Homero Feitosa Cavalcanti

Campina Grande, abril de 1983



A474c Alves, Raimundo Nazareno Cunha.
Controle de velocidade de motor DC por microcomputador /
Alves, Raimundo Nazareno Cunha. - Campina Grande, 1983.
117 f.

Dissertação (Mestrado em Engenharia Elétrica) -
Universidade Federal da Paraíba, 1983.
"Orientação: Prof. Dr. Gurdip Singh Deep, Prof. Dr. José
Homero Feitosa Cavalcanti".
Referências.

1. Motor DC. 2. Microcomputador. 3. Sistema de Controle
de Velocidade. 4. Engenharia Elétrica - Dissertação. I.
Deep, Gurdip Singh. II. Cavalcanti, José Homero Feitosa.
III. Universidade Federal da Paraíba - Campina Grande (PB).
IV. Título

CDU 621.313.13:004.382.7(043)

CONTROLE DE VELOCIDADE DE MOTOR DC
POR MICROCOMPUTADOR

RAIMUNDO NAZARENO CUNHA ALVES

DISSERTAÇÃO APROVADA EM 19/04/83


GURDIP SINGH DEEP

Orientador


JOSÉ HOMERO FEITOSA CAVALCANTI

Co-Orientador


WANDERLEY LOPES DE SOUSA

Componente da Banca


EDISON ROBERTO CABRAL DA SILVA

Componente da Banca

CAMPINA GRANDE - ABRIL - 1983

AGRADECIMENTOS

Agradeço aos professores C.S. Deep e J.H.F. Cavalcanti pela orientação e apoio sempre presentes durante a elaboração deste trabalho.

Agradeço aos integrantes do Laboratório de Instrumentação Eletrônica, do Laboratório de Sistemas Digitais, da Oficina Mecânica, do Laboratório Fotográfico, do Laboratório de Eletrônica de Potência e demais amigos do CCT que em momento algum negaram sua ajuda para que este trabalho fôsse realizado.

À minha mãe e à lembrança de meu
pai

Aos meus irmãos, esposa e filhos.

RESUMO

Neste trabalho é apresentado um sistema de controle de velocidade de um motor DC baseado em microcomputador. O "hardware" e o "software" desenvolvidos permitem uma comunicação "on line" entre o operador e o sistema de controle. A flexibilidade do sistema fornece ao operador condições para dimensionar um esquema de controle dedicado a partir das especificações do motor e do tipo de aplicação. Através de recursos do "software" pode-se determinar algumas constantes do motor e do sistema de compensação e controle. Um método para aumentar o fator de utilização do microcomputador é abordado.

ABSTRACT

This work presents a microcomputer system to control the velocity of a dc motor. The hardware and software developed allow an on-line communication between the operator and the control system. The system flexibility gives the operator conditions to dimension a scheme of specific control with a knowledge the motor specifications and the type of application. By means of the software resources one can determine some constants of the motor and of the system of compensation and control. A method to increase the microcomputer utilization factor is also studied.

CONTEÚDO

	Pág.
CAPÍTULO 1 - INTRODUÇÃO -----	1
1.1 - O Microcomputador e Sistemas de Controle	2
1.2 - Objetivo e alcance deste trabalho -----	3
CAPÍTULO 2 - "HARDWARE" DO SISTEMA -----	7
2.1 - Introdução -----	7
2.2 - O Microcomputador -----	11
2.3 - Codificador de Velocidade -----	17
2.4 - Circuito de Potência - O Chaveador -----	20
2.5 - Circuito de Gatilhamento -----	22
2.6 - Esquema de Proteção -----	30
CAPÍTULO 3 - ORGANIZAÇÃO DO "SOFTWARE" E CARACTERÍSTICAS DE CONTROLE -----	34
3.1 - Introdução -----	34
3.2 - Tarefas do Sistema de Controle -----	37
3.2.1 - Inicialização -----	37
3.2.2 - Partida -----	37
3.2.3 - Regime -----	40
3.2.4 - Vídeo / Teclado -----	46
3.2.5 - Sobrecorrente -----	46
3.2.6 - Reinicialização -----	48
3.2.7 - Ajustes -----	49
3.3 - Considerações Finais -----	50
CAPÍTULO 4 - TESTES EXPERIMENTAIS E CONCLUSÕES -----	53
APÊNDICE A - Programa de Apoio Mínimo - PAM -----	68
1 - Apresentação -----	68
2 - Modo de Operação -----	68
3 - Funções do PAM -----	69
4 - Manipulação das Interrupções -----	71
5 - Subrotinas Utilitárias do PAM -----	74
APÊNDICE B - COMANDOS E MENSAGENS -----	76
1 - Comandos -----	76
2 - Mensagens -----	79

	Pág.
APÊNDICE C - ORGANIZAÇÃO DOS PARÂMETROS DE CONTROLE -----	82
APÊNDICE D - LISTAGEM DOS PROGRAMAS -----	86
1 - Programa de Apoio Mínimo -----	87
2 - Programas do Sistema de Controle -----	92
APÊNDICE E - RESULTADOS EXPERIMENTAIS -----	111
REFERÊNCIAS BIBLIOGRÁFICAS -----	113

TABELA DE FIGURAS

	Pág.
CAPÍTULO 2	
Fig. 2.1 - Diagramas de blocos do sistema de controle ("Hardware") -----	9
Fig. 2.2 - Diagrama de blocos detalhado do Sistema --	10
Fig. 2.3 - Arquitetura do microcomputador -----	12
Fig. 2.4 - Mapeamento de memória do microcomputador -	14
Fig. 2.5 - Circuito completo do microcomputador -----	16
Fig. 2.6 - Circuitos do codificador de velocidade-Formador de pulso e multiplicador de frequência -----	18
Fig. 2.7 - Diagrama de tempo do codificador de velocidade -----	19
Fig. 2.8 - Configuração do Chaveador -----	21
Fig. 2.9 - Principais formas de onda no funcionamento do chaveador -----	23
Fig. 2.10 - Diagrama de blocos do gerador de pulsos--	25
Fig. 2.11 - Circuitos do gerador de pulsos -----	26
Fig. 2.12 - Diagrama de tempo do bloco 3 (GPG) -----	29
Fig. 2.13 - Diagrama de tempo do bloco 11 (GPG) -----	29
Fig. 2.14 - Circuito sensor de corrente e o comparador Schmitt-Trigger -----	32
CAPÍTULO 3	
Fig. 3.1 - Diagrama de blocos do sistema de controle ("software") -----	36
Fig. 3.2 - Fluxograma do esquema de partida controlada	39
Fig. 3.3 - Comportamento dinâmico da tarefa REGIME --	41
Fig. 3.4 - Fluxograma das operações em REGIME -----	42
Fig. 3.5 - Fluxograma da subrotina de contagem -----	43
Fig. 3.6 - Característica do controlador de velocidade -----	45
Fig. 3.7 - Fluxograma da rotina de atendimento a IRQ-	47
CAPÍTULO 4	
Fig. 4.1 - Fluxograma da subrotina de contagem modificada para determinar o tempo de resposta do motor -----	56
Fig. 4.2 - Fluxograma da subrotina de contagem para obtenção de intervalos de amostragem $T \leq 200\text{mseg.}$	58
Fig. 4.3 - Fluxograma da rotina de atendimento a NMI-	60

	Pág.
Fig. 4.4 - Fluxograma da rotina de atendimento a IRQ ----	60
Fig. 4.5 - Diagrama de tempo das operações durante a contagem por interrupções -----	61
Fig. 4.6 - Base de tempo e controle das interrupções para a contagem da velocidade -----	64
Fig. 4.7 - Modificações efetuadas no circuito do microcomputador -----	66

CAPÍTULO 1

INTRODUÇÃO

O controle de velocidade de motores é essencial em várias aplicações industriais. Os sistemas de controle tem acompanhado a evolução dos componentes disponíveis para realizá-los. No passado, o sistema motor-gerador (Ward-Leonard) era amplamente utilizado no controle de motores d.c. A tecnologia de dispositivos de potência a semicondutores permitiu o uso de circuitos com inversores tiristorizados que substituíram com vantagem aquele método [1]. Os sistemas de controle analógicos [1], embora tenham boa performance, trazem consigo algumas desvantagens tais como: não linearidade, variação com a temperatura, ruídos externos, etc. Estes problemas aparecem tanto no transdutor de velocidade (tacogerador) como no circuito do controlador. Sistemas de controle totalmente digitais eliminam essas limitações [2]. O princípio de "phase-locked loop" (PLL) incorporado tanto em sistemas digitais [3] como em sistemas analógicos [4] constitui outra técnica importante no desenvolvimento de novos métodos de controle.

1.1. - O Microcomputador e Sistemas de Controle

Até o advento dos microprocessadores, os sistemas de controle tinham sua flexibilidade limitada a modificações no circuito. Com os microprocessadores, um novo recurso se tornou disponível, o "software", e essa limitação foi acentuadamente diminuída. Um mesmo sistema pode operar de vários modos diferentes mudando-se apenas os programas de controle. Preço, espaço e confiabilidade foram fatores que também contribuíram para a viabilidade do uso de microprocessadores nos sistemas de controle em geral e de velocidade de motores em particular.

O uso de microprocessadores abrange sistemas de controle de velocidade de motores d.c [5,6], a.c [7] até sistemas multimotor [8]. O tipo de controlador comumente usado tem característica não-linear. Consiste em se corrigir a velocidade, de uma maneira quantizada, conforme a magnitude e o sinal do erro. Esta técnica pode ser implementada por um programa que executa as operações (comparação, adição ou subtração) para selecionar o incremento ou decremento (e.g. do defasamento entre dois trens de pulsos [5]), ou uma tabela é previamente organizada com os valores de correção (e.g. do ângulo de gatilhamento [9]) em função do erro. Com este último método, as operações aritméticas são evitadas durante a execução do algoritmo de controle, pois o valor do erro endereça diretamente a tabela, entretanto, comparativamente, são requeridas mais locações de memória para sua implementação.

A maioria dos sistemas relatados até agora na literatura, usam o microprocessador incorporado apenas ao laço de velocidade para implementação do controle. Entretanto, esquemas incluindo o la-

ço de corrente, onde este pode inclusive englobar dois sub-laços (um com característica não-linear e outro com característica proporcional + integral, PI) tem sido abordados [10].

Os sistemas de controle a microprocessador são inerentemente discretos no tempo, nos quais a correção da velocidade é feita a intervalos regulares. Portanto existe um compromisso entre o período de amostragem (medição da velocidade) e a estabilidade. Para melhorar o comportamento dinâmico do sistema, esquemas multi-microprocessadores [11,28] podem ser usados. A versatilidade do microprocessador permite sua utilização em circuitos que usam o princípio de PLL [12, 13, 14] até em esquemas tradicionais Ward-Leonard [15]. De certa forma, qualquer circuito que seja controlado através de pulsos, é facilmente operado por meio de um microprocessador, tal é o caso de circuitos tiristorizados (conversores, inversores, chaveadores, etc.).

Referindo-se apenas a controle de motores, é difícil se prever quando as potencialidades do microprocessador se esgotarão. O desenvolvimento de sistemas cada vez mais abrangentes [16] acompanha o surgimento de microprocessadores cada vez mais poderosos (palavras de 16 bits, instruções mais eficazes, etc) e mais rápidos no mercado.

1.2. - Objetivo e alcance deste trabalho

A partir da experiência adquirida com um trabalho de estrutura semelhante ao de Lin [5], porém com alguns recursos adicionais, anteriormente implementado [17], partiu-se para o desenvolvimento de um sistema de propósitos gerais. Este possui tal flexibi-

lidade para modificações dos parâmetros de controle (velocidade de referência, ganho do controlador, etc) que, a partir da definição das características do processo (no caso um motor d.c), tais como: tensão nominal, torque, constante de tempo mecânica, etc, é possível dimensionar um sistema de controle para um caso específico.

A interação entre o operador e o processo é a característica mais importante no presente projeto. Esta interação é efetuada através de um terminal de vídeo/teclado e consiste de comandos acionados via teclado e de mensagens descritivas do estado do sistema na tela do terminal.

Num sistema de controle "on-line" usando microcomputador, é necessário minimizar o tempo em que o sistema permanece sem a atuação do processador executando operações de controle. Assim, intervenções que dependam estritamente do operador e atraso (espera) durante a comunicação (comandos ou mensagens) devem ser evitadas. Para tanto, um sistema baseado em interrupções é a solução adotada.

O desenvolvimento de "software" para sistemas de controle por microcomputador é atualmente a parte mais crítica e também a mais onerosa, principalmente se a programação tem de ser feita em linguagem assembler (como no presente sistema). Para contornar essa dificuldade, linguagens de alto nível para microprocessadores voltadas a aplicação no controle industrial têm sido desenvolvidas |18|.

A parte principal do "software" deste trabalho é o programa que controla os pedidos de interrupção. Ele permite principalmente uma comunicação "on-line" com o operador e o acionamento de

tarefas específicas concernentes à situações especiais do programa (parada do motor, aceleração/desaceleração controlada, etc).

Existem várias maneiras de se implementar (se por meio de "hardware" ou "software") as principais operações de um sistema típico de controle de velocidade [17]. Estas operações são: 1) Medição da Velocidade, 2) Implementação do algoritmo de controle e 3) Geração dos sinais adequados para o circuito de potência. A alternativa aqui adotada inclui um microprocessador MC6800 (Motorola) para executar por meio de programas a medição da velocidade e a implementação do algoritmo de controle e um circuito separado para gerar os comandos do estágio de potência que aciona o motor.

O circuito de potência é um chaveador d.c descrito na Referência [19]. O princípio usado para variar a velocidade do motor d.c (com excitação independente) é o da Modulação em Largura de Pulso (MLP). A velocidade atual é obtida medindo-se a frequência de um trem de pulsos gerado por um tacogerador digital. Este valor é então comparado com a velocidade de referência (previamente armazenada em uma locação de memória) através do microprocessador, no qual é gerado um número binário de acordo com o valor do erro. Por meio de um gerador de pulsos de gatilhamento dos tiristores do chaveador, este número controla o tempo em que o tiristor principal do mesmo permanece conduzindo, variando assim a tensão média de armadura e conseqüentemente a velocidade. O intervalo de medição da velocidade é de 200 msec. e a precisão do sistema é de 1 rpm em regime.

No Capítulo 2 são descritos os circuitos do sistema. . No Capítulo 3 são descritas a organização do "software" e as caracte-

rísticas do sistema de controle. Os testes realizados e conclusões são abordados no Capítulo 4. No Apêndice A é descrito um programa de apoio, cujos recursos são usados tanto na fase de depuração dos programas como durante a execução de operações de controle. Uma descrição detalhada dos comandos e mensagens disponíveis ao operador é mostrada no Apêndice B. Como organizar os parâmetros de controle do processo é explicado no Apêndice C. No Apêndice D é apresentada uma listagem completa dos programas e no Apêndice E os resultados de alguns testes.

CAPÍTULO 2

"HARDWARE" DO SISTEMA

2.1 - INTRODUÇÃO

Existem várias maneiras de se implementar um sistema de controle de velocidade de motores baseado em microprocessador. Os sistemas, como apresentados no capítulo anterior, são diferenciados basicamente pelo tipo de aplicação a que se destinam. A precisão no controle, tempo de resposta, recursos de proteção, tipo de motor ac ou dc, etc, são fatores que determinam as características do sistema conforme o caso. Estes requisitos estabelecem qual o tipo de circuito de potência que aciona o motor (inversor, conversor, chaveador, etc), o tipo de controlador (não-linear, PI, PID, etc) a ser empregado, quais as malhas de realimentação (velocidade e/ou corrente), como são gerados (por "hardware" ou "software") os comandos para o circuito de potência, etc. Baseada nisso, a escolha do microprocessador adequado [6] é uma etapa muito importante no projeto do sistema.

O presente sistema de controle é predominantemente digital e contém apenas a malha de realimentação de velocidade. Nesta malha (Fig. 2.1) estão incluídos o microcomputador (executando as funções de controlador e contador de velocidade), o circuito gerador de pulsos de gatilhamento - GPG, o chaveador, o codificador de velocidade - CV e o transdutor ótico de velocidade acoplado ao eixo do motor. Externamente tem-se a ponte retificadora trifásica para alimentar o chaveador, o sensor de corrente - SC para detectar sobrecorrente de armadura e a alimentação independente da corrente de campo. Inclui-se ainda, um terminal de vídeo/teclado para comunicação entre o operador e o sistema.

A Fig. 2.2 mostra o diagrama de blocos detalhado do sistema. Os blocos que compõem a malha de realimentação interagem durante a operação em regime do sistema de controle da seguinte maneira: No transdutor de velocidade (tacogerador) é gerado um trem de pulsos de frequência proporcional à velocidade do motor. Estes pulsos são formatados e multiplicados por 10 (devido a resolução estabelecida de 1 pulso = 1 rpm, ver seção 2.3) no codificador de velocidade. O bloco de interface do microcomputador recebe estes pulsos e após o intervalo de contagem, um número correspondente ao valor da velocidade atual está calculado. Este valor é comparado com o de referência e em função da magnitude do erro é gerado um número binário (N). Este número, através dos contadores programáveis do GPG, efetua a correção no ciclo de trabalho do trem de pulsos de potência aplicado à armadura do motor. As operações que não ocorrem durante o funcionamento em regime, entre elas as acionadas por interrupção, serão abordadas no capítulo seguinte.

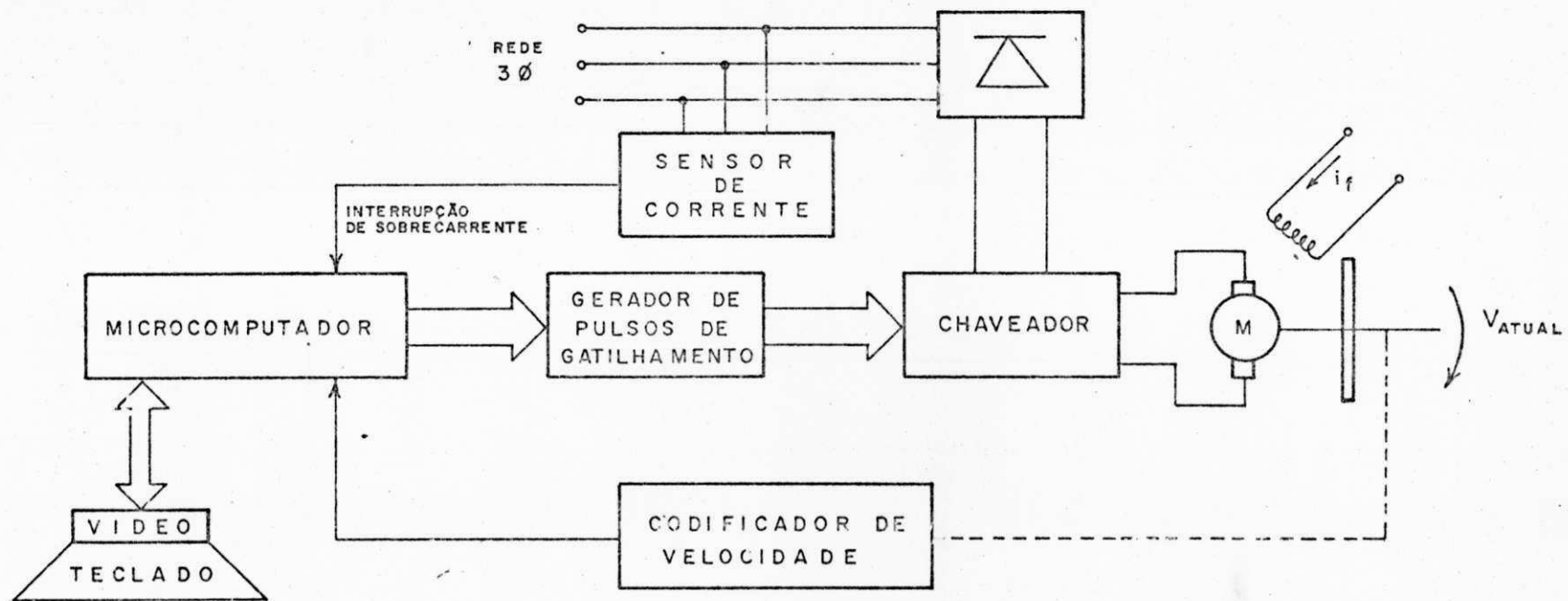


Fig. 2.1 - Diagrama de blocos do sistema de controle.

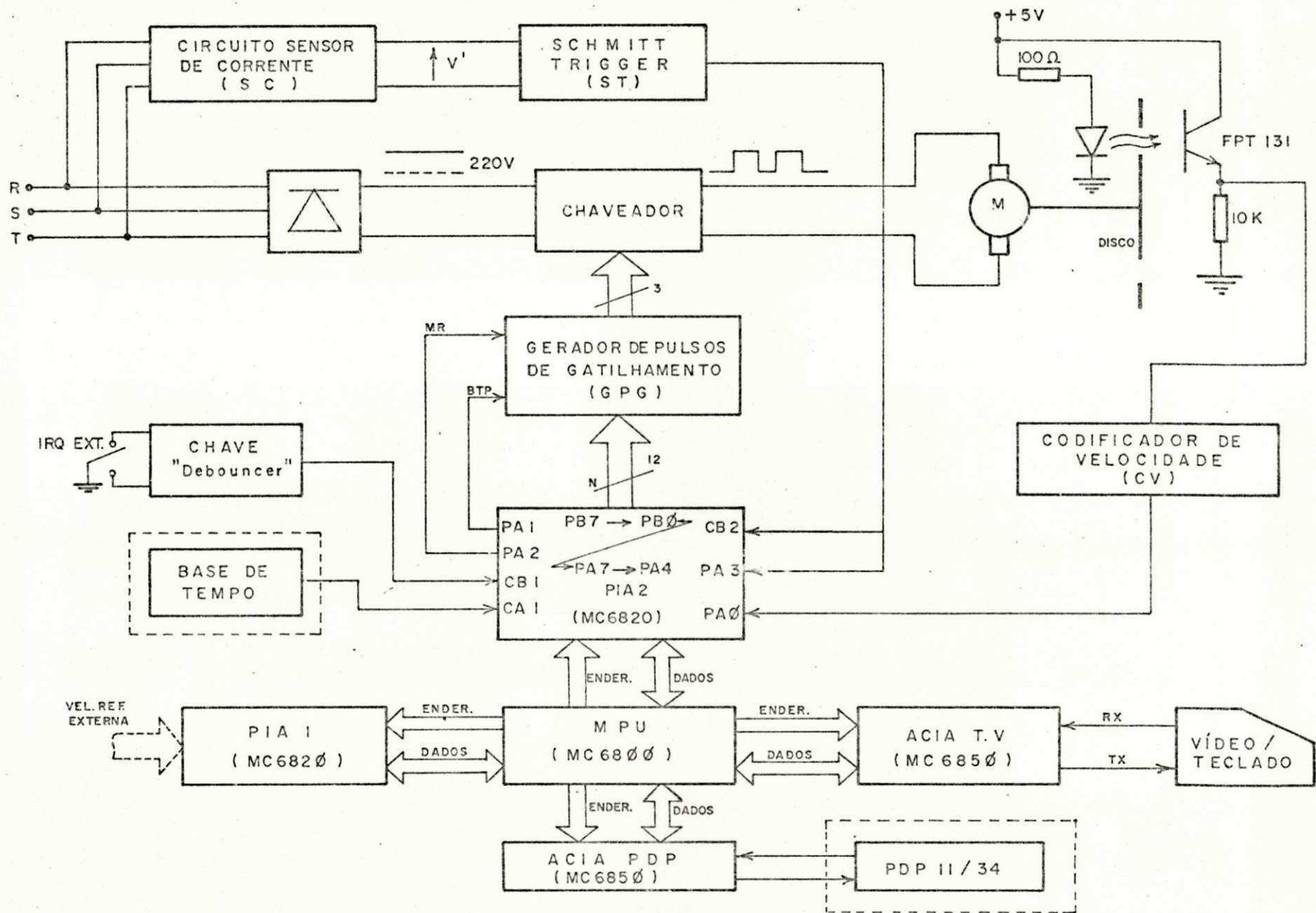


Fig. 2.2 - Diagrama de blocos detalhado do sistema.

2.2. - O Microcomputador

O presente sistema de controle utiliza um microcomputador baseado na família M6800 da Motorola. As operações implementadas no microcomputador são: medição da velocidade, execução do algoritmo de controle, geração dos sinais adequados para correção da velocidade e atendimento aos pedidos de interrupção. A arquitetura do microcomputador é mostrada na Fig. 2.3.

Devido à disponibilidade de blocos funcionais de interface para periféricos seriais-ACIA ("Asynchronous Communications Interface Adapter"), para entrada e saída de dados em paralelo - PIA ("Peripheral Interface Adapter") e manejo automático da pilha ("stack") durante as interrupções e encadeamento de subrotinas, o MC6800 tem sido utilizado em vários sistemas de controle [20, 21]. Algumas de suas características são: palavras de 8 bits, "clock" variável de 100 KHz a 1 MHz, alimentação única de 5 volts e via de saída de dados capaz de acionar uma carga padrão TTL e até 130 pF. Assim as ligações entre o circuito de interface de saída de dados do microcomputador e o GPG são feitas diretamente, bem como as ligações para pedidos de interrupção e da entrada de dados vindos do codificador de velocidade. O MC6800 aceita 3 tipos de interrupção acionados por hardware: RESET (geralmente usado para inicializações), IRQ - "Interrupt Request" (mascarável) e NMI - "Nonmaskable Interrupt" (não mascarável). A outra interrupção é acionada pela instrução SWI - "Software Interrupt". Detalhes sobre as interrupções do MC6800 são dados no Apêndice A.

A memória tipo RAM ("Random Access Memory") é utilizada durante a depuração dos programas, para armazenamento dos parâme-

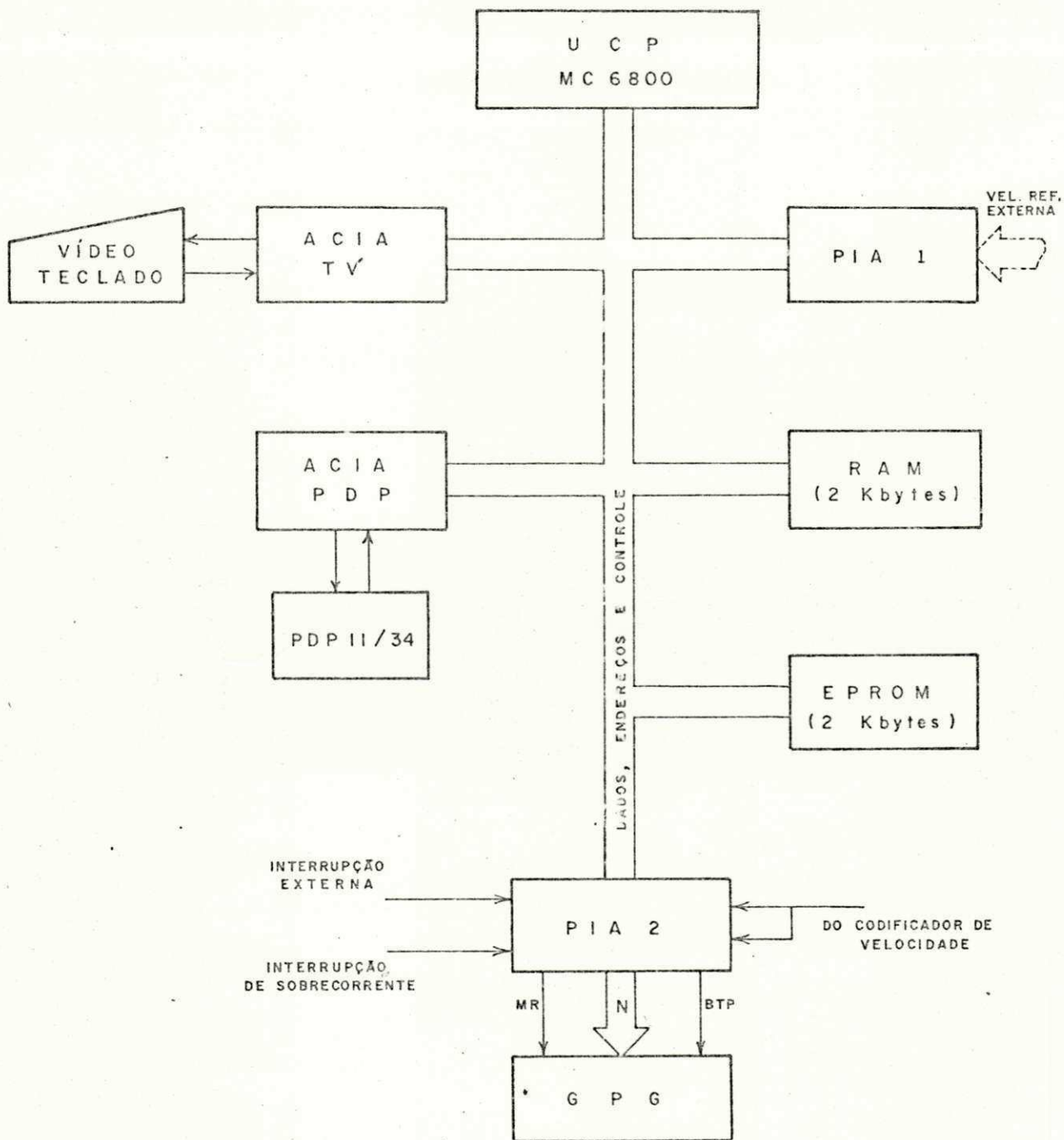


Fig. 2.3 - Arquitetura do microcomputador.

tros de controle e para definir o tamanho pilha, na qual são guardados os registradores do programa quando este é interrompido ou quando ocorre uma chamada de subrotina. Na EPROM ("Erasable Programmable Read-Only Memory") estão gravados o programa de apoio (Apêndice A), vetor de interrupção, mensagens, rotinas de atendimento à interrupções, etc. A Fig. 2.4 mostra o mapeamento de memória do microcomputador (endereços na base hexadecimal).

Na família M6800, a ACIA é equivalente a uma UART ("Universal Asynchronous Receiver/Transmitter"). Ela executa a transmissão serial de dados do microprocessador (que só opera com dados em paralelo, 8 bits de cada vez) para a tela do terminal e recebe dados em série do teclado e os converte em paralelo para o microprocessador. A taxa de transmissão de dados é programada pelo usuário. No presente caso a taxa é de 1200 bps ("bits" por segundo). Durante a operação em regime, a entrada de dados é feita através de interrupção a cada caracter, ou seja, não é necessário verificar se o registro de recepção de dados está cheio ou se o registro de transmissão de dados está vazio. A ACIA é programada para requisitar interrupção cada vez que uma dessas situações está estabelecida.

A transferência de dados paralelo é executada através da PIA. Esta consiste de duas partes (Parte A e Parte B) e cada uma possui um registrador de dados de 8 "bits". Estes "bits" podem ser programados individualmente como entrada ou saída. Eles estão identificados como PA_i e PB_i (i variando de 0 a 7) na Fig. 2.2. Pedidos de interrupção (IRQ) ao microprocessador, podem ser efetuados através da PIA. Esta possui um registrador de controle, no qual um dos "bits" indica a ocorrência de um pedido de interrupção até

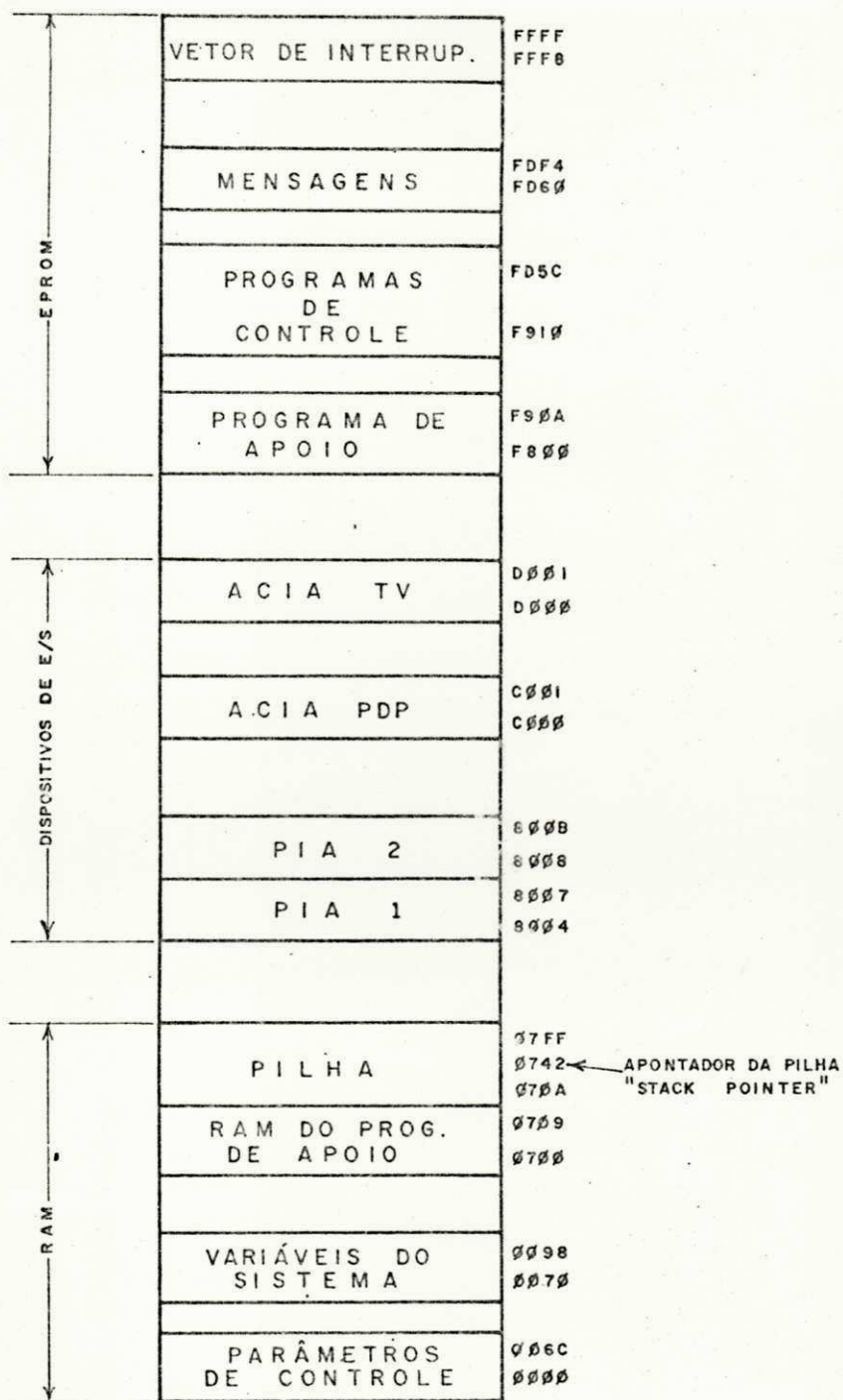


Fig. 2.4 - Mapeamento de memória do microcomputador.

que uma operação de leitura ("LOAD") seja efetuada no registrador de dados da mesma. Assim, um pedido de interrupção que foi mascarado (inibido), permanece registrado e é atendido logo após a máscara de interrupção ser desabilitada através de instruções apropriadas. Maiores detalhes sobre os componentes da família M6800 são dados na Referência [22].

O sistema M6800 usa o conceito de "E/S mapeado na memória", ou seja, os seus dispositivos de interface ocupam locações pré-definidas de memória (Fig. 2.4). Assim, não existem instruções específicas de E/S (como em alguns microprocessadores). A transmissão de dados pela ACIA é efetuada através de instruções tipo "LOAD" e "STORE". Para a PIA, além destas, instruções de operações lógicas (AND, OR, etc.), operações aritméticas (soma, subtração, incremento, decremento, etc) e de deslocamento ("rotate", "shift", etc) que envolvam o endereço correspondente ao registrador de dados, podem ser usadas. Estes recursos podem facilitar bastante a programação, como no caso de se desejar enviar a um circuito externo (ou a vários) uma determinada seqüência de pulsos. O controle das linhas MR ("Master Reset") e BTP (Bloqueia Tiristor Principal) (ver Fig. 2.3) é feito por programa e será abordado no Capítulo 3. A Fig. 2.5 mostra o circuito completo do microcomputador. A utilização da ACIA PDP e PIA1 (ver Fig. 2.2) bem como a ligação da base de tempo e da entrada NMI (usadas nos testes realizados de um método alternativo de medição da velocidade) constituem possíveis expansões do sistema e são comentadas no Capítulo 4.

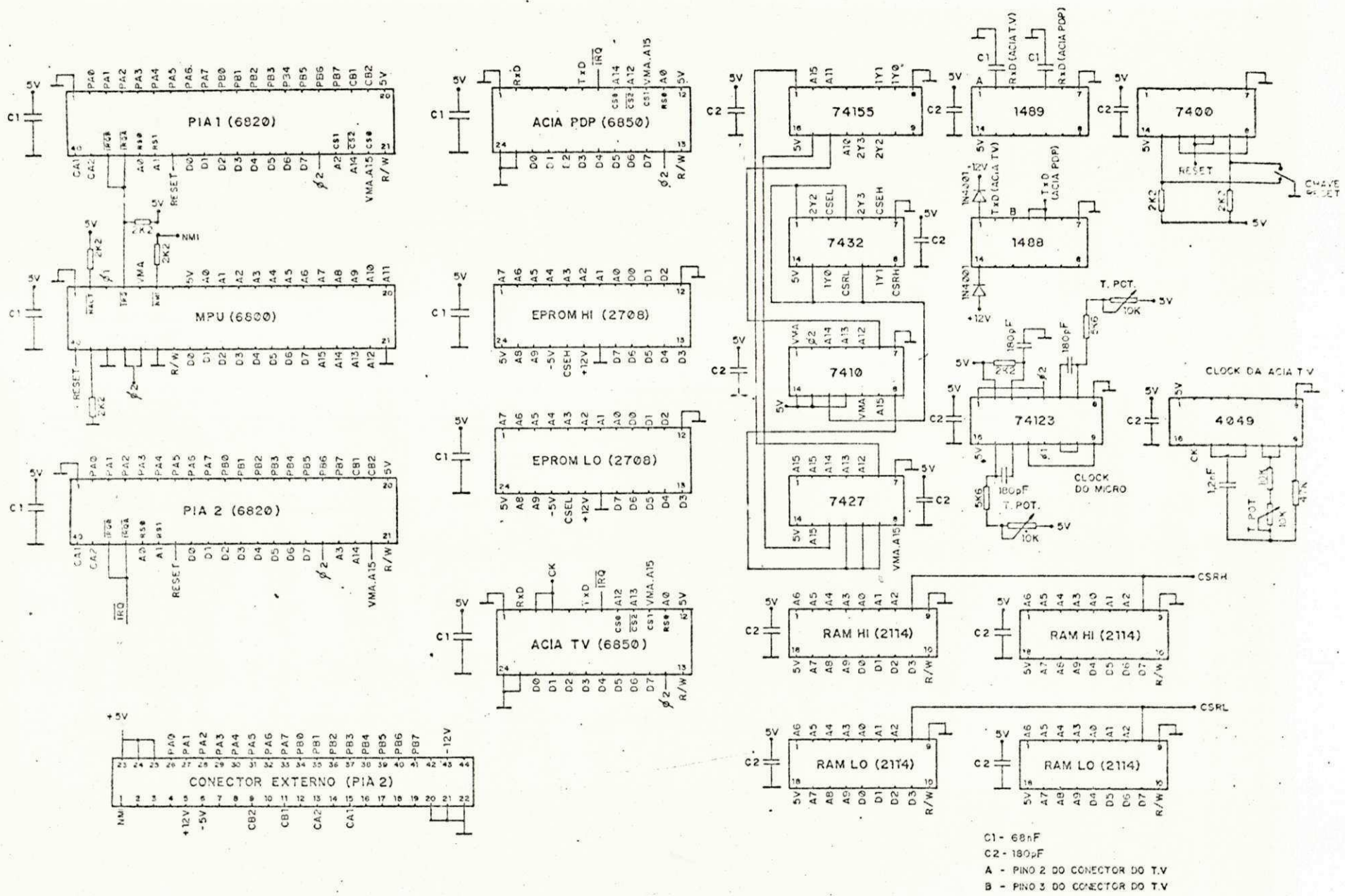


FIG. 2.5 - CIRCUITO COMPLETO DO MICROCOMPUTADOR.

- C1 - 68nF
- C2 - 180pF
- A - PINO 2 DO CONECTOR DO TV
- B - PINO 3 DO CONECTOR DO TV

2.3. - Codificador de Velocidade

Os pulsos contados no microcomputador para efetuar a medição de velocidade, são inicialmente gerados por um tacogerador digital. Este é constituído de um disco perfurado, na periferia, acoplado ao eixo do motor, que gira entre um LED e um fototransistor. Cada vez que um furo passa pelo feixe de luz, um pulso é gerado. De talhe da implementação do tacogerador é dado na Fig. 2.2.

O intervalo de contagem dos pulsos, ou seja, da medição de velocidade é de 200 msec. Como a resolução para o esquema de contagem é de 1 pulso equivalente a 1 rpm, é necessário se ter 300 pulsos por revolução do disco. Assim, o número de pulsos contados durante os 200 msec., fornece diretamente o valor da velocidade em rpm, de acordo com a expressão abaixo:

$$X = \frac{V \cdot P \cdot A}{60} \quad \text{pulsos}$$

onde V - velocidade do motor em rpm

P - número de pulsos por revolução do disco

A - intervalo de contagem em segundos

X - número de pulsos durante o intervalo A

O codificador de velocidade é constituído de um circuito que formata os pulsos vindos do tacômetro e de um multiplicador de frequência por 10 baseado no PLL-4046. As Figs. 2.6 e 2.7 mostram a implementação do CV e o diagrama de tempo do mesmo. A multiplicação foi necessária devido a problemas mecânicos para se construir um disco com 300 furos. Usou-se então um disco com 30 furos e o

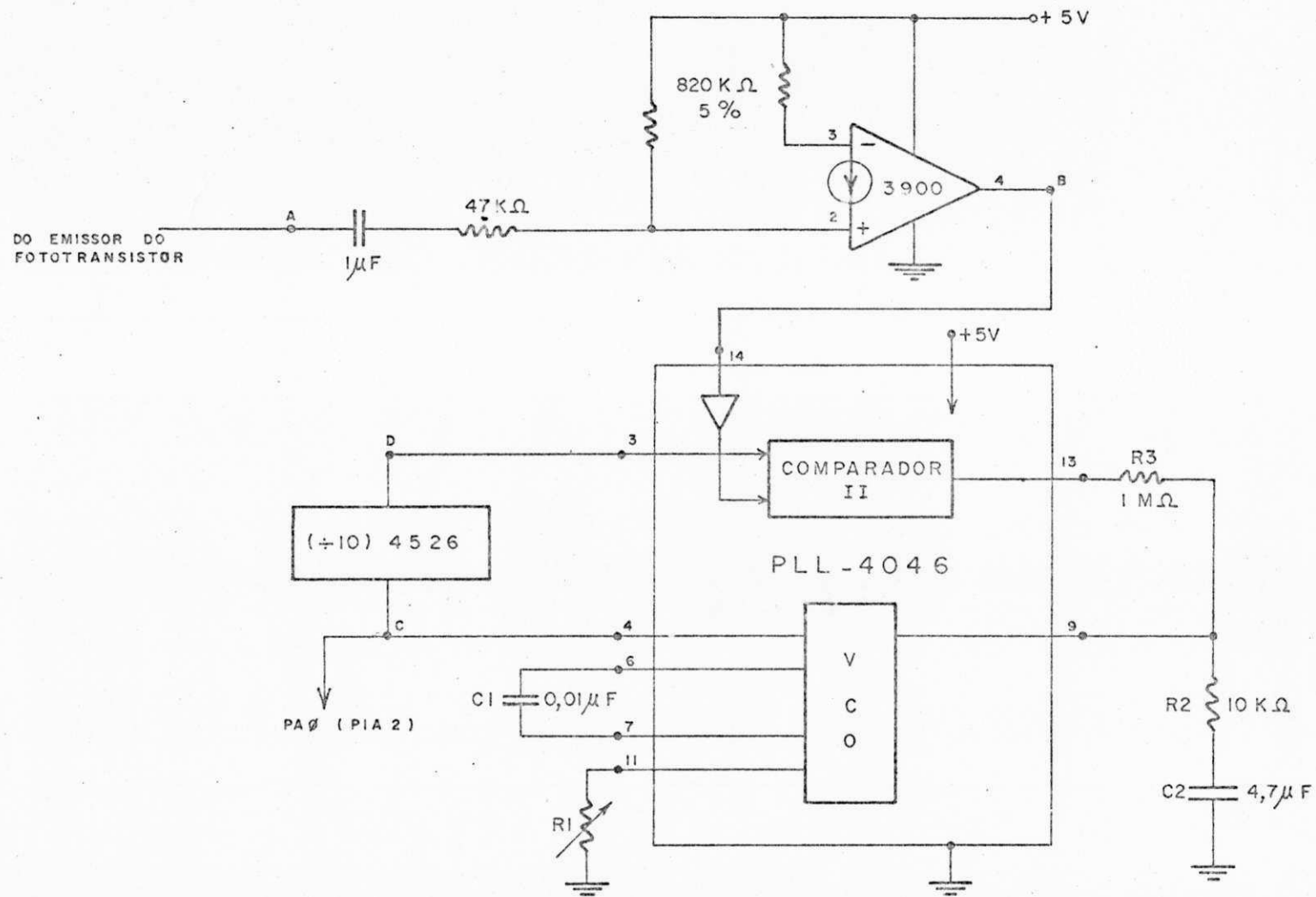


Fig. 2.6 - Circuitos do codificador de velocidade - Formador de pulso + multiplicador de frequência.

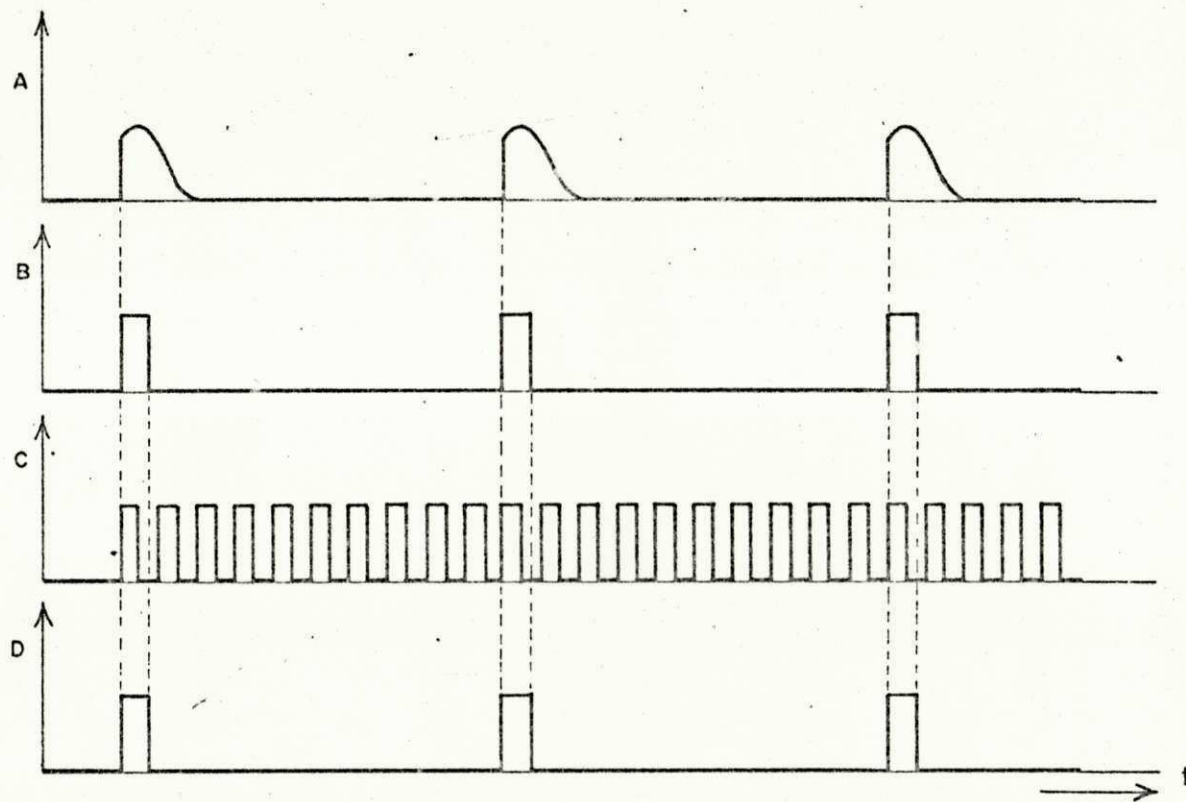


Fig. 2.7 - Diagrama de tempo do codificador de velocidade.

multiplicador para se obter a resolução desejada.

2.4. - Circuito de Potência - O Chaveador [19]

Existem vários circuitos de potência para acionamento e controle de motores dc, como já citado no Capítulo 1, No presente sistema usou-se um chaveador, o qual converte uma tensão dc constante de entrada numa tensão dc variável de saída pelo processo de Modulação em Largura de Pulso. Neste chaveador utiliza-se um tiristor principal - T_p e quatro tiristores auxiliares, os quais revertem a polaridade do capacitor de comutação C (Fig. 2.8). Esta configuração, dependendo do circuito de gatilhamento, pode apresentar um funcionamento independente da carga, como será explicado na seção seguinte. Os comandos para habilitar/bloquear os pulsos de gatilhamento do chaveador (MR e BTP) são controladas pelo microcomputador (ver Capítulo 3). Quando o comando de habilitar é acionado, tem início uma sequência de operações descrita abaixo:

a) Os tiristores TA1 e TA4 são gatilhados simultaneamente, o capacitor C se carrega com aproximadamente a tensão de entrada E e polaridade a^+ e b^- bloqueando assim esses tiristores (comutação natural) e preparando o capacitor para comutar o tiristor principal com o próximo pulso de gatilhamento de TA2,3. É indispensável para o bom funcionamento do chaveador, que o primeiro pulso (após o comando de habilitar), gatilhe um par de tiristores auxiliares, no caso o par TA1,4.

b) O tiristor T_p é então gatilhado, permitindo a aplicação da tensão de entrada E na armadura do motor.

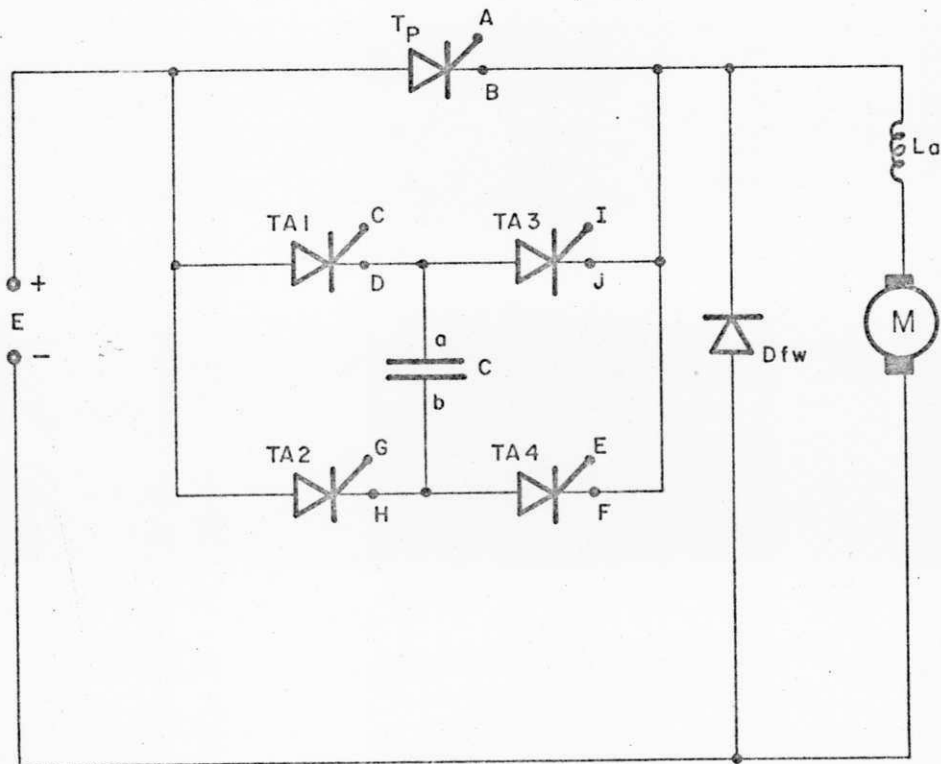


Fig. 2.8 - Configuração do chaveador

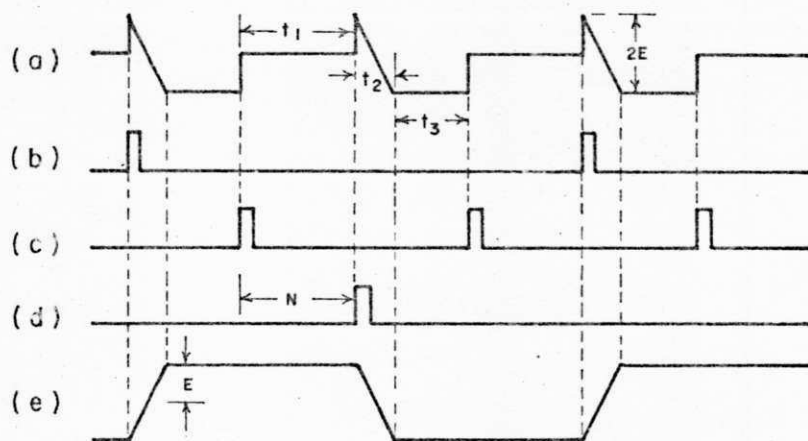
c) Os tiristores TA2 e TA3 são gatilhados simultaneamente, aplicando a tensão do capacitor reversamente no tiristor principal, comutando-o (comutação forçada). Após um intervalo t_2 (Fig. 2.9), a tensão no capacitor está invertida (a^- e b^+), todos os tiristores estão bloqueados e a corrente do motor passa pelo diodo de circulação livre ("free-wheeling" - Dfw).

d) O tiristor Tp é gatilhado, aplicando novamente a tensão de entrada na armadura do motor. Este tiristor será comutado em seguida pela condução dos tiristores TA1,4 e o ciclo se repete.

A Fig. 2.9 mostra as formas de onda mais ilustrativas relacionadas com a operação do chaveador. O intervalo t_1 corresponde ao tempo decorrido após N (número gerado pelo microprocessador) períodos do "clock" dos contadores do GPG (ver seção 2.5, bloco 2). O tempo decorrido a partir do bloqueio do tiristor principal (quando um dos pares de tiristores auxiliares entra em condução) até que o capacitor de comutação do chaveador tenha invertido de polaridade, está representado por t_2 . Este intervalo é variável, depende da corrente de armadura e varia com a carga do motor. O tiristor principal não deve ser gatilhado durante o intervalo t_2 , para que o chaveador possa operar independente da variação de carga (ver seção 2.5, bloco 11). Durante o intervalo de circulação livre (t_3), a tensão aplicada ao motor é nula (todos os tiristores do chaveador estão bloqueados).

2.5. - Circuito de Gatilhamento

Como visto na seção anterior, o tiristor principal é comutado pelo processo de comutação forçada. Assim sendo, é necessário



(a) TENSÃO APLICADA AO MOTOR

(b) PULSOS DE GATILHAMENTO PARA OS TIRISTORES TA 1,4

(c) PULSOS DE GATILHAMENTO PARA O TIRISTOR PRINCIPAL

(d) PULSOS DE GATILHAMENTO PARA OS TIRISTORES TA 2,3

(e) TENSÃO DO CAPACITOR

Obs. A representação dos pulsos de gatilhamento está fora de escala.

Fig. 2.9 - Principais formas de onda no funcionamento do chaveador

que, com a condução de um par de tiristores auxiliares, o capacitor de comutação se carregue com a tensão adequada (aproximadamente igual a tensão de entrada do chaveador) para garantir a comutação do tiristor principal, quando o outro par de tiristores auxiliares for gatilhado. Esta condição é satisfeita, implementando-se circuitos para detetar o carregamento do capacitor de comutação e para inibir a geração de pulsos de gatilhamento para o tiristor principal durante o intervalo t_2 . Estes circuitos são mostrados na Fig. 2.11 (blocos 10 e 11 respectivamente). Caso contrário, se o tiristor T_p for gatilhado durante o intervalo t_2 , o esquema para comutação forçada (capacitor + tiristores auxiliares) do referido tiristor não tem condições de comutá-lo e a tensão de entrada do chaveador ficará aplicada à armadura do motor diretamente e sem controle, o que obviamente não é desejável.

O circuito do GPG baseia-se no uso do contador binário programável de 4 "bits" MC4526B, como divisor de frequência, o qual permite ligação em cascata sem portas adicionais. O pulso que aparece na saída "0" do contador a cada ciclo de contagem é usado para gatilhar os tiristores. As Figs. 2.10 e 2.11 mostram respectivamente o diagrama de blocos e os circuitos do GPG. Segue a descrição de cada um dos blocos.

Bloco 0 - Este bloco é constituído de um circuito de "clock" cuja frequência é de 1 MHz.

Bloco 1 - Este bloco é formado por 3 contadores MC4526B em cascata, formando um divisor de frequência por 2^{12} (4096) que gera os pulsos para gatilhar o tiristor T_p .

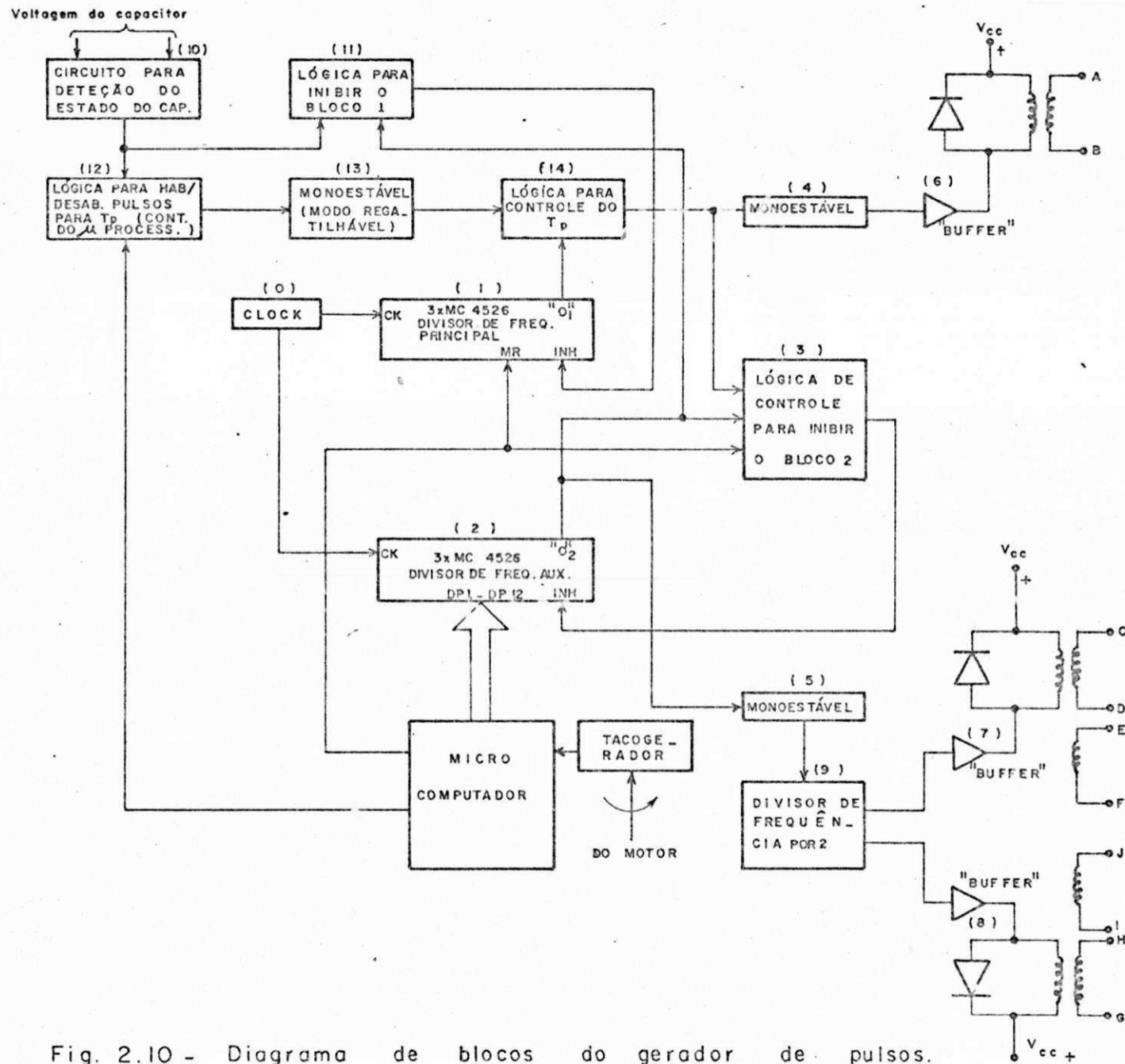


Fig. 2.10 - Diagrama de blocos do gerador de pulsos.

Bloco 2 - Este bloco é um divisor de frequência por N ($N < 4096$) com uma configuração semelhante a do bloco 1. Usa-se uma lógica tal que a saída deste bloco é atrasada em relação a saída do bloco 1 de N períodos do "clock". N é controlado pelo microprocessador e determina o tempo em que o tiristor T_p permanece conduzindo. Os pulsos na saída deste bloco gatilham os tiristores auxiliares. Na montagem atual também existe a possibilidade de controlar N manualmente.

Bloco 3 - Quando ocorre um pulso na saída " O_2 " do divisor de frequência por N , a lógica do bloco 3 atua de tal modo que o bloco 2 fica inibido até que ocorra o próximo gatilhamento do tiristor T_p , através dos pulsos na saída " O_1 " do bloco 1. Isto é necessário para que não ocorram dois pulsos consecutivos nos tiristores auxiliares visto que o período de contagem do bloco 2 é sempre menor que o período do bloco 1. Essa lógica também garante o sincronismo entre os divisores de frequência a cada novo ciclo. A Fig. 2.12 mostra o diagrama de tempo correspondente ao bloco 3.

Blocos 4 a 8- Estes blocos são constituídos de multivibradores monoestáveis para dar aos pulsos de gatilho a largura adequada e de pares Darlington de transistores que fornecem a corrente neces-

sária para gatilhar os tiristores.

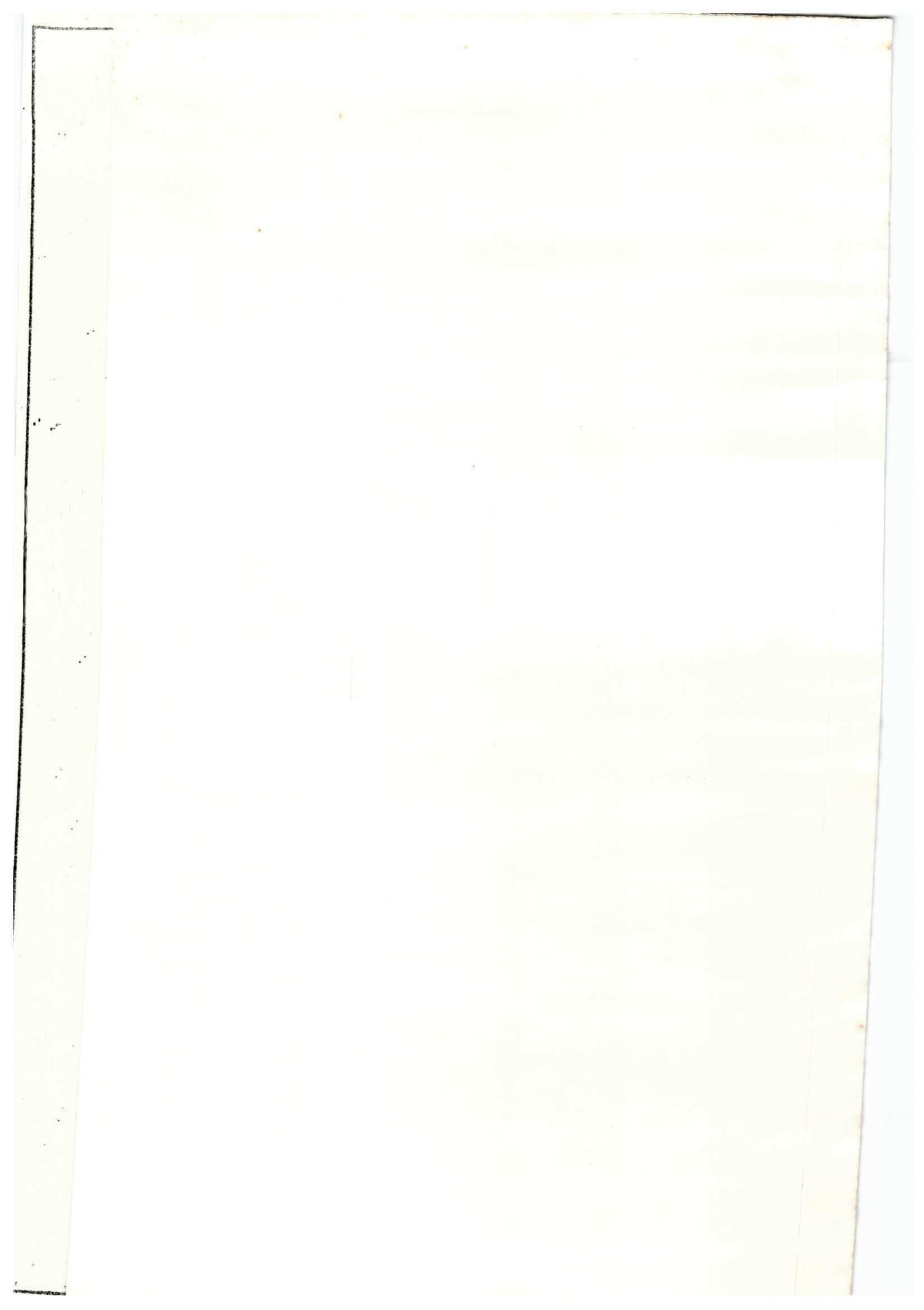
Bloco 9 - Este bloco é um divisor de frequência por dois para gatilhar alternadamente os pares de tiristores auxiliares. Este bloco também garante que o primeiro pulso de gatilhamento acicne os tiristores TA1,4.

Bloco 10 - Este circuito serve para detetar a tensão no capacitor de comutação do chaveador. A saída deste bloco indica se o capacitor se carregou com tensão suficiente (aproximadamente igual a tensão de entrada do chaveador) para comutar o tiristor principal.

Bloco 11 - Esta lógica de controle serve para inibir os contadores do bloco 1 durante o intervalo t_2 (ver Fig. 2.9). A saída deste bloco (M na Fig. 2.11) está ligada à entrada INH ("inhibit") do bloco 1. O estado da saída (M) só muda (para nível lógico baixo) quando ocorre um pulso na saída do bloco 10 (K na Fig. 2.11). A Fig. 2.13 mostra o diagrama de tempo do bloco 11.

Bloco 12 - Este bloco é controlado pelo microcomputador, o qual pode habilitar/bloquear o envio de pulsos de gatilhamento para o tiristor principal (através da linha BTP, Fig. 2.2). Este controle é essencial durante a parada automática do motor.

Bloco 13 - Este bloco é constituído de um multivibrador



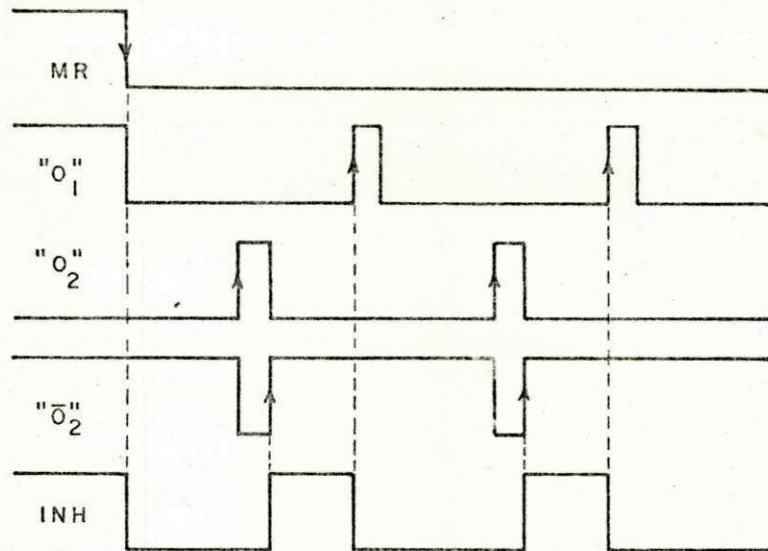


Fig. 2.12 - Diagrama de tempo do bloco 3
(INH é ativo em nível alto)

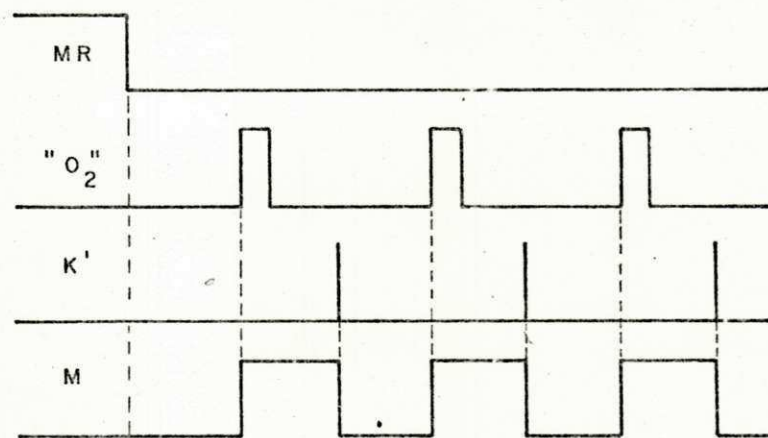


Fig. 2.13 - Diagrama de tempo do bloco 11.

monoestável, funcionando no modo regatilhável, acionado por pulsos gerados pela variação da tensão no capacitor de comutação (saída do bloco 10). Se o bloco 12 estiver habilitado e se os citados pulsos deixarem de ocorrer, o monoestável deixa de ser ativado, e após determinado intervalo de tempo sua saída muda de estado (para nível alto), evitando o gatilhamento do tiristor principal, através do bloco 14.

Bloco 14 - Uma porta OU controlada pela saída do bloco 13 constitui este bloco. Ela permite ou não o envio de pulsos de gatilhamento ao tiristor principal.

Maiores detalhes sobre o gerador de pulsos de gatilhamento são dados na Referência [24].

2.6. - Esquema de Proteção

Um sistema de controle de velocidade pode incluir vários tipos de proteção: limitação da corrente na partida, verificação da existência de corrente de campo, limitação da corrente de armadura, etc. O presente sistema além de possuir característica de partida controlada e um algoritmo de controle que evita variações bruscas de corrente durante a correção de grandes erros de velocidade, inclui também um esquema de proteção contra sobrecorrente de armadura. Esta proteção consiste no bloqueio dos tiristores do chavador, desacoplando a fonte de alimentação do mesmo do resto do

circuito. O esquema de proteção é acionado através de interrupção externa requisitada ao microprocessador. Um circuito sensor de corrente - SC e um comparador Schmitt-Trigger - ST formam o conjunto de circuitos que executa esse pedido de interrupção. A Fig. 2.14 mostra o circuito do sensor de corrente e do comparador. A seqüência de operações para efetuar o bloqueio do chaveador é executada através de programa e será abordada no capítulo seguinte.

O sensor de corrente é constituído de transformadores de corrente ligados à rede-3 Φ e de um circuito retificador. Obtém-se assim uma tensão V' proporcional à corrente fornecida pela rede. Para o circuito montado V' é da ordem de $1,2 V_{dc}$ com uma ondulação de aproximadamente $0,25 V_{pp}$ quando a corrente média atinge 6 A.

Para evitar que ocorram pedidos de interrupção consecutivos (várias transições na saída do comparador devido à ondulação de V'), usou-se um comparador com histerese (ST na Fig. 2.14). Quando a corrente de armadura atinge o valor especificado (6 A), a entrada do circuito de interface do microprocessador (CB2 na Fig. 2.2) irá detetar apenas uma transição.

A montagem dos circuitos do presente sistema foi feita por partes. O microcomputador, o GPG, e os outros circuitos - codificador de velocidade, comparador Schmitt-Trigger e chaves (com "debouncers") para requisitar interrupções externas, estão montados em 3 unidades (caixas metálicas) independentes com fontes de alimentação próprias. As ligações entre o tacogerador e o codificador de velocidade e entre as saídas do GPG e os gatilhos dos tiristores são feitas com fios blindados.

Os testes de funcionamento do gerador de pulsos de gati-

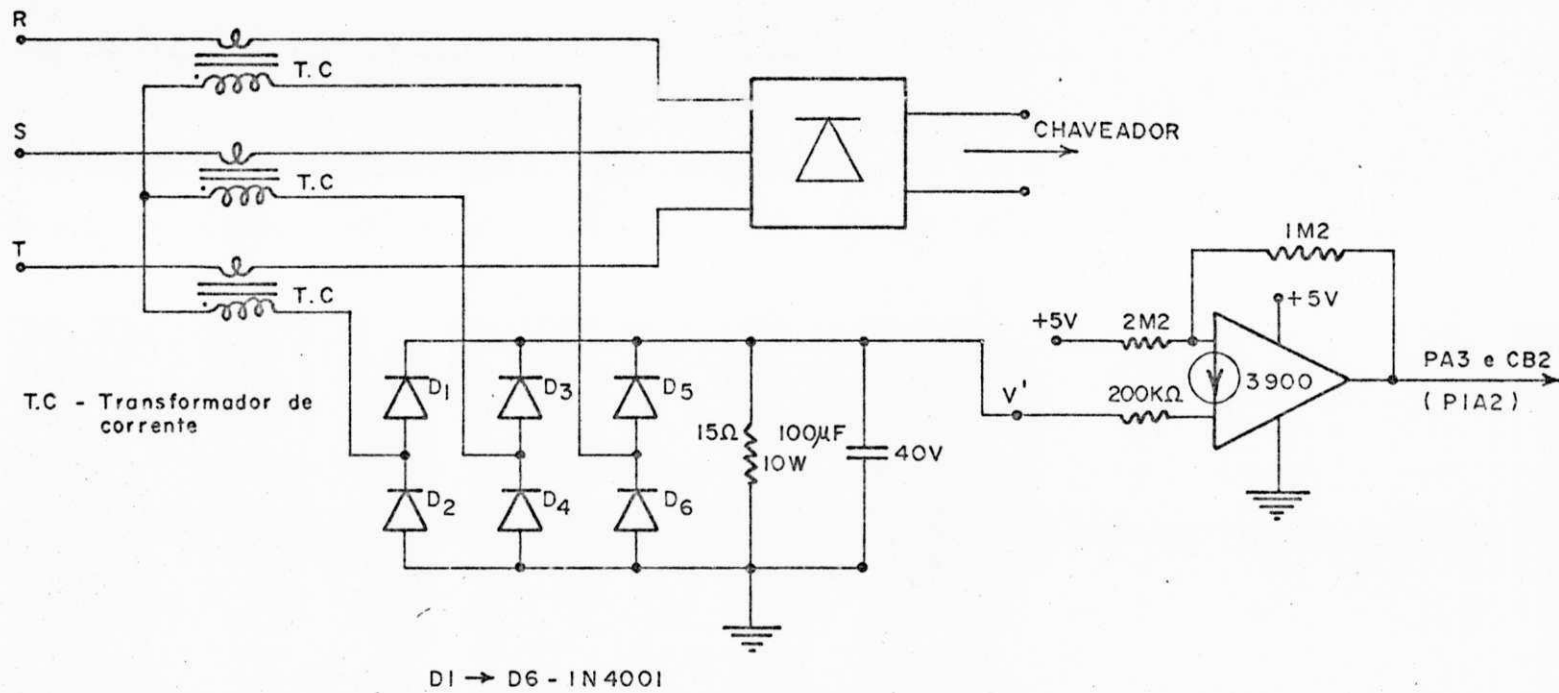


Fig. 2.14 - Circuito sensor de corrente e o comparador Schmitt-Trigger.

lhamento foram realizados utilizando-se chaves "toggle" (disponíveis na própria montagem) para controlar o ciclo de trabalho do trem de pulsos de potência na saída do chaveador. Para várias situações de carga, o sistema (GPC e chaveador) funcionou satisfatoriamente em uma faixa de 0,1 a 98% de ciclo de trabalho. Estes testes serviram para dimensionar o mínimo e o máximo valor de N , quando o controle através do microcomputador fosse realizado. O multiplicador de frequência do codificador de velocidade também foi testado separadamente. Houve necessidade de redimensionar o valor do capacitor C_2 (Fig. 2.6) para minimizar o atraso na resposta do PLL, durante a operação em malha fechada.

CAPÍTULO 3

ORGANIZAÇÃO DO "SOFTWARE" E CARACTERÍSTICAS DE CONTROLE

3.1. - INTRODUÇÃO

O rápido avanço na tecnologia de circuitos integrados (VLSI) tornou o desenvolvimento de "software" o fator determinante do custo total de sistemas de controle de processos, principalmente daqueles, baseados em microprocessadores.

Em sistemas de controle usando microcomputador em tempo real, além da obtenção, monitorização e correção dos parâmetros do processo, é importante se ter uma interação entre o operador e o processo. A partir das características de controle do motor, dos recursos do microcomputador e da utilização do terminal de vídeo/teclado, desenvolveu-se o "software" de controle baseado em tarefas acionadas por interrupções. As principais características do sistema de controle são:

- 1) A medição da velocidade é discreta no tempo, e ocorre a cada 200 mseg.
- 2) A entrada e saída de dados através do terminal é feita por interrupções a cada carácter.
- 3) Comandos externos podem interromper o sistema de controle.

A Fig. 3.1 mostra os módulos de "software" (i.e. tarefas do sistema de controle). A ativação das tarefas é feita pelo supervisor, o qual aceita as seguintes interrupções:

- a) RESET - Usada para inicializar a ACIA e a PIA, preparar o acesso aos endereços das interrupções (IRQ, SWI e NMI), determinar o endereço das variáveis utilizadas pelo supervisor e inicializar o conteúdo do apontador da pilha ("stack pointer").
- b) NMI - Pode ser usada para a contagem dos pulsos gerados pelo codificador de velocidade (ver Capítulo 4).
- c) SWI - Usada para efetuar correções em variáveis do processo durante a ocorrência de situações especiais do sistema de controle.
- d) IRQ - Usada para as interrupções através do terminal, interrupções externas e interrupções acionadas pelo

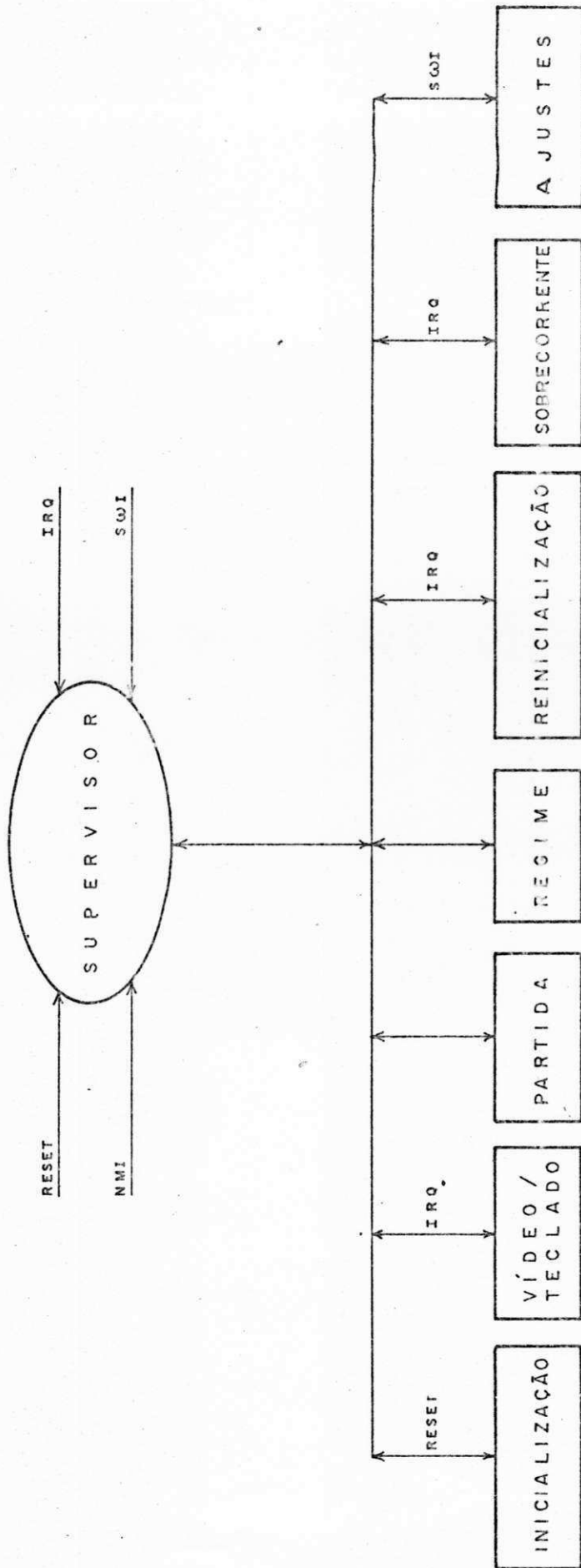


Fig. 3.1 - Diagrama de blocos do sistema de controle.

circuito no qual é gerada a base de tempo para efetuar a medição da velocidade (ver Capítulo 4).

O supervisor inclui um programa de apoio (Apêndice A) e uma rotina que aciona através de "polling" (prioridade fixa) as várias tarefas do sistema requisitadas por IRQ. A manipulação, efetuada pelo supervisor, dos pedidos de interrupção garante a mútua exclusão entre as tarefas, como será explicado no desenvolvimento deste capítulo. Várias rotinas e subrotinas funcionais (blocos) são compartilhadas pelas tarefas. Em seguida é feita a descrição das tarefas.

3.2. - TAREFAS DO SISTEMA DE CONTROLE

3.2.1 - INICIALIZAÇÃO

A tarefa INICIALIZAÇÃO é ativada logo após o acionamento da chave externa ligada a entrada RESET do microprocessador. Esta tarefa, além da execução automática das operações citadas acima (item a), inclui ainda a inicialização, via teclado, dos valores dos parâmetros de controle (ver Apêndice C) e o acionamento, também via teclado, da tarefa PARTIDA. Nenhuma ação de controle é efetuada durante a execução da tarefa INICIALIZAÇÃO.

3.2.2 - PARTIDA

As operações efetuadas por esta tarefa são: preparação da ACIA para requisitar interrupções na transmissão/recepção de caracte

teres, estabelecer a direção adequada de dados (entrada ou saída) do registrador de dados da PIA, preparação da PIA para aceitar interrupções externas, definição dos valores iniciais das variáveis que servem como indicadores de algumas situações especiais do processo e inicialização dos endereços das rotinas de interrupção. Em seguida a estes procedimentos, o microcomputador passa, efetivamente, a controlar o motor. As linhas MR e BTP (citadas no capítulo anterior) são ativadas quando o primeiro valor de N é enviado ao gerador de pulsos de gatilhamento e tem início, então, a partida controlada do motor. O esquema para executá-la consiste em dividir a diferença entre a velocidade final e a inicial (neste caso $V_{\text{final}} = V_{\text{refer.}}$ e $V_{\text{inic.}} = 0$) em partes iguais, determinando pontos de velocidade intermediária ($V_{\text{int.}}$) que são atingidos passo a passo como mostrado na Fig. 3.2. Este esquema tende a limitar a corrente de armadura durante a partida do motor. O motor gasta mais ou menos tempo para atingir a velocidade de referência de acordo com os valores dos seguintes parâmetros de controle previamente definidos pelo operador:

1. Valor inicial da tensão média aplicada ao motor (primeiro valor de N).
2. Número de partes em que é dividida a variação de velocidade.

Durante a partida o operador pode obter na tela do terminal o valor corrente da velocidade do motor. Uma mensagem informa que a velocidade final ($V_{\text{refer.}}$) foi atingida e o controle passa

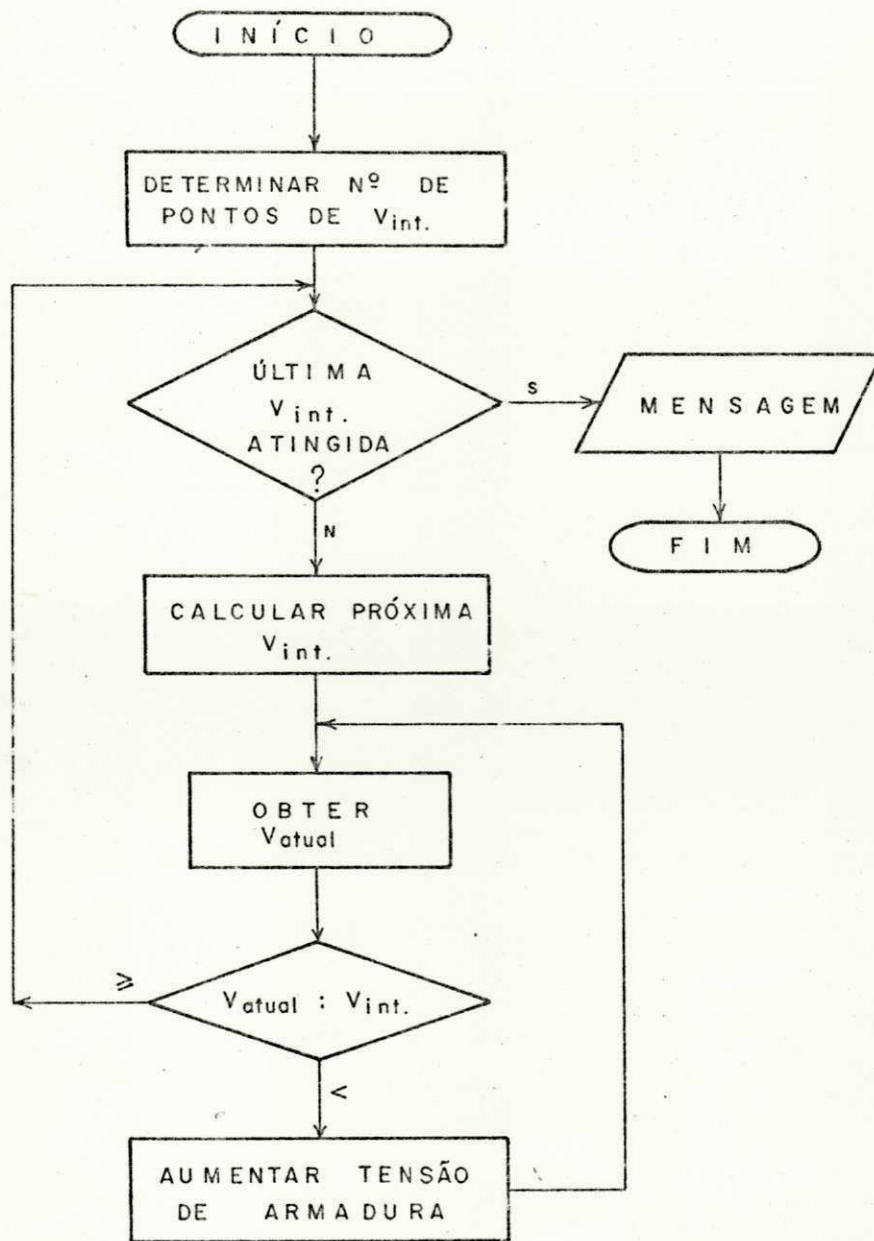


Fig. 3.2 - Fluxograma do esquema de partida controlada.

diretamente à tarefa REGIME.

3.2.3 - REGIME

Os blocos de "software" que interagem entre si durante a operação em regime estão mostrados na Fig. 3.3. O tempo decorrido para efetuar um ciclo completo de controle (laço 1 ou laço 2, na Fig. 3.3) é aproximadamente 200 msec., visto que o tempo de execução dos blocos 1 e 3 (Fig. 3.3) é de apenas algumas dezenas de microsegundos. Portanto, a probabilidade de ocorrência de pedidos de interrupção fora do intervalo de contagem (bloco 2) é desprezível. Como os pedidos de interrupção são mascarados durante o referido intervalo, tem-se um atraso máximo de 200 msec. para cada pedido de interrupção ser atendido. A prioridade de atendimento às tarefas que requisitam interrupção é estabelecida pelo supervisor por meio de "polling". É dada maior prioridade às tarefas envolvidas na comunicação entre o operador e o processo. A Fig. 3.4 mostra o fluxograma da tarefa REGIME. A medição da velocidade é efetuada de acordo com o fluxograma mostrado na Fig. 3.5. Durante a execução ao laço A é testado se o nível lógico da entrada (pulsos do codificador de velocidade) é "1". Quando ocorre uma transição para "0", o contador de pulsos é incrementado e o laço B é executado até que a entrada passe novamente a nível "1". Ambos os laços devem ter o mesmo tempo de execução.

A característica de controle da velocidade é não-linear como mostrado na Fig. 3.6. O método empregado consiste em aumentar ou diminuir (de uma maneira quantizada) a tensão média de armadura,

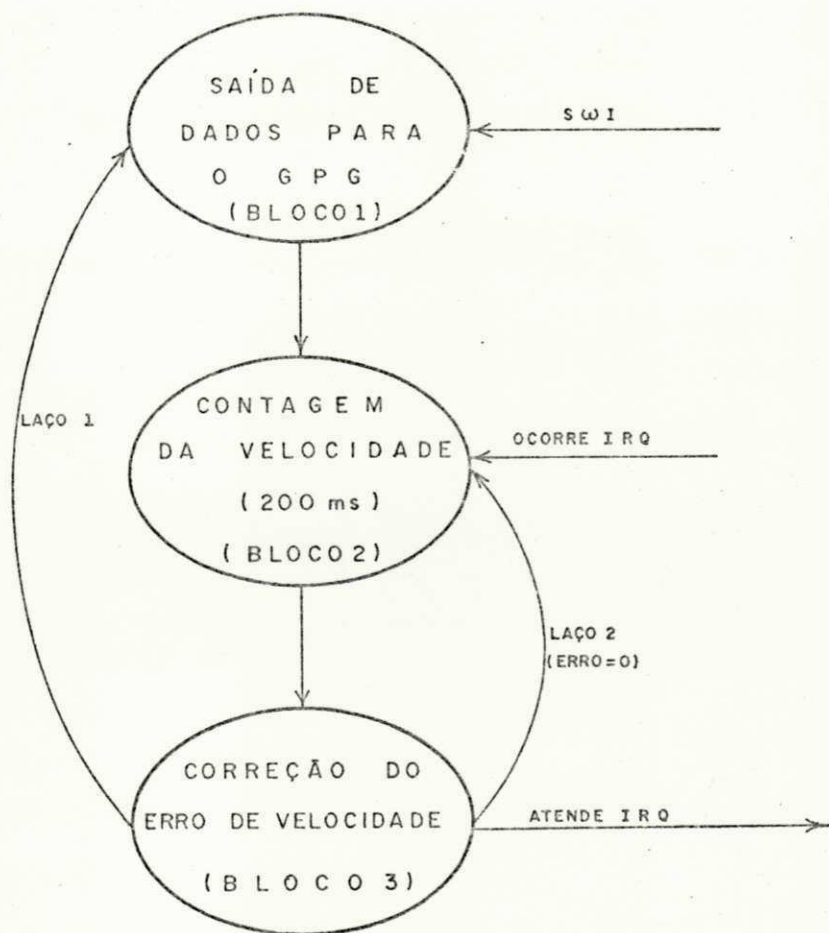


Fig. 3.3 - Comportamento dinâmico da tarefa REGIME.

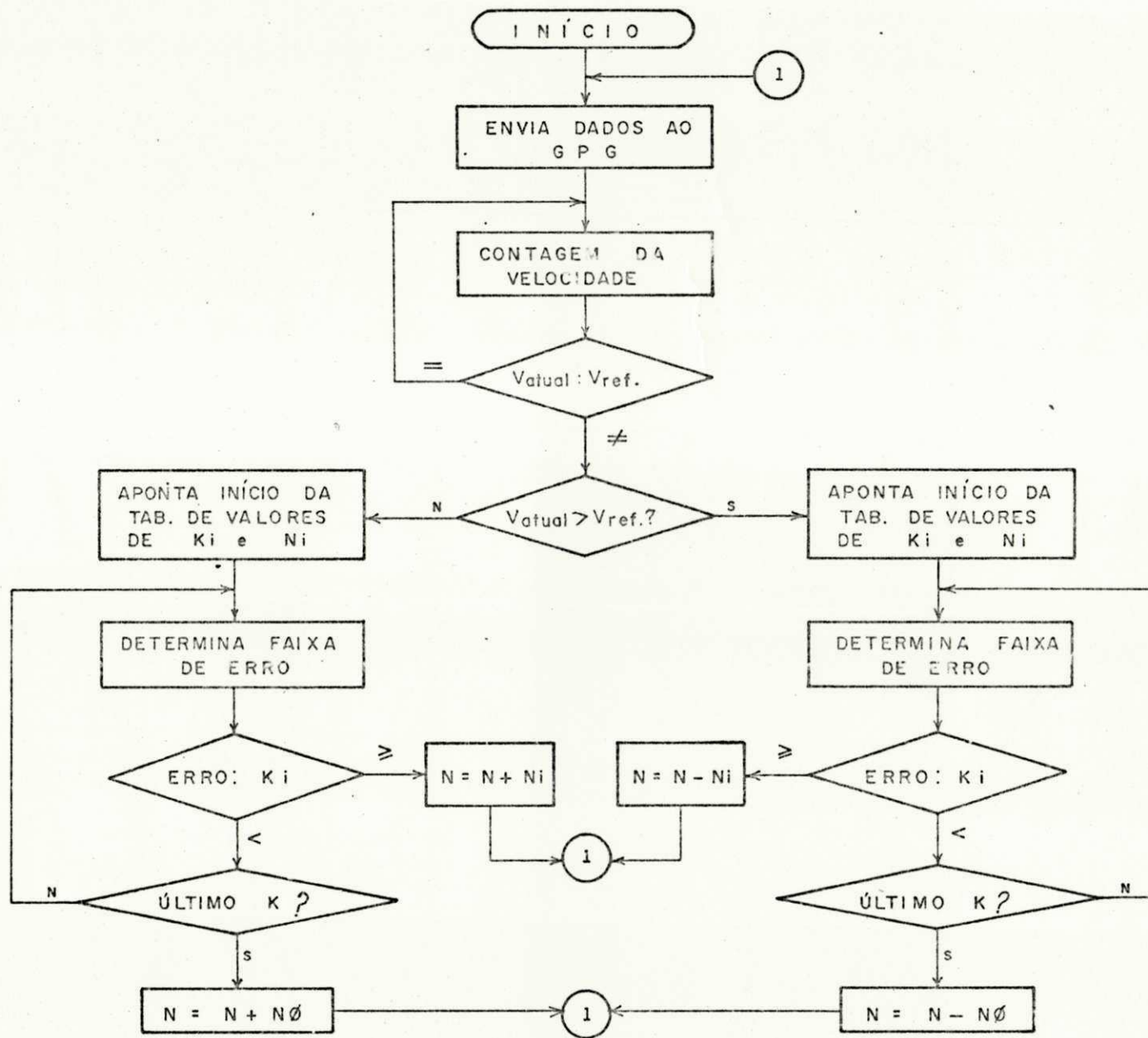


Fig. 3.4 - Fluxograma das operações em REGIME.

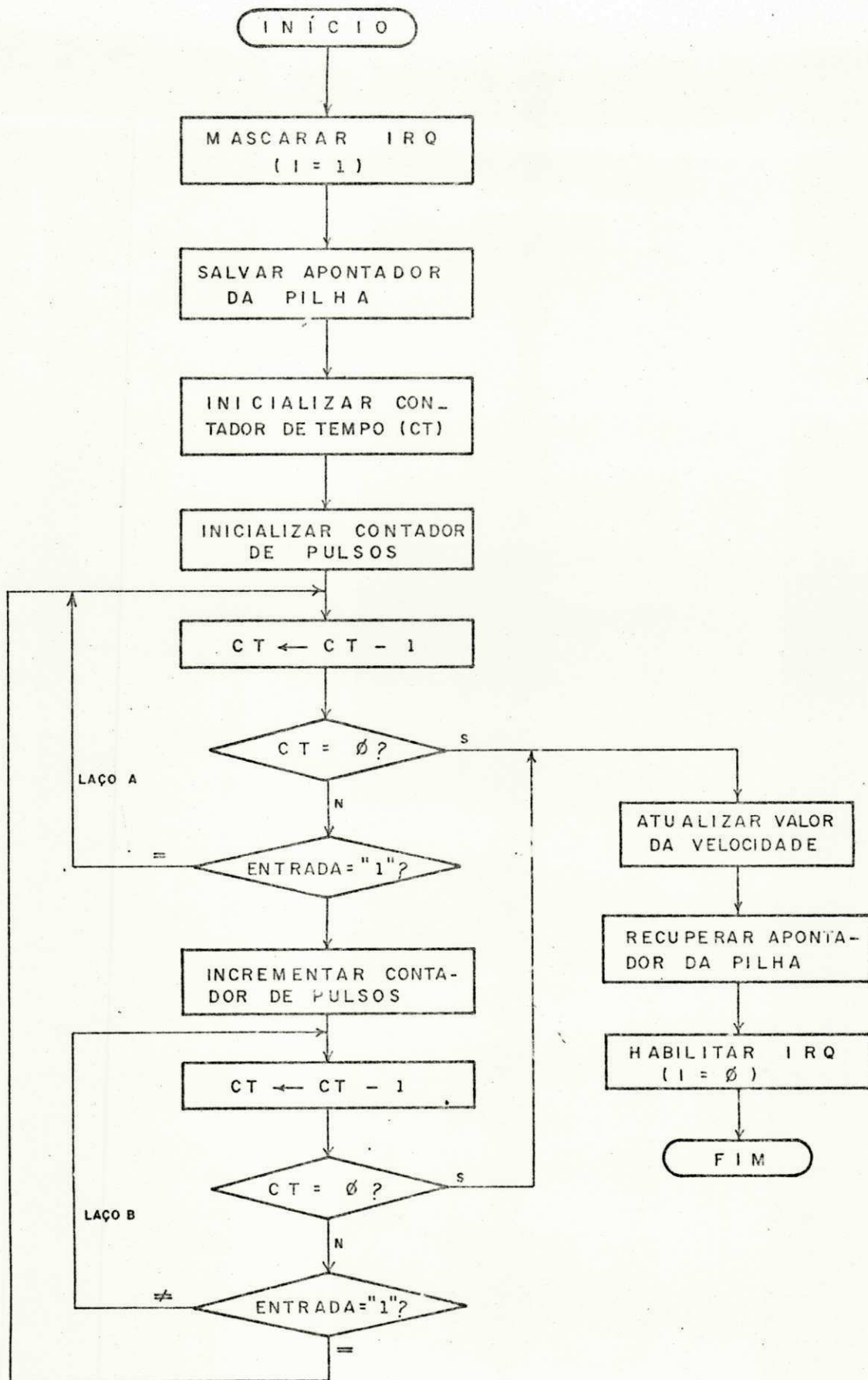


Fig. 3.5 - Fluxograma da subrotina de contagem.

que corresponde ao valor atual de N na entrada do gerador de pulsos, de acordo com a magnitude e o sinal do erro. Na Fig. 3.6, N_i ($i = 0, 1, 2, \dots$) são os valores de correção que devem ser somados ou subtraídos do valor corrente do número N . Os valores de K determinam as faixas de quantização do erro. O algoritmo de controle corrige a velocidade a intervalos de $200 \text{ mseg.} + T_{\text{cor.}}$, onde $T_{\text{cor.}}$ é o tempo de processamento requerido para calcular o erro e determinar qual o valor de correção adequado. Este tempo varia com o número de níveis de quantização especificados. Para um bom desempenho, $T_{\text{cor.}}$ deve ser muito menor que a constante de tempo mais significativa do processo, no caso a constante de tempo mecânica do motor, que deve ser também maior que o intervalo de contagem. A estratégia de controle mostrada na Fig. 3.6 limita a correção na tensão de armadura a um valor limite quando o erro ultrapassa determinado ponto, que corresponde ao módulo do maior valor de K especificado pelo usuário. Esta característica garante a estabilidade do sistema de controle. Os valores limites de ganho (inclinação das linhas tracejadas na Fig. 3.6) devem ser escolhidos de modo a evitar oscilações (ciclo limite) muito grandes em torno do estado de erro nulo. Sob este aspecto, deve-se também levar em consideração o número de níveis de quantização escolhido, pois dele depende o valor da diferença entre correções consecutivas.

As tarefas descritas a seguir são executadas em situações especiais do sistema de controle. Elas podem ser acionadas durante a partida bem como durante a operação em regime. A sequência em que essas tarefas estão descritas corresponde a ordem de prioridade dada as mesmas. Na Fig. 3.7 é mostrado o fluxograma da

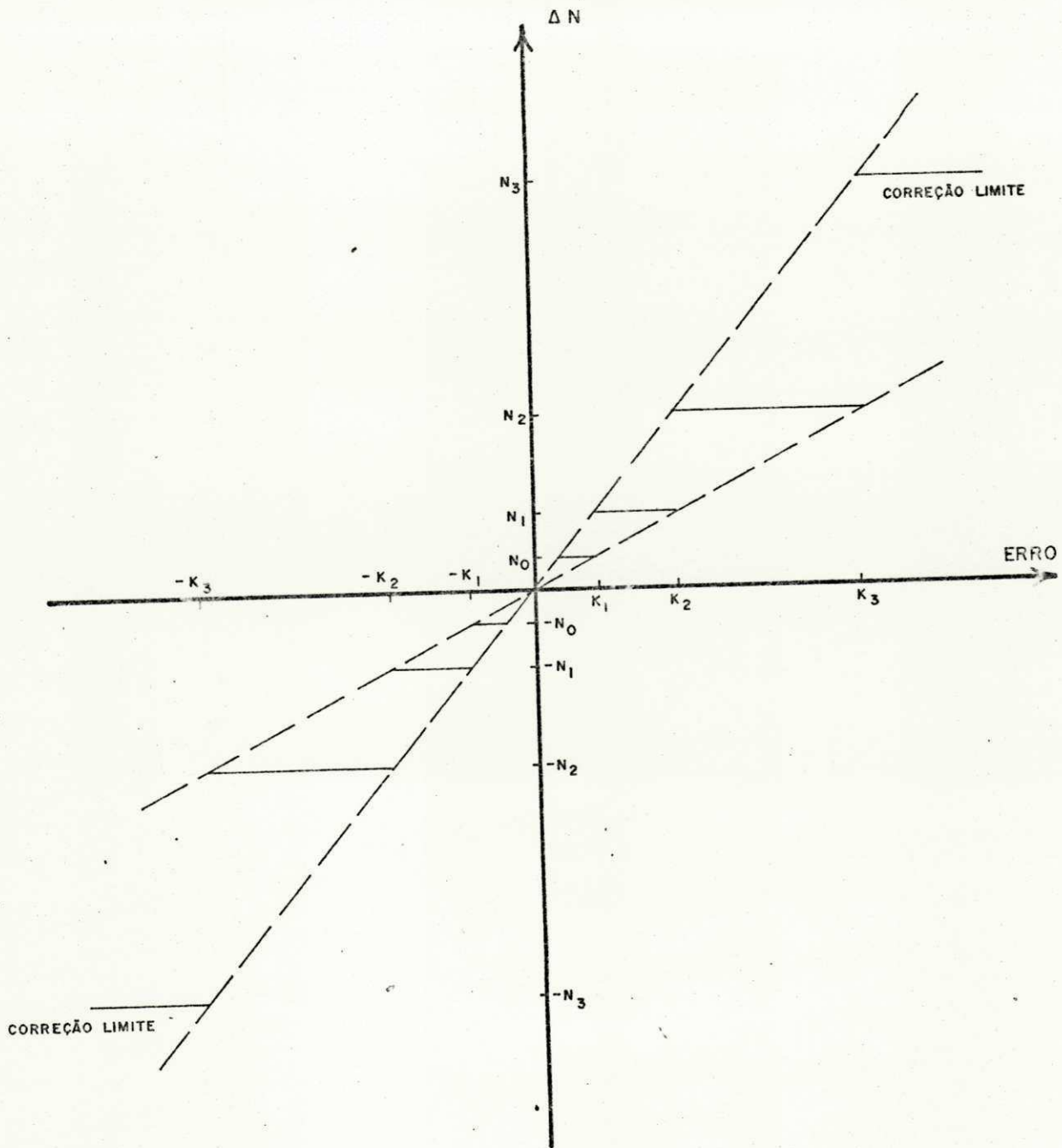


Fig. 3.6 - Característica do controlador de velocidade .

rotina de atendimento a essas tarefas.

3.2.4 - VÍDEO/TECLADO

A execução desta tarefa inclui duas operações distintas: transmitir ao operador, através da tela do terminal, mensagens que descrevem a situação especial em que se encontra o processo e receber, via teclado, comandos que permitam obter informações ou alterar as condições correntes de controle. A mútua exclusão entre as operações de transmissão/recepção de dados é efetuada utilizando-se os próprios recursos da ACIA. Desde a partida do motor, o sistema está normalmente preparado para requisitar interrupção na entrada de caracteres. Quando é necessário enviar uma mensagem, os pedidos de interrupção via teclado são desabilitados (através de um registrador de controle disponível) e a ACIA é preparada para gerar interrupções na saída de caracteres. Ao término da mensagem ela é reprogramada para voltar à situação anterior. Como no presente sistema, a utilização do terminal não é uma tarefa crítica no tempo (tarefa cuja frequência de acionamento e tempo de execução são conhecidos, ver Referência [25]) este esquema funciona satisfatoriamente. As mensagens e comandos disponíveis ao operador estão descritos no Apêndice B.

3.2.5 - SOBRECORRENTE

Um comando gerado por um circuito externo à malha de controle da velocidade (ver seção 2.6) aciona, por interrupção através da PIA, a tarefa SOBRECORRENTE. Esta tarefa tem a função de

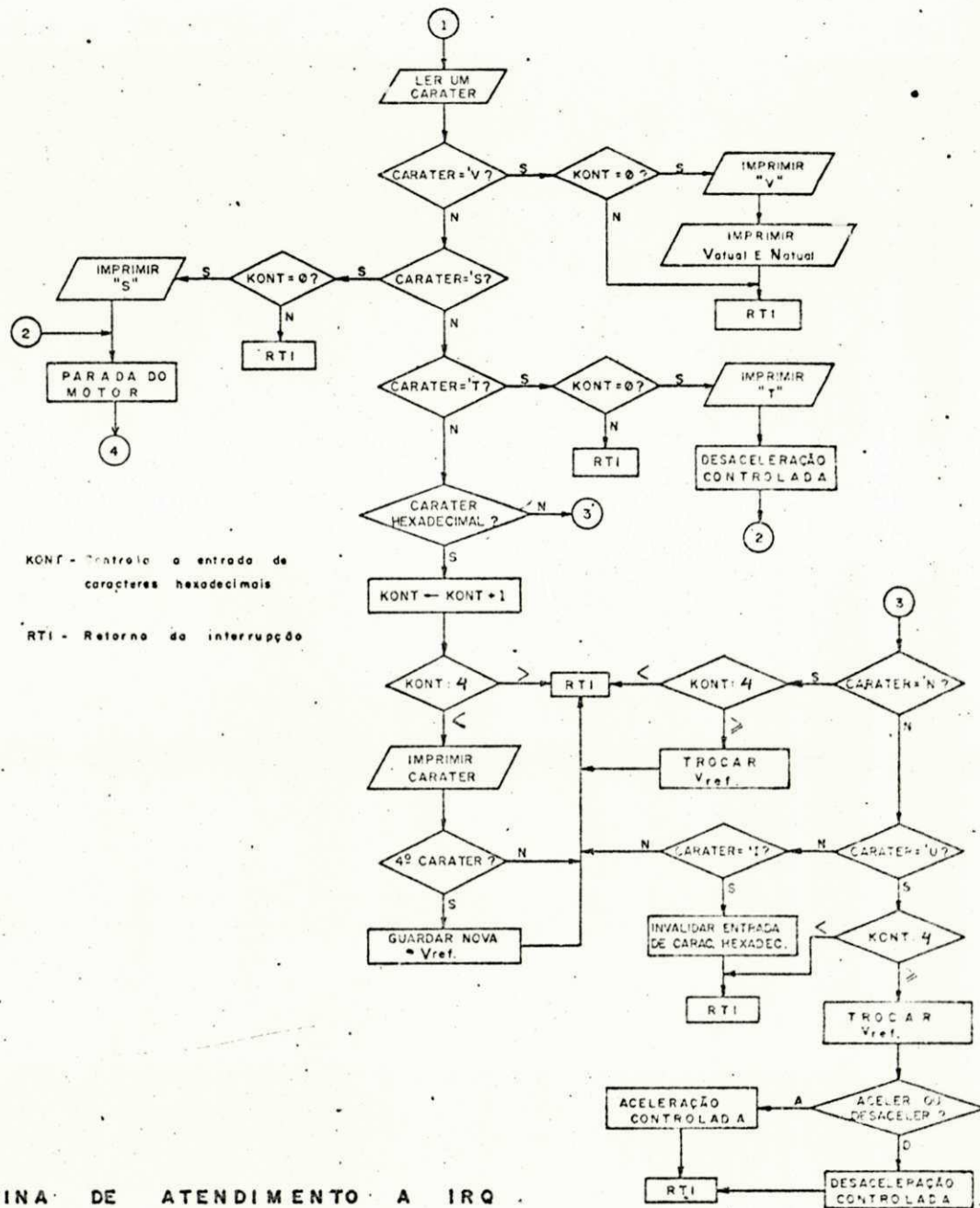
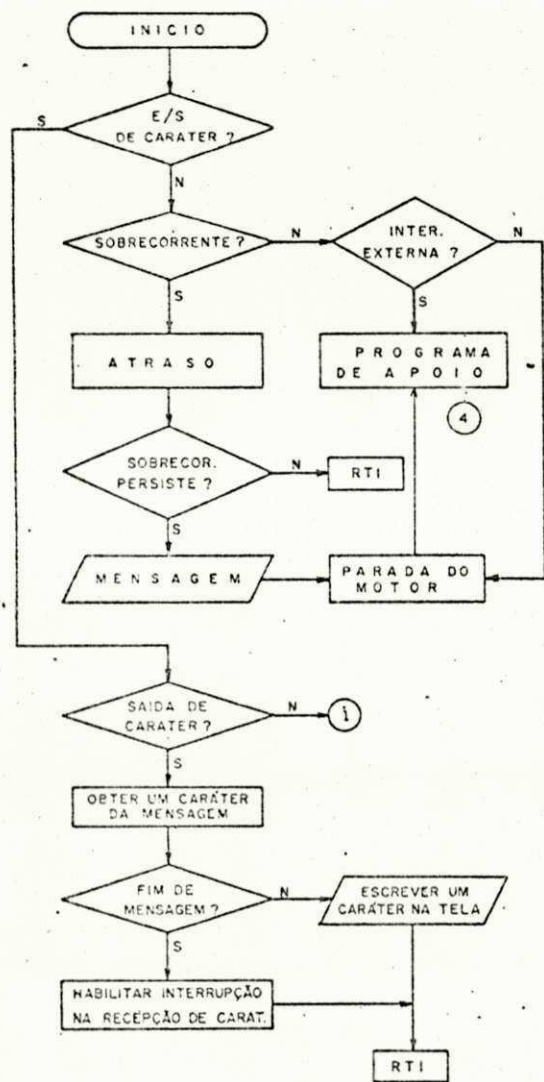


FIG. 3.7 - FLUXOGRAMA DA ROTINA DE ATENDIMENTO A IRQ

acionar os circuitos de proteção do sistema (chaveador e motor) quando a corrente de armadura ultrapassa determinado valor. A sequência de operações para efetuar essa tarefa é a seguinte:

1. Uma transição (↓) na saída do circuito de detecção de corrente (Fig. 2.14) requisita uma interrupção (através de CB2, na Fig. 2.2).
2. Na rotina de atendimento à esta interrupção é verificado, durante determinado intervalo de tempo, o nível lógico de um dos "bits" do registrador de dados da PIA (PA3, na Fig. 2.2). Se ao final desse intervalo o nível lógico permanece baixo, o esquema de parada do motor é acionado. Caso contrário o sistema de controle volta ao processamento normal a partir do ponto em que estava quando a tarefa foi acionada, significando que a situação de sobrecorrente não se estabeleceu de modo a provocar danos no motor ou no circuito de potência.

A finalidade deste procedimento é evitar paradas desnecessárias do motor durante a partida ou quando ocorre variação brusca de carga. Ambos os casos podem provocar picos transitórios de corrente de armadura. O esquema de parada do motor pode ser acionado de 4 maneiras e é abordado no final deste capítulo.

3.2.6 - REINICIALIZAÇÃO

O operador pode por meio de uma chave externa, interrom -

per a operação do sistema de controle. Quando isto ocorre a tarefa REINICIALIZAÇÃO é acionada. As operações efetuadas por esta tarefa, identicamente às inicializações via teclado da tarefa acionada por RESET, se processam "off-line". A diferença principal é que aqui as operações são executadas com o motor em funcionamento. O operador pode então, alterar os valores dos parâmetros de controle (ver Apêndice C) e acionar novamente a tarefa REGIME. Deste modo o sistema de controle passa a funcionar com outra configuração (novos parâmetros) sem a necessidade de uma nova partida do motor. A parada do motor pode ser acionada durante a execução da tarefa em questão.

3.2.7 - AJUSTES

O acionamento desta tarefa não tem a característica aleatória daquelas acionadas por IRQ. Ela é ativada por interrupção de "software" (SWI) e portanto, é executada em pontos determinados da estrutura dos programas. A execução integral da tarefa AJUSTES é garantida pelas características inerentes do microprocessador. Os pedidos de interrupção efetuados em seguida a ocorrência de uma SWI são mascarados automaticamente e somente são atendidos após a interrupção de "software" ter sido servida. Este recurso simplifica o controle para garantir a mútua exclusão entre a tarefa AJUSTES e aquelas acionadas por IRQ.

A função da presente tarefa é efetuar ajustes no número N enviado ao gerador de pulsos quando certas condições acarretam ao sistema (GPG e motor) um funcionamento fora de suas especificações.

Para certos valores de velocidade de referência e condições de carga, o ciclo de trabalho do trem de pulsos de potência aplicado à armadura do motor pode ultrapassar ou ser menor que os valores máximo e mínimo estabelecidos pelo usuário (ver Apêndice C). Quando uma destas situações ocorre, o operador é avisado através de uma mensagem na tela do terminal. Em seguida a tarefa AJUSTES é acionada e a tensão de armadura é automaticamente corrigida. O número de ajustes a serem efetuados é previamente especificado pelo usuário (ver Apêndice C). Se a situação for incontrolável, ou seja, as condições requeridas de velocidade e carga não puderem ser supridas pelo sistema após os ajustes, o esquema de parada do motor é acionado.

3.3.- CONSIDERAÇÕES FINAIS

Em sistemas de controle por microcomputador em tempo real, é, de um modo geral, necessário que cada mudança no parâmetro sob controle (no caso a velocidade) seja corrigida pelo sistema de controle antes da próxima mudança ser reconhecida. O "software" para este tipo de aplicação deve portanto, ter capacidade de usar os recursos do sistema para minimizar os atrasos nas respostas a cada mudança.

Todo o "software" do presente sistema (incluindo as rotinas de interrupção) está dividido em 26 blocos, com uma média de 21 instruções por bloco. Cada bloco tem no máximo um tempo de processamento da ordem de dezenas de μseg . Como as mudanças de velocidade são reconhecidas pelo processador a cada 200 μseg ., a respos-

ta a essas ocorrências é garantida.

Como citado anteriormente, o esquema de parada do motor pode ser acionado em 4 situações distintas, a saber:

- 1) Quando a condição de sobrecorrente de armadura persiste após certo número de ajustes (ver seção 3.5).
- 2) Quando certas condições de velocidade e carga não podem ser supridas pelo sistema de controle (ver seção 3.7).
- 3) Quando o operador aciona, via teclado, os comandos S ou T (ver Apêndice B).
- 4) Durante a execução da tarefa REINICIALIZAÇÃO. Neste caso, o operador pode, através de um comando de desvio (ver comando G, no Apêndice A) endereçar a rotina de parada do motor, a qual passa a ser executada. Durante esta operação nenhuma ação de controle é executada. O microcomputador está apenas esperando a entrada de dados ou comandos.

A parada do motor consiste no bloqueio dos tiristores do chaveador, ela é efetuada através do controle das linhas BTP (Bloqueia Tiristor Principal) e MR ("Master Reset"). BTP quando em nível lógico "1" não permite o envio de pulsos de gatilhamento ao tiristor principal. A linha MR também é ativa em nível alto. O blo-

queio total dos pulsos obedece a sequência abaixo:

- 1) A linha BTP é levada ao nível lógico alto.
- 2) O microcomputador entra em um laço de espera de aproximadamente 10 mseg. Este tempo é suficiente para que ocorram dois pulsos consecutivos (alternados), um em cada par de tiristores auxiliares, garantindo a comutação forçada do tiristor principal.
- 3) A linha MR também é levada ao nível lógico alto, zerando os contadores do gerador de pulsos. Isto elimina totalmente a geração de pulsos.

Ao final, uma mensagem avisa ao operador que o bloqueio foi efetuado e o controle passa ao programa de apoio. O sistema pode então receber novos parâmetros de controle e outra partida ser iniciada, se for necessária. É importante salientar que embora o acionamento do RESET do microcomputador faça $BTP = MR = "1"$, este comando não deve ser acionado com o motor em funcionamento. Se isto ocorrer, os passos citados acima não são seguidos e a tensão de entrada fica diretamente aplicada ao motor através do tiristor principal. Neste caso a fonte de alimentação do chaveador tem de ser desligada.

CAPÍTULO 4

TESTES EXPERIMENTAIS E CONCLUSÕES

A experiência adquirida com um sistema de controle por microcomputador anteriormente implementado e testado [27], acarretou a elaboração do presente trabalho. O referido sistema embora permitisse uma comunicação com o operador através do terminal de vídeo/teclado, tinha suas limitações. O microcomputador utilizado foi o MEK-D1 (Motorola Evaluation Kit) [30], o qual utiliza a PIA como circuito de interface de comunicação com o terminal. A PIA, como utilizada no MEK-D1, não é apropriada para comunicação por meio de interrupção na entrada e saída de caracteres. Desse modo, toda a comunicação entre o operador e o microcomputador era feita "off-line". O sistema de "software" de desenvolvimento estava, conseqüentemente, limitado a atender apenas interrupções externas através de chaves ou quando ocorria sobrecorrente de armadura. Além disso, o MEK-D1 é um microcomputador de arquitetura pouco flexível, dificultando a expansão de memória ou o acréscimo de novos blocos de interface de E/S.

Devido as limitações do MEK-D1 construiu-se um microcomputador (ver seção 2.2) com maiores recursos de "hardware" para permitir principalmente, uma comunicação "on-line" entre o operador e o sistema de controle. Para isto, foi desenvolvido um sistema de "software" baseado em tarefas acionadas por interrupção (ver Capítulo 3). No sistema implementado, como as tarefas não são críticas no tempo, um esquema de acionamento das mesmas com níveis de prioridade fixos ("polling") funciona satisfatoriamente. Para sistemas de controle de processos baseados em microcomputador tipo MATER (Multimicrocomputador para Aplicações em Tempo Real)|27,28|, estudou-se algoritmos de KERNEL|26|e SCHEDULER|25|onde são estabelecidas filas de classes de prioridade, descritores de tarefas, garantia de mútua exclusão no acesso a variáveis compartilhadas pelas tarefas, etc.

Além das características interativas já mencionadas, é importante salientar que o sistema pode ser empregado para determinar algumas constantes do motor experimentalmente. Para isto, usa-se a própria subrotina de contagem (ver Fig. 3.5) com pequenas modificações, como mostrado na Fig. 4.1. O método consiste em enviar ao gerador de pulsos um número N correspondente ao degrau de voltagem a ser aplicado ao motor. Em seguida os valores de velocidade atual, obtidos após cada intervalo de contagem, são armazenados em locações consecutivas de memória. Depois de determinado número de valores amostrados, o controle do microcomputador é desviado automaticamente ao programa de apoio. O operador pode então, obter na tela do terminal os valores de velocidade anteriormente armazenados e a partir deles, sabendo que o tempo decorrido entre cada amostragem é de 200 msec, construir um gráfico da variação de velo-

cidade x tempo, para determinar o tempo de resposta do motor, Este método também pode ser empregado para determinar o tempo de resposta do sistema de controle para uma variação na velocidade de referência ou na carga. Alguns testes foram realizados aplicando-se variações de grau de voltagem ao motor. Os resultados estão mostrados no Apêndice E. Embora as condições de teste não fossem satisfatórias (principalmente devido ao atraso na resposta do multiplicador de frequência baseado em PLL do codificador de velocidade), os resultados indicam que o conjugado sob teste (motor + alternador) tem uma constante de tempo maior que 200 mseg., condição necessária para um bom desempenho do sistema de controle. A Fig. 4.1 mostra o fluxograma da subrotina de contagem adaptada para que se possa obter os resultados mostrados no Apêndice E.

A subrotina de contagem da velocidade também pode ser adaptada para que o sistema possa funcionar com intervalos de contagem (T) menores que 200 mseg. O método consiste em multiplicar o número de pulsos contados (decodificador de velocidade) de acordo com o intervalo estabelecido. O fator de multiplicação é igual a $2^{NDES L}$, onde NDES L estabelece quantos deslocamentos para a esquerda devem ser efetuados no número binário representativo da quantidade de pulsos contados (NPT) após cada intervalo. Na Tabela I são mostrados alguns resultados utilizando o método proposto. Estes resultados demonstram que o método pode ser usado satisfatoriamente em sistemas de controle nos quais um erro de até $2^{NDES L}$ rpm, na medição da velocidade, seja aceitável. A Fig. 4.2 mostra o fluxograma salientando as modificações efetuadas na subrotina de contagem para que a mesma possa operar com $T \leq 200$ mseg.

Durante a medição da velocidade, como mostrado na Fig. 3.5,

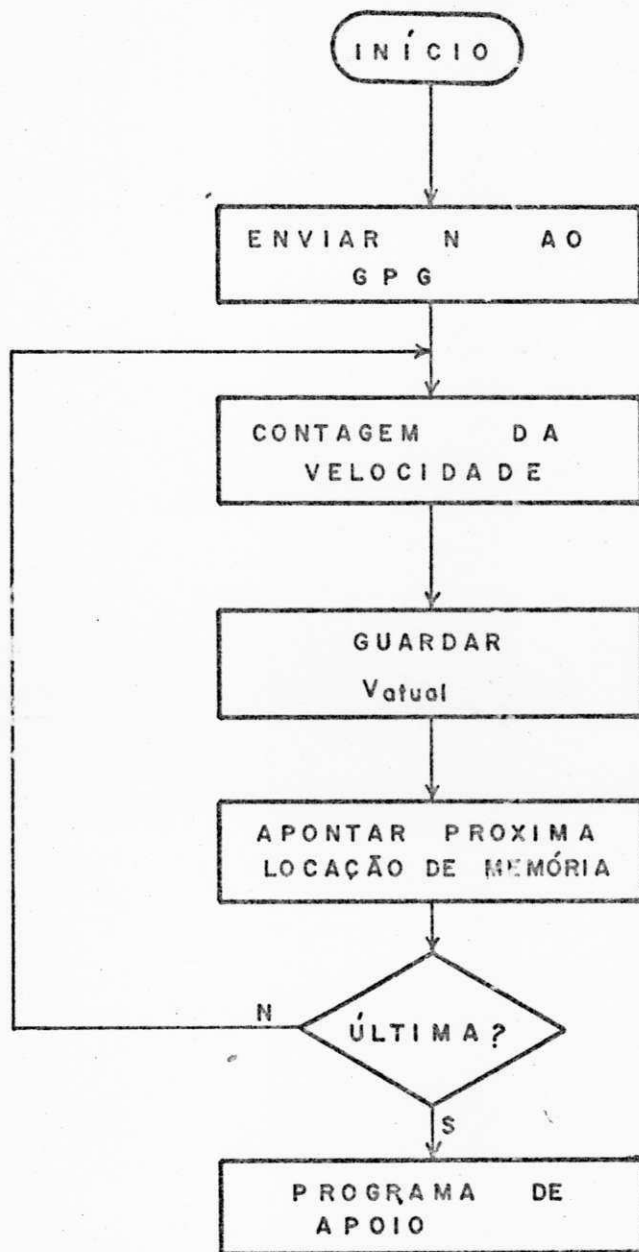


Fig. 4.1 - Fluxograma da subrotina de contagem modificada para determinar o tempo de resposta do motor.

T (mseg.)	NDESL	NPT	V (rpm)
200	0	425	425
		425	425
		425	425
		426	426
		425	425
100	1	213	426
		213	426
		212	424
		213	426
		213	426
50	2	107	428
		108	432
		107	428
		107	428
		107	428
		107	428
25	3	54	432
		54	432
		54	432
		53	424
		54	432
12,5	4	27	432
		27	432
		27	432
		27	432
		27	432

Tabela I - Resultados dos testes
com a subrotina de
contagem para $T \leq 200$ mseg.

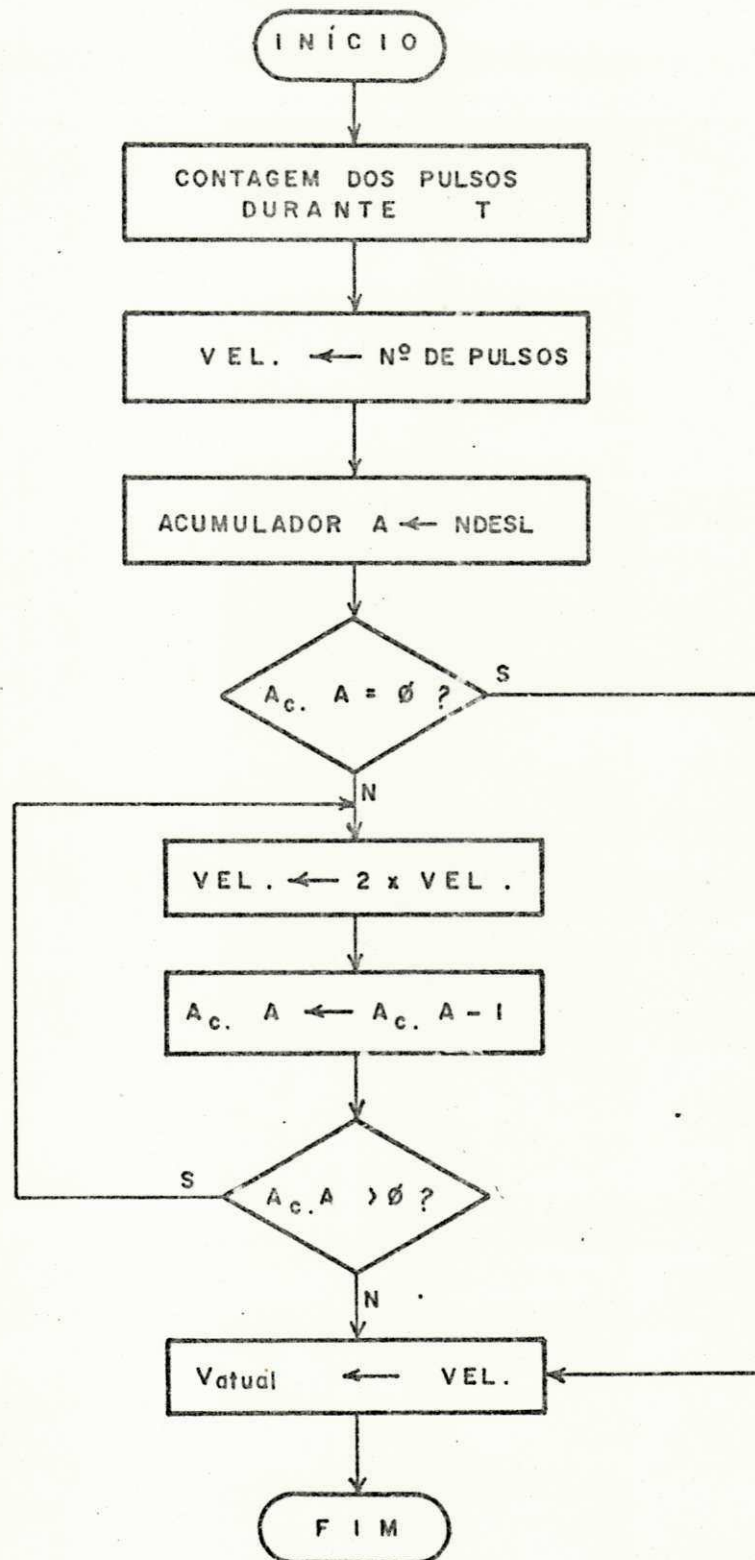


Fig. 4.2 - Fluxograma da subrotina de contagem para obtenção de intervalos de amostragem $T \leq 200$ msec.

o microprocessador passa a maior parte do tempo apenas verificando a ocorrência de transições dos pulsos gerados no codificador de velocidade. Para aumentar o fator de utilização do microprocessador, testou-se um método de medição da velocidade, no qual a contagem dos pulsos é feita por interrupções, como é descrito a seguir:

- 1) Os pulsos da saída do codificador de velocidade são aplicados à entrada NMI do microprocessador e cada transição negativa nesses pulsos gera uma interrupção, cuja rotina de atendimento apenas incrementa o conteúdo de uma locação de memória aqui chamada CONTPUL.
- 2) Em um circuito externo é gerada uma base de tempo para definir o intervalo de contagem. Ao final de cada intervalo uma interrupção é requisitada (IRQ) e a rotina de atendimento tem a função de guardar o valor da velocidade atual (número de pulsos contados) e zerar o conteúdo de CONTPUL para nova contagem.

As Figs. 4.3 e 4.4 mostram o fluxograma das rotinas de atendimento citadas acima. O diagrama de tempo das operações supracitadas está ilustrado na Fig. 4.5. A diferença entre o período T' e o tempo de execução das rotinas de NMI e IRQ - T_E , garante a precisão de ± 1 pulso de contagem dos pulsos a cada intervalo. Por exemplo, quando a velocidade é de 1800 rpm (velocidade nominal do motor utilizado nos testes), tem-se que a relação $T'/T_E \approx 4$. T_E é da ordem de 30 μ seg. incluindo o tempo necessário para mover os registradores do microprocessador para a pilha, quando ocorre uma interrupção.

A principal vantagem do método de medição da velocidade por interrupções em relação ao apresentado no Capítulo 3 (subroti-

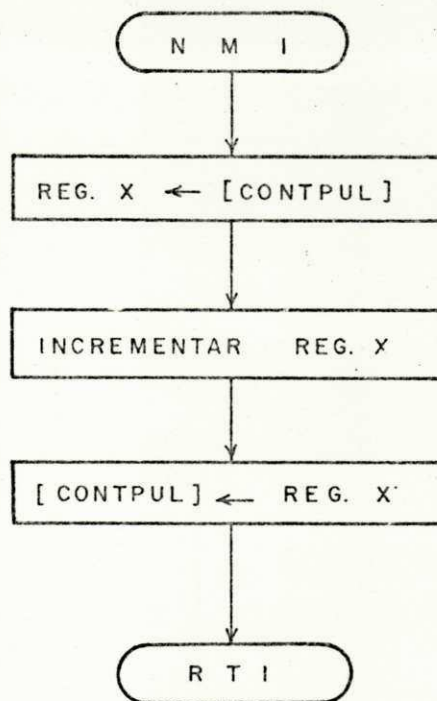


Fig. 4.3 - Fluxograma da rotina de atendimento a N M I.

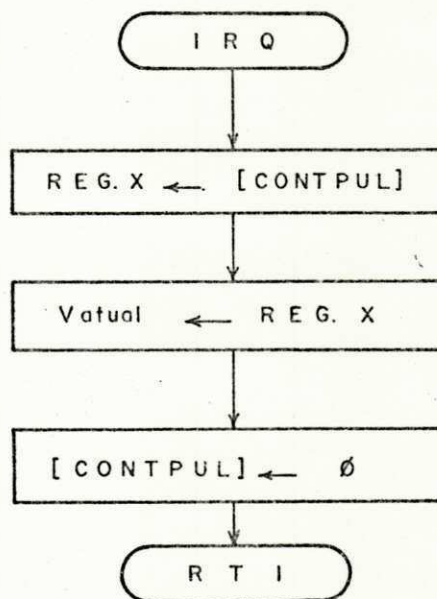


Fig. 4.4 - Fluxograma da rotina de atendimento a I R Q.

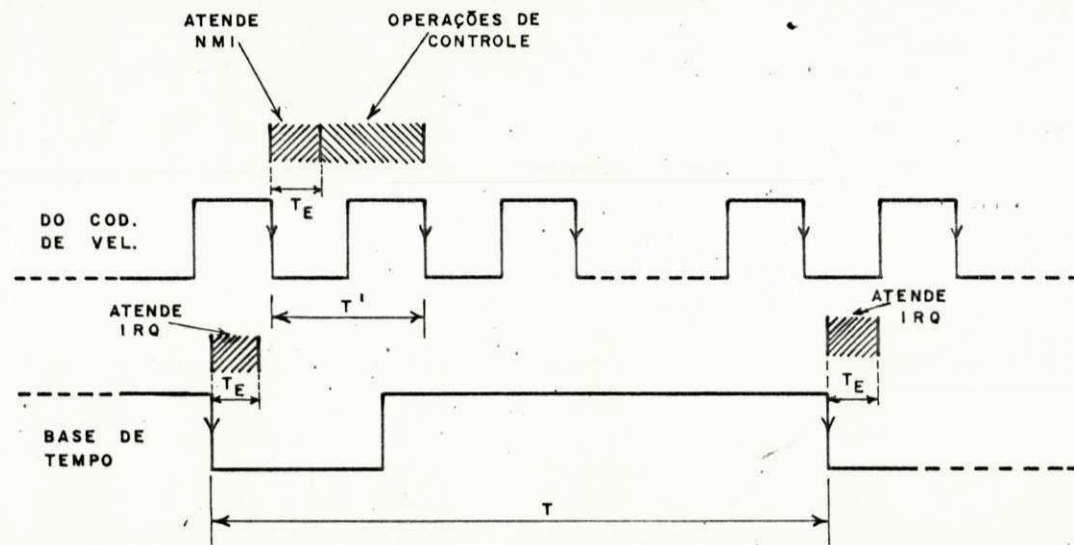


Fig. 4.5 - Diagrama de tempo das operações durante a contagem por interrupções.

na de contagem), é que após a contagem de cada pulso, o microcomputador volta a executar operações de controle. Na Tabela II são mostrados os resultados dos testes utilizando ambos os métodos, para vários valores de frequências dos pulsos (f'). A relação entre f' e a velocidade do motor é dado por:

$$f' = V(\text{rpm}) \times \frac{300 \text{ pulsos/rotação}}{60 \text{ seg./min.}}$$

$$f' = V \times 5 \text{ pulsos/seg.}$$

Para a utilização do método de medição da velocidade por interrupções no presente sistema, foi projetado e montado um circuito como mostrado na Fig. 4.6. Ele é constituído de um oscilador astável, cujo período de oscilação é o intervalo de contagem e de portas lógicas usadas para inibir ou habilitar as operações de contagem, o que é necessário nos seguintes casos:

- 1) Quando o microcomputador é ligado. Neste caso, após o acionamento da chave RESET, PA0 (ver Fig. 4.6) passa automaticamente ao nível lógico "1" e a contagem dos pulsos só é iniciada quando através de programa PA0 é levado a nível "0".
- 2) Durante a execução de tarefas tais como: AJUSTES, REINICIALIZAÇÃO, etc. Neste caso, o bloqueio da interrupção NMI é necessário, pois sendo a mesma não-mascarável, a contagem dos pulsos prosseguirá sem que seu valor seja atualizado e zerado ao final do intervalo de contagem. Os pedidos de interrupção IRQ estão mascarados até que as referidas tarefas sejam completadas.

Nos testes realizados no laboratório (resultados na Tabe-

FREQ. PULSOS (f') (KHz)	CONTAGEM P/ INTERRUPÇÃO (rpm)		SUBROTINA DE CONTAGEM (rpm)		VEL. (V) (rpm)
	Mín.	Máx.	Mín.	Máx.	
0,3	58	59	59	60	60
0,4	79	80	80		80
1,0	198	199	198		200
1,5	297	298	298		300
2,0	398		397	398	400
2,5	497	498	497		500
3,0	598	599	600		600
3,5	699	700	700	701	700
4,0	800		800		800
4,5	900	901	900		900
5,0	999	1.000	1.000		1.000

Tabela II - Resultados Comparativos entre os métodos de contagem testados.

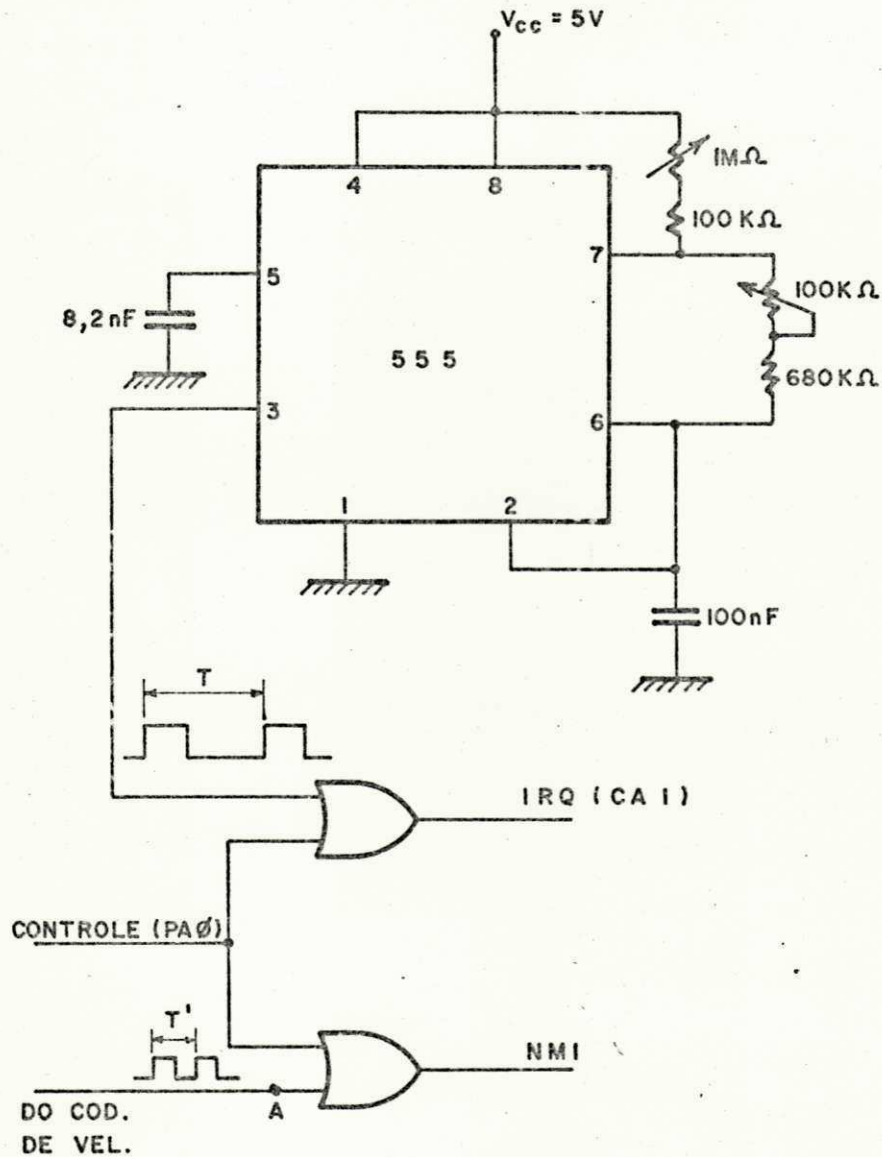


Fig. 4.6 - Base de tempo + controle das interrupções para a contagem da velocidade.

la II), o codificador de velocidade foi simulado (substituído) por um gerador de pulsos (ver Fig. 4.6). O sistema atual utiliza a subrotina de contagem para medição da velocidade, por isso, a entrada CONTROLE (PAØ) está ligada em V_{cc} (5V).

Tendo em vista ainda a proposição de suprir o presente sistema com recursos para que a partir das especificações do motor, tipo de aplicações, etc, se possa desenvolver esquemas de controle adequados para cada caso, foram incorporados ao sistema dois blocos de interface, aqui chamados ACIAPDP e PIA1. Através da ACIAPDP, o microcomputador do sistema pode ser ligado a um minicomputador PDP 11/34. Para efetuar esta ligação, foi desenvolvido um sistema interativo [29] que permite ao operador, através de comandos especiais, utilizar os recursos do PDP11/34. A PIA1 além de incluir 4 entradas adicionais para pedidos de interrupção, que podem ser usadas para acionar outros tipos de proteção, pode ser programada para fornecer ao sistema o valor da velocidade de referência gerada externamente (e.g. outro sistema de controle) ou pode, acoplada a um conversor A/D, prover o microcomputador de dados referentes à corrente de armadura e assim incorporar ao sistema a malha de realimentação de corrente.

Na montagem inicial do microcomputador como mostrado na Fig. 2.5, usou-se dois "chips" de memória EPROM MCM2708C de 1Kbyte cada. Devido às dificuldades de gravação destas memórias (manutenção do Exorciser), elas foram substituídas por uma MCM2716C de 2Kytes, que dispõe de um circuito de gravação já montado no laboratório. As modificações efetuadas no circuito do microcomputador são mostradas na Fig. 4.7.

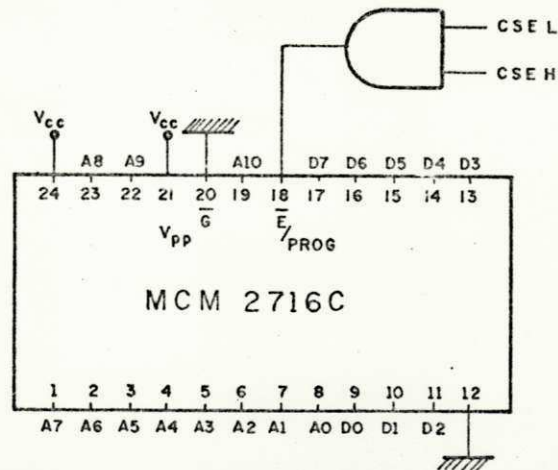


FIG. 4.7 - Modificações efetuadas no circuito do microcomputador

Os recursos disponíveis de "hardware" e "software" do presente projeto foram apresentados. Eles permitem o desenvolvimento de sistemas de controle de velocidade específicos de acordo com o tipo de aplicação definida. Adicionalmente, foram abordados métodos para determinar constantes do motor e do próprio sistema de controle. Nos testes usou-se um par motor-gerador acoplado a uma carga de lâmpadas. O esquema de aceleração controlada funcionou satisfatoriamente para várias condições de carga e velocidade final a ser atingida. A parada do motor, acionada por interrupção, devido à sobrecorrente também foi testada aumentando-se a carga, bem como o esquema de ajustes automáticos na voltagem média de armadura. Durante as operações em regime, a estratégia de controle não-linear, juntamente com a possibilidade de mudança dos parâmetros de controle, através de interrupção externa, permite ao operador

usar o método de tentativa e erro para dimensionar esses parâmetros (principalmente a faixa de variação do ganho do controlador) a fim de obter a precisão e tempo de resposta adequados. Como expansões do sistema, sugere-se a implementação de controladores tipo PI, PID, etc., a incorporação dos algoritmos de KERNEL e SCHEDULER para facilitar a adição de novas tarefas ao sistema, a ligação com o PDD11/34 e através de uma lógica simples inibir a interrupção RESET durante a operação do sistema de controle, pois como visto na seção 3.8, não é permitido o acionamento dessa interrupção com o motor em funcionamento.

APÊNDICE A

PROGRAMA DE APOIO MÍNIMO - PAM

1. APRESENTAÇÃO

Este programa é oriundo de um trabalho mais abrangente [23]. Ele possui os recursos mínimos para que o usuário possa depurar e testar seus próprios programas. Estes recursos ficam disponíveis logo que a chave RESET é acionada, quer seja quando o sistema (microcomputador mais terminal de vídeo/teclado) é ligado ou quando o processador perde o controle do programa que está executando. Possui também meios para que se possa atender aos três tipos de interrupção do MC6800 (IRQ, SWI e NMI). Ocupa 267 bytes de memória EPROM (no presente caso a partir da locação F800), além das posições em RAM para armazenamento de variáveis e definição de pilha ("stack"). A relocação deste programa é simples, visto que, todos os desvios do mesmo são feitos através de endereçamento relativo ("branches").

2. MODO DE OPERAÇÃO

O modo de operação do PAM é o seguinte:

- 1º) O sistema é ligado.
- 2º) A chave RESET é acionada.
- 3º) O microcomputador envia ao terminal os comandos LF ("Line Feed") e CR ("Carrige Return") seguidos do caráter: ":" (dois pontos).
- 4º) O microcomputador está, então, esperando que o usuário entre (via teclado) com um dos caracteres pré-estabelecidos (M, R ou G) para realização de uma função específica.
- 5º) Se qualquer outro caráter é introduzido, o sistema volta ao 3º) passo.

3. FUNÇÕES DO PAM

As funções realizadas pelo PAM são:

- a) Troca/verifica conteúdo de locações de memória - Função M.

Exemplo de operação da função M:

```
:M 010A
010A 39 BD
010B 4C F8
010C 6E SP
010D 78 .
:
```

OBS.: Os caracteres sublinhados representam aqueles introduzidos pelo usuário. Endereços e dados na base hexadecimal.

Neste exemplo, os conteúdos das locações 010A e 010B fo-

ram trocados de 39 e 4C para BD e F8 respectivamente.

A entrada do caráter SP (espaço) mostra o conteúdo de locações consecutivas de memória sem alterá-los. E a entrada de um caráter não hexadecimal e diferente de SP faz o controle voltar ao PAM. Quando uma locação de memória apenas de leitura (no caso EPROM) é acessada e um novo conteúdo é introduzido, o sistema responde com um ponto de interrogação na tela do terminal e o controle retorna ao PAM, significando que o conteúdo daquela locação não pode ser trocado. Segue um exemplo.

```
:M FFFD
FFFDF8 A0?
:
```

b) Mostra conteúdo dos registradores da pilha - Função R

Esta função mostra na tela do terminal o conteúdo dos registradores da pilha na seguinte ordem: registrador de códigos de condição (CCR), acumulador B(B), acumulador A(A), registrador de indexação (X), contador de programa (PC) e apontador da pilha (SP). O exemplo abaixo mostra o modo de operação desta função.

```
:R
D1 02 1B 0103 050A 0742
:
```

c) Executa o programa do usuário - Função G

O usuário deve entrar com o caráter G seguido de quatro caracteres hexadecimais correspondentes ao endereço da primeira instrução de seu programa, como no exemplo seguinte:

:G 010A

Neste exemplo, o microcomputador passa a executar o programa gravado a partir da locação 010A. O controle volta ao PAM nos seguintes casos:

- 1) Se a chave RESET for acionada.
- 2) Se o programa do usuário desviar para um dos seguintes pontos no PAM: RESET (loc. F8B0). CONTRL (loc. F8C1) ou CONTR1 (loc. F8C6).
- 3) Ou se uma instrução SWI for executada. Este caso será visto com mais detalhes adiante.

4. MANIPULAÇÃO DAS INTERRUPÇÕES

Inicialmente serão feitos alguns comentários gerais sobre interrupção, antes de citar como ela é manipulada no PAM.

O MC6800 possui (além do RESET) 3 tipos de interrupções : duas requisitadas através de "hardware" - IRQ ("Interrupt Request") e NMI ("Nonmaskable Interrupt"); e uma requisitada através de uma instrução SWI ("Software Interrupt"). Uma interrupção "hardware" ocorre quando um nível lógico zero é aplicado na entrada \overline{IRQ} ou na

entrada $\overline{\text{NMI}}$ do MC6800. A diferença entre estes dois casos é a seguinte: Para que uma IRQ seja atendida é necessário que a máscara de interrupção I (bit 4 do CCR) seja zero. O programador pode anular o efeito de qualquer interrupção via $\overline{\text{IRQ}}$ fazendo I = 1 através de instruções apropriadas. A NMI, entretanto, sempre aciona a sequência de atendimento à interrupção do MC6800, como segue:

- 1) A instrução que estava sendo executada quando ocorreu a interrupção é completada.
- 2) Os registradores do programa interrompido são guardados na pilha na seguinte ordem: PC, X, A, B e CCR.
- 3) A máscara de interrupção passa ao estado lógico "1".
- 4) O endereço inicial da rotina de interrupção é colocado no PC.
- 5) A rotina de interrupção é iniciada.

As interrupções IRQ e NMI podem ocorrer em qualquer ponto do programa em execução, visto serem iniciadas a partir de um sinal externo. A SWI é totalmente controlada pelo programador. A execução de uma instrução SWI acarreta o desenvolvimento de uma sequência como a citada acima. Isto, como será visto adiante, é muito importante durante a depuração de programas. Segue a descrição de como as interrupções são manipuladas utilizando-se o PAM.

- IRQ

O PAM usa as locações 0700 e 0701 para guardar o endereço (2 bytes) da primeira instrução da rotina de atendimento a IRQ. Em 0700 é guardado o byte mais significativo e em 0701 é guardado o byte menos significativo. Por exemplo, se a rotina começa no endereço F910, o procedimento seguinte prepara o sistema para executar a mesma quando ocorrer uma $\overline{\text{IRQ}}$.

:M 0700

xx - não interessa

0700 xx F9

0701 xx 10

0702 xx .

:

- NMI

O procedimento é semelhante ao usado para IRQ. A diferença é que o endereço inicial da rotina de NMI deve ser guardado nas locações 0704 e 0705 respectivamente.

- SWI

Do mesmo modo que nos casos anteriores, o endereço inicial da rotina de SWI deve ser guardado nas locações 0702 e 0703 respectivamente. O PAM reinicializa automaticamente estas locações toda vez que a chave RESET é acionada. Essa inicialização prepara o sistema de modo que, os registradores da pilha sejam mostrados na

tela (como na função R), toda vez que uma instrução de SWI for executada, passando o controle de volta ao PAM. Isto permite que o programador estabeleça pontos de parada no seu programa, acompanhando cada passo de sua execução. Após esta fase de depuração, o programador deve atualizar as referidas locações (usando a função M) com o endereço de sua própria rotina de SWI.

5. SUBROTINAS UTILITÁRIAS DO PAM

O PAM possui algumas subrotinas que podem ser utilizadas pelo usuário em seus programas. O quadro abaixo indica o endereço de acesso a essas subrotinas, sua denominação e função.

ENDEREÇO	NOME	FUNÇÃO
F85D	SAICAR	Envia a tela do terminal o caráter (em ASCII) contido no acumulador A.
F86A	ENTCAR	Armazena (em ASCII) no acumulador A um caráter introduzido via teclado, o qual também aparece na tela do terminal.
F844	INHEX*	Aceita um caráter hexadecimal (0 - F) como entrada e o armazena no acumulador A na forma 0H, onde H representa o caráter hexadecimal introduzido.
F8A5	BYTE*	Aceita um byte de entrada (dois caracteres hexadecimais), que é armazenado no acumulador.

ENDEREÇO	NOME	FUNÇÃO
		dor A.
F897	BADDR*	Aceita 2 bytes de entrada, que são armazenados no registrador X.
F893	OUTS	Envia um espaço à tela do terminal.
F87D	SLFCR	Envia ao terminal os comandos LF ("Line Feed") e CR ("Carriage Return").
F886	OUT2H**	Envia à tela do terminal dois caracteres armazenados (em hexadecimal) em uma localização de memória apontada pelo registrador X.
F891	OUT2HS**	O mesmo que OUT2H mais espaço.
F88F	OUT4HS**	Envia à tela do terminal quatro caracteres armazenados (em hexadecimal) em duas localizações consecutivas de memória mais espaço. O registrador X deve conter o endereço da primeira localização.

* Se o caráter introduzido via teclado não for hexadecimal, o controle passa automaticamente ao PAM.

** Qualquer uma dessas subrotinas pode ser chamada consecutivamente dependendo do número de caracteres hexadecimais que se deseja mostrar na tela.

APÊNDICE B

COMANDOS E MENSAGENS

O presente sistema de controle se caracteriza por uma forte interação entre operador e o processo. Os comandos e mensagens disponíveis para esta comunicação são descritos a seguir. Os números que aparecem nos exemplos citados estão na forma hexadecimal.

1 - COMANDOS

- Comando V : Comunica ao operador o valor da velocidade atual seguido do número (N) lançado no gerador de pulsos.

Exemplo:

V 03A2 080F → representa a posição final do
→ cursor da tela do terminal.

- Comando S : Efetua a parada do motor bloqueando os pulsos de gatilhamento do chaveador. O controle retorna ao PAM.

Exemplo:

S
GATILHAMENTO BLOQUEADO
:

- Comando T : Efetua uma desaceleração controlada do motor, antes de bloquear os pulsos de gatilhamento do chaveador. O controle também retorna ao PAM.

Exemplo:

T
GATILHAMENTO BLOQUEADO
:

- Comando U : Efetua a troca da velocidade de referência. Para atingir a nova velocidade introduzida pelo operador, o motor passa por vários pontos de velocidades intermediárias, seguindo uma característica aproximadamente linear.

Exemplo:

040B U

VELOCIDADE FINAL ATINGIDA

→

Neste exemplo, o motor passa a operar a 1035 rpm (040B_{Hex}). A mensagem citada só é transmitida se o novo valor de referência for coerente com as especificações do sistema. A outra mensagem possível é descrita na seção seguinte.

- Comando N : Tem uma função semelhante ao comando anterior. A diferença é que a nova velocidade de referência é atingida através do algoritmo de compensação.

Exemplo:

040B N

→

Neste caso nenhuma mensagem é transmitida, visto que esta tarefa é executada no estado de regime e o operador dispõe do comando V para saber qual a velocidade corrente. A tarefa do comando anterior é executada em estado de interrupção.

- Comando I : Anula uma sequência de dados de entrada (1 até 4 caracteres), considerada inadequada.

quada para representar uma nova velocidade de referência.

Exemplo:

0B8I ou

→

AF91 I

→

Em ambos os casos, os valores introduzidos são anulados. O comando I também deve ser usado, toda vez que uma situação de máximo ou mínimo ciclo de trabalho for corrigida automaticamente pelo microcomputador.

Enquanto o operador está introduzindo um novo valor de velocidade de referência, qualquer caráter não hexadecimal teclado não afeta a operação e nem aparece na tela. O sistema só valida os 4 primeiros caracteres hexadecimais. Após estes, apenas os comandos U, N ou I são reconhecidos. Se nenhum caráter hexadecimal foi ainda introduzido, o sistema também ignora qualquer entrada diferente de V, S, T ou I.

2 - MENSAGENS

O sistema possui um total de 6 mensagens para comuni-

car ao operador as diferentes situações do processo. As mensagens estão gravadas (em ASCII) em memória EPROM, da locação FD60 a FDF4. Elas são as seguintes:

1) VELOCIDADE FINAL ATINGIDA

Esta mensagem é gerada toda vez que a tarefa requerida pelo comando U for completada. Ela também é usada para comunicar que o sistema passou do estado de partida ao regime.

2) MÁXIMO CICLO DE TRABALHO

Indica que o valor do número (N) a ser lançado no GPG é maior que o máximo especificado pelo operador (ver Apêndice C)

3) MÍNIMO CICLO DE TRABALHO

Indica que N é menor que o valor mínimo estabelecido.

4) GATILHAMENTO BLOQUEADO

Indica que o chaveador está desativado e consequentemente o motor está parado. Esta mensagem segue as requisições dos comandos S e T.

5) SOBRECORRENTE

Informa que a situação de sobrecorrente se estabeleceu. Esta mensagem é sempre seguida da anterior.

6) VARIAÇÃO INSUFICIENTE

Avisa que a diferença entre a nova velocidade de referência e a anterior é menor que 2^{NDIV+1} (ver Apêndice C). Esta mensagem só aparece se a troca de velocidade for requisitada pelo comando U.

As mensagens 2) e 3) são emitidas tantas vezes quanto o número de ajustes especificados, se a situação que as gerou for definitiva. Neste caso a mensagem 4) é transmitida em seguida.

APÊNDICE C

ORGANIZAÇÃO DOS PARÂMETROS DE CONTROLE

Após o acionamento da chave RESET o sistema fica sob controle do programa de apoio. Isto é indicado pela impressão do caráter ":" (dois pontos) na tela do terminal. O operador pode então inicializar os parâmetros do processo conforme a tabela a abaixo. Nesta tabela está especificado o endereço de memória onde deve ser introduzido o valor do parâmetro, o mnemônico do mesmo e a descrição de sua função. Todos os valores devem estar na base hexadecimal. Os parâmetros constituídos de dois bytes tem a terminação H para representar o byte mais significativo e L o menos significativo.

ENDER	PARÂM	DESCRIÇÃO
0000	VREFH	Velocidade de referência.
0001	VREFL	
0002	NMAXH ⁽¹⁾	Nº especificado para o máximo valor de N.
0003	NMAXL	
0004	NMINH ⁽¹⁾	Nº especificado para o mínimo valor de N.
0005	NMINL	

ENDER	PARAM	DESCRIÇÃO
0006 0007	VALINH ⁽²⁾ VALINL	Nº correspondente ao valor inicial de N.
0008 0009	CTH ⁽³⁾ CTL	Determina o tempo gasto na contagem da velocidade (200mseg.).
000A	NAJUST ⁽⁴⁾	Nº de ajustes.
000B	NDIV ⁽⁵⁾	Indica o nº de divisões de ΔV .
000C	NIVEIS ⁽⁶⁾	Nº de quantizadores do erro.
000D 000E	K(i)H ⁽⁷⁾ K(i)L	Corresponde ao maior valor de K (i=NIVEIS).
000F 0010	N(i)H N(i)L	Corresponde à correção em N se erro \geq K(i).
0011 0012	K(i-1)H K(i-1)L	Valor de K imediatamente inferior a K(i).
0013 0014	N(i-1)H N(i-1)L	Correção em N se K(i) > erro \geq K(i-1).
..... K ₁ H K ₁ L	Menor valor de K.
.....	N ₁ H N ₁ L	Correção em N se K ₂ > erro \geq K ₁ .
.....	XX XX	Não interessa.
.....	NOH NOL	Correção em N se K ₁ > erro.

(1) N tem um limite igual a 4095 (FFF_{Hex}). Podem ser especificados números entre 98% e 0,1% desse valor.

(2) VALIN representa a quantidade inicial da tensão aplicada ao motor. VALIN deve ser maior que NMIN.

(3) CT é um número calculado pela seguinte expressão:

$$CT = \frac{200 \text{ mseg.}}{22/fck}$$

onde fck é a frequência de "clock" do microprocessador.

(4) O número de ajustes se a situação de máximo ou de mínimo ciclo de trabalho ocorrer é limitado por esse parâmetro.

(5) Este parâmetro determina se a aceleração ou desaceleração controlada será rápida ou lenta. O número de divisões da variação de velocidade (ΔV) é dado por 2^{NDIV} ($NDIV > 0$).

(6) Podem ser especificados até 23 (17_{Hex}) valores de K.

APÊNDICE D

LISTAGEM DOS PROGRAMAS

PROGRAMA DE APOIO MÍNIMO

ENDER.	OP.	CODE	LABEL	MNEM.	OPERANDO	COMENTÁRIOS
F800	FE	0700	IRQ	LDX	EIRQ	Desvios para as rotinas de interrupção.
03	6E	00		JMP	0,X	
05	FE	0702	SWI	LDX	ESWI	
08	6E	00		JMP	0,X	
0A	FE	0704	NMI	LDX	ENMI	
0D	6E	00		JMP	0,X	
0F	8E	0742	PROXED	LDS	#STACK	Aponta próxima locação.
F812	20	12		BRA	NCHAN	
14	44		OUTH1	LSRA		
15	44			LSRA		
16	44			LSRA		
17	44			LSRA		
18	84	0F	OUTH2	ANDA	#\$0F	
1A	8B	30		ADDA	#\$30	
1C	81	39		CMPA	#\$39	
1E	23	3D		BLS	SAICAR	
F820	8B	07		ADDA	#\$07	
22	20	39		BRA	SAICAR	
24	8D	71	CHANGE	BSR	BADDR	Troca conteúdo de locação de memória.
26	8D	55	NCHAN	BSR	SLFCR	
28	CE	0708		LDX	#XHI	
2B	8D	62		BSR	OUT4HS	
2D	FE	0708		LDX	XHI	
F830	8D	5F		BSR	OUT2HS	
32	FF	0708		STX	XHI	
35	8D	6E		BSR	BYTE	
37	09			DEX		
38	A7	00		STAA	0,X	

3A	A1	00		CMPA	0,X	
3C	27	E8		BEQ	NCHAN	
3E	86	3F		LDAA	#'?	
F840	8D	1B		BSR	SAICAR	
F842	20	7D		BRA	CONTRL	
F844	8D	24	INHEX	BSR	ENTCAR	Entrada de ca
46	81	20		CMPA	#'SP	ráter hexade-
48	27	C5		BEQ	PROXED	cimal.
4A	80	30		SUBA	#\$30	
4C	2B	78		BMI	CONTR1	
4E	81	09		CMPA	#\$09	
F850	2F	0A		BLE	SAISUB	
52	81	11		CMPA	#\$11	
54	2B	70		BMI	CONTR1	
56	81	16		CMPA	#\$16	
58	2E	6C		BGT	CONTR1	
5A	80	07		SUBA	#\$07	
5C	39		SAISUB	RTS		
F85D	37		SAICAR	PSHB		Imprime um ca
5E	F6	D000	LOOP	LDAB	ACIACS	ráter (ASCII).
F861	57			ASRB		
62	57			ASRB		
63	24	F9		BCC	LOOP	
65	B7	D001		STAA	ACIAD	
68	33			PULB		
69	39			RTS		
F86A	B6	D000	ENTCAR	LDAA	ACIACS	Entrada de um
6D	47			ASRA		caráter.
6E	FA			BCC	ENTCAR	
F870	B6	D001		LDAA	ACIAD	
73	84	7F		ANDA	#\$7F	
75	81	7F		CMPA	#\$7F	
77	27	F1		BEQ	ENTCAR	

79	20	E2		BRA	SAICAR	
F87B	20	A7	CHANG1	BRA	CHANGE	
F87D	86	0A	SLFCR	LDAA	#\$0A	
F87F	8D	DC		BSR	SAICAR	
F881	86	0D		LDAA	#\$0D	
83	8D	D8		BSR	SAICAR	
85	39			RTS		
F886	A6	00	OUT2H	LDAA	0,X	Saída de 2 ca
88	8D	8A		BSR	OUTH L	racteres hexa
8A	A6	00		LDAA	0,X	decimais.
8C	08			INX		
8D	20	89		BRA	OUTHR	
F88F	8D	F5	OUT4HS	BSR	OUT2H	Saída de 4 ca
F891	8D	F3	OUT2HS	BSR	OUT2H	racteres hexa
93	86	20	OUTS	LDAA	#' SP	decimais + es
95	20	C6		BRA	SAICAR	paço.
F897	8D	0C	BADDR	BSR	BYTE	Entrada de 4
99	B7	0708		STAA	XHI	caracteres he
9C	8D	07		BSR	BYTE	
9E	B7	0709		STAA	XLOW	
F8A1	FE	0708		LDX	XHI	
A4	39			RTS		
F8A5	8D	9D	BYTE	BSR	INHEX	Entrada de 1
A7	48			ASLA		byte.
A8	48			ASLA		
A9	48			ASLA		
AA	48			ASLA		
AB	16			TAB		
AC	8D	96		BSR	INHEX	
AE	1B			ABA		
AF	39			RTS		

F8B0	8E	0742	RESET	LDS	#STACK	Seqüência de RESET.
B3	BF	0706		STS	SPH	
B6	CE	F8F1		LDX	#ENDSWI	
B9	FF	0702		STX	ESWI	
F8BC	86	03		LDAA	#\$03	
BE	B7	D000		STAA	ACIACS	
F8C1	86	11	CONTRL	LDAA	#\$11	
C3	B7	D000		STAA	ACIACS	
F8C6	8E	0742	CONTR1	LDS	#STACK	
C9	8D	B2		BSR	SLFCR	
CB	86	3A		LDAA	#':	Espera entra- da de um co - mando.
CD	8D	8E		BSR	SAICAR	
CF	8D	99		BSR	ENTCAR	
F8D1	16			TAB		
D2	8D	BF		BSR	OUTS	
D4	C1	4D		CMPB	#'M	
D6	27	A3		BEQ	CHANG1	
D8	C1	52		CMPB	#'R	
DA	27	18		BEQ	PRINT	
DC	C1	47		CMPB	#'G	
DE	26	E6		BNE	CONTR1	
F8E0	8D	B5		BSR	BADDR	
E2	BE	0706		LDS	SPH	
E5	30			TSX		
E6	B6	0708		LDAA	XHI	
E9	A7	05		STAA	5,X	
EB	B6	0709		LDAA	XLOW	
EE	A7	06		STAA	6,X	
F8F0	3B			RTI		
F1	BF	0706	ENDSWI	STS	SPH	
F4	8D	87	PRINT	BSR	SLFCR	
F6	FE	0706		LDX	SPH	Imprime con- teúdo dos re- gistradores.
F9	08			INX		
FA	8D	95		BSR	OUT2HS	
FC	8D	93		BSR	OUT2HS	
FE	8D	91		BSR	OUT2HS	
F900	8D	8D		BSR	OUT2HS	

02	8D	8B	BSR	OUT4HS
04	CE	0706	LDX	#SPH
07	8D	86	BSR	OUT4HS
09	20	BB	BRA	CONTR1

VETOR DE INTERRUPTÃO

ENDER.	CONTEÚDO	
FFF8	F8	IRQ
FFF9	00	
FFFA	F8	SWI
FFFB	05	
FFFC	F8	NMI
FFFD	0A	
FFFE	F8	RESET
FFFF	B0	

PROGRAMAS DO SISTEMA DE CONTROLE

ENDER	OP.	CODE	LABEL	MNEM.	OPERANDO	COMENTÁRIOS
						* Rotina INILIZ
						* Efetua inicializações + partida do motor
F910	4F			CLRA		Inicializar
11	B7	8009		STAA	PIACRA	PIA.
14	B7	800B		STAA	PIACRB	
17	86	F6		LDAA	#F6	
19	B7	8008		STAA	PIADRA	
1C	86	04		LDAA	#04	
1E	B7	8009		STAA	PIACRA	
F921	86	FF		LDAA	#FF	
23	B7	800A		STAA	PIADRB	
26	86	0F		LDAA	#0F	
28	B7	800B		STAA	PIACRB	
2B	86	91		LDAA	#91	Interrupção na
2D	B7	D000		STAA	ACIACS	entrada de ca racteres.
F930	86	01		LDAA	#01	
32	97	92		STAA	KEY	
34	DE	06		LDX	VALINH	Val. inic. de N.
36	DF	72		STX	NH	
38	CE	0600		LDX	#INICBF	Apont. do
3B	DF	7E		STX	APBUFH	"buffer" de
3D	DE	00		LDX	VREFH	escrita
3F	DF	88		STX	TEMP1H	
F941	4F			CLRA		
42	97	7C		STAA	VELINH	
44	97	7D		STAA	VELINL	
46	97	93		STAA	KONT	
48	97	94		STAA	INDICA	
4A	97	96		STAA	NCTMAX	
4C	97	97		STAA	NCTMIN	
4E	97	98		STAA	INDINT	

F950	CE	FB80		LDX	#ENDIRQ	Inicl. ender.
53	FF	0700		STX	EIRQ	rot. de IRQ.
56	CE	FDOC		LDX	#ENDSWI	Idem p/rot. de SWI.
59	FF	0702		STX	ESWI	
5C	0E			CLI		
5D	8D	0A		BSR	PREPAD	
F95F	8D	33		BSR	ACELCT	Aceleração con
F961	8D	11		BSR	INCFIN	trolada.
63	BD	FCFC		JSR	HABTRM	Iniciar mensa
66	7E	FB09		JMP	CMPPRG	gem. Compensação e controle.
				*Subrotina	PREPAD	
				*Prepara variáveis	p/aceleração/desac.	
				*controlada.		
F969	8D	0F	PREPAD	BSR	DETVRI	
6B	DE	7C		LDX	VELINH	
6D	DF	88		STX	TEMP1H	
6F	DE	74		LDX	INVRIH	
F971	DF	8A		STX	TEMP2H	
73	39			RTS		
				*Subrotina	INCFIN	
				*Estabelece último	incremento de vel. na	
				*aceleração.		
F974	BD	FAA4	INCFIN	JSR	PVFIN	
77	8D	3B		BSR	NOVCON	
79	39			RTS		
				*Subrotina	DETVRI	
				*Calcula valores de	incrementos de vel. e N.	
F97A	96	0B	DETVRI	LDAA	NDIV	
7C	74	0088	LOOP1	LSR	TEMP1H	
7F	76	0089		ROR	TEMP1L	
F982	4A			DECA		
83	26	F7		BNE	LOOP1	

85	DE	88		LDX	TEMP1H	
87	DF	74		STX	INVRIH	Incremento de
89	74	0088		LSR	TEMP1H	Vel.
8C	76	0089		ROR	TEMP1L	
8F	DE	88		LDX	TEMP1H	
F991	DF	76		STX	INCRNH	Incremento de
93	39			RTS		N.

*Subrotina ACELCT

*Efetua a aceleração controlada do motor.

F994	8D	7C	ACELCT	BSR	SAIDA	
96	8D	33	DESV2	BSR	CNTSUB	
98	DE	70		LDX	RPMH	
9A	26	04		BNE	ACEL	Verifica se o
9C	8D	66		BSR	ESPERA	motor partiu.
9E	20	F6		BRA	DESV2	
F9A0	BD	FAAA	ACEL	JSR	CNVRI	Calc. Nº de
A3	5A		ACELER	DECB		Vinterms.
A4	2A	01		BPL	SEGUE	
A6	39			RTS		
A7	0C		SEGUE	CLC		Calc. próxima
A8	96	89		LDAA	TEMP1L	Vinterm.
AA	9B	8B		ADDA	TEMP2L	
AC	97	89		STAA	TEMP1L	
AE	96	88		LDAA	TEMP1H	
F9B0	99	8A		ADCA	TEMP2H	
B2	97	88		STAA	TEMP1H	
B4	8D	15	NOVCON	BSR	CNTSUB	
B6	8D	3C		BSR	RPMVRI	Calc. Vatuai -
B8	2C	E9		BGE	ACELER	Vinterm.
BA	0C			CLC		Incrementar N
BB	96	73		LDAA	NL	
BD	9B	77		ADDA	INCRNL	
BF	97	73		STAA	NL	
F9C1	96	72		LDAA	NH	
C3	99	76		ADCA	INCRNH	

C5	97	72	STAA	NH
C7	8D	49	BSR	SAIDA
C9	20	E9	BRA	NOVCON

*Subrotina CNTSUB

*Efetua a contagem da velocidade.

F9CB	0F		CNTSUB	SEI	
CC	9F	84	STS	SAVEH	Salvar SP.
CE	DE	08	LDX	CTH	
F9D0	8E	0000	LDS	#0000	Zerar contador
D3	09		LOOP5	DEX	de pulsos.
D4	27	18	BEQ	FIM	
D6	B6	8008	LDAA	PIAORA	
D9	85	01	BITA	#01	
DB	26	02	BNE	DESV3	Entrada = "1"?
DD	20	F4	BRA	LOOP5	
DF	31		DESV3	INS	Inc. contador
F9E0	09		LOOP6	DEX	de pulsos.
E1	27	0B	BEQ	FIM	
E3	B6	8008	LDAA	PIAORA	
E6	85	01	BITA	#01	
E8	27	02	BEQ	DESV4	Entrada = "0"?
EA	20	F4	BRA	LOOP6	
EC	20	E5	DESV4	BRA	LOOP5
EE	9F	70	FIM	STS	Armazenar Va-
F9F0	9E	84	LDS	SAVEH	tual e recuperar
F2	0E		CLI		SP.
F3	39		RTS		

*Subrotina RPMVRI

*Calcula Vactual - Vinterm.

F9F4	0C		RPMVRI	CLC	
F5	96	71	LDAA	RPML	
F7	90	89	SUBA	TEMP1L	
F9	97	7B	STAA	RESUTL	
FB	96	70	LDAA	RPMH	
FD	92	88	SBCA	TEMP1H	

FF	97	7A		STAA	RESUTH
FA01	DE	7A		LDX	RESUTH
03	39			RTS	

*Subrotina ESPERA

FA04	86	04	ESPERA	LDAA	#04
06	8D	03	DENOVO	BSR	LOOP
08	4A			DECA	
09	26	FB		BNE	DENOVO
0B	CE	FFFF	LOOP	LDX	#FFFF
FA0E	09		ATS	DEX	
0F	26	FD		BNE	ATS
FA11	39			RTS	

*Subrotina SAIDA

*Envia o nº N ao GPG.

FA12	0C		SAIDA	CLC		
13	96	03		LDAA	NMAXL	
15	90	73		SUBA	NL	
17	97	8D		STAA	TEMP3L	
19	96	02		LDAA	NMAXH	
1B	92	72		SBCA	NH	
1D	97	8C		STAA	TEMP3H	
1F	DE	8C		LDX	TEMP3H	
FA21	2E	17		BGT	TSNMIN	NMAX>N?
23	CE	FD7D		LDX	#MSG1	
26	8D	59		BSR	DSPLAY	Mensagem.
28	7F	0094		CLR	INDICA	
2B	3F			SWI		Ajustar Valor
2C	0E			CLI		de N.
2D	96	96		LDAA	NCTMAX	
2F	91	0A		CMPA	NAJUST	Ajustes efetu
						ados?
FA31	27	59		BEQ	STOP	Parar motor.
33	96	98		LDAA	INDINT	
35	26	DB		BNE	SAIDA	
37	7E	FB09		JMP	CMPPRG	

3A	0C		TSNMIN	CLC		
3B	96	05		LDAA	NMINL	
3D	90	73		SUBA	NL	
FA3F	97	8D		STAA	TEMP3L	
FA41	96	04		LDAA	NMINH	
43	92	72		SBCA	NH	
45	97	8C		STAA	TEMP3H	
47	DE	8C		LDX	TEMP3H	
49	2D	1B		BLT	NVALID	N>NMIN?
4B	96	92		LDAA	KEY	KEY = 0 indica
4D	27	3D		BEQ	STOP	desac. até pa-
4F	CE	FD98		LDX	#MSG2	rar o motor.
FA52	8D	2D		BSR	DSPLAY	
54	7C	0094		INC	INDICA	
57	3F			SWI		Ajustar valor
58	0E			CLI		de N.
59	96	97		LDAA	NCTMIN	
5B	91	0A		CMPA	NAJUST	
5D	27	2D		BEQ	STOP	
5F	96	98		LDAA	INDINT	
FA61	26	AF		BNE	SAIDA	
63	7E	FB09		JMP	CMPPRG	
66	DE	72	NVALID	LDX	NH	
68	DF	8C		STX	TEMP3H	
6A	CE	0004		LDX	#0004	
6D	78	008D	LOOP3	ASL	TEMP3L	Ajustar N(16
FA70	79	008C		ROL	TEMP3H	bits) p/o GPG
73	09			DEX		(12 bits).
74	2E	F7		BGT	LOOP3	
76	96	8C		LDAA	TEMP3H	Enviar 8 bits
78	B7	800A		STAA	PIAORB	mais signf.de N.
7B	96	8D		LDAA	TEMP3L	Idem p/os 4 bits
7D	B7	8008		STAA	PIAORA	menos signf.
FA80	39			RTS		

*Subrotina DSPLAY

*Envia mensagem à tela do terminal.

FA81	A6	00	DSPLAY	LDAA	0,X	
83	27	06		BEQ	FINAL	
85	BD	F85D		JSR	SAICAR	
88	08			INX		
89	20	F6		BRA	DSPLAY	
8B	39		FINAL	RTS		

*Rotina STOP

*Efetua a parada do motor.

FA8C	86	02	STOP	LDAA	#02	
8E	B7	8008		STAA	PIAORA	BTP = "1".
FA91	CE	2400		LDX	#2400	
94	BD	FA0E		JSR	ATS	
97	86	06		LDAA	#06	
99	B7	8008		STAA	PIAORA	MR = "1".
9C	CE	FDB3		LDX	#MSG4	
9F	8D	E0		BSR	DSPLAY	
FAA1	7E	F8C1	JM	JMP	CONTRL	

*Subrotina PVFIN

*Faz última Vintrm = Vfinal.

FAA4	5F		PVFIN	CLRB		
A5	DE	00		LDX	VREFH	
A7	DF	88		STX	TEMP1H	
A9	39			RTS		

*Subrotina CNVRI

*Calcula n° de Vintrms.

FAAA	96	0B	CNVRI	LDAA	NDIV	
AC	C6	01	SHIFTB	LDAB	#01	
AE	58		ND	ASLB		Ac.B = N° de
AF	4A			DECA		Vintrms.
FAB0	26	FC		BNE	ND	
B2	39			RTS		

*Subrotina DECFIN

*Estabelece último incremento de Vel.

*na desaceleração.

FAB3	8D	EF	DECFIN	BSR	PVFIN
B5	8D	14		BSR	NCON
B7	39			RTS	

*Subrotina DSACTL

*Efetua a desaceleração controlada do motor.

FAB8	8D	F0	DSACTL	BSR	CNVRI
BA	5A		DSAL	DECB	
BB	2A	01		BPL	VAI
BD	39			RTS	
BE	0C		VAI	CLC	
BF	96	89		LDAA	TEMP1L
FAC1	90	8B		SUBA	TEMP2L
C3	97	89		STAA	TEMP1L
C5	96	88		LDAA	TEMP1H
C7	92	8A		SBCA	TEMP2H
C9	97	88		STAA	TEMP1H
CB	BD	F9CB	NCON	JSR	CNTSUB
CE	BD	F9F4		JSR	RPMVRI
FAD1	2F	E7		BLE	DSAL
D3	0C			CLC	
D4	96	73		LDAA	NL
D6	90	77		SUBA	INCRNL
D8	97	73		STAA	NL
DA	96	72		LDAA	NH
DC	92	76		SBCA	INCRNH
DE	97	72		STAA	NH
FAE0	BD	FA12		JSR	SAIDA
E3	20	E6		BRA	NCON

*Subrotina QUANER

*Determina a faixa de erro.

FAE5	7A	0095	QUANER	DEC	NTEST
------	----	------	--------	-----	-------

E8	0C		CLC		Calcula Erro-
E9	96	8F	LDAA	TEMP4L	K(i).
EB	A0	01	SUBA	1,X	
ED	97	91	STAA	TEMP5L	
FAEF	96	8E	LDAA	TEMP4H	
FAF1	A2	00	SBCA	0,X	
F3	97	90	STAA	TEMP5H	
F5	39		RTS		

*Subrotina INIAPA

*Inicializa apontador da tabela de K e ΔN.

FAF6	96	0C	INIAPA	LDAA	NIVEIS
F8	97	95	STAA	NTEST	
FA	CE	0000	LDX	#ENDPAR	
FD	DF	80	STX	APPARH	
FF	39		RTS		

*Subrotina ATUAPA

*Atualiza apontador da tabela de K e ΔN.

FB00	DE	80	ATUAPA	LDX	APPARH
FB02	08		INX		
03	08		INX		
04	08		INX		
05	08		INX		
06	DF	80	STX	APPARH	
08	39		RTS		

*Rotina CMPPRG

*Executa o algoritmo de controle.

FB09	BD	FA12	CMPPRG	JSR	SAIDA	
0C	BD	F9CB	ERZERO	JSR	CNTSUB	
0F	0C		CLC			
FB10	96	71	LDAA	RPML	Calcular Erro=	
12	90	01	SUBA	VREFL	Vatual -Vref.	
14	97	8F	STAA	TEMP4L		
16	96	70	LDAA	RPMH		
18	92	00	SBCA	VREFH		

*Subrotina DECFIN

*Estabelece último incremento de Vel.

*na desaceleração.

FAB3	8D	EF	DECFIN	BSR	PVFIN
B5	8D	14		BSR	NCON
B7	39			RTS	

*Subrotina DSACTL

*Efetua a desaceleração controlada do motor.

FAB8	8D	F0	DSACTL	BSR	CNVRI
BA	5A		DSAL	DECB	
BB	2A	01		BPL	VAI
BD	39			RTS	
BE	0C		VAI	CLC	
BF	96	89		LDAA	TEMP1L
FAC1	90	8B		SUBA	TEMP2L
C3	97	89		STAA	TEMP1L
C5	96	88		LDAA	TEMP1H
C7	92	8A		SBCA	TEMP2H
C9	97	88		STAA	TEMP1H
CB	BD	F9CB	NCON	JSR	CNTSUB
CE	BD	F9F4		JSR	RPMVRI
FAD1	2F	E7		BLE	DSAL
D3	0C			CLC	
D4	96	73		LDAA	NL
D6	90	77		SUBA	INCRNL
D8	97	73		STAA	NL
DA	96	72		LDAA	NH
DC	92	76		SBCA	INCRNH
DE	97	72		STAA	NH
FAE0	BD	FA12		JSR	SAIDA
E3	20	E6		BRA	NCON

*Subrotina QUANER

*Determina a faixa de erro.

FAE5	7A	0095	QUANER	DEC	NTEST
------	----	------	--------	-----	-------

1A	97	8E		STAA	TEMP4H	
1C	DE	8E		LDX	TEMP4H	
1E	27	EC		BEQ	ERZERO	Erro = 0 ?
FB20	2E	27		BGT	RPMOR	
FB22	8D	48		BSR	COM2BY	Comp. '2 do
24	DF	8E		STX	TEMP4H	erro.
26	8D	CE		BSR	INIAPA	
28	8D	BB	NOVOIN	BSR	QUANER	Erro < 0 en-
2A	DE	90		LDX	TEMP5H	tão incremen-
2C	2C	0A		BGE	INCRE	tar N.
2E	96	95		LDAA	NTEST	
FB30	27	04		BEQ	ULTINC	
32	8D	CC		BSR	ATUAPA	
34	20	F2		BRA	NOVOIN	
36	8D	C8	ULTINC	BSR	ATUAPA	
38	DE	80	INCRE	LDX	APPARH	
3A	0C			CLC		
3B	96	73		LDAA	NL	
3D	AB	03		ADDA	3,X	
3F	97	73		STAA	NL	
FB41	96	72		LDAA	NH	
43	A9	02		ADCA	2,X	
45	97	72		STAA	NH	
47	20	C0		BRA	CMPPRG	
49	8D	AB	RPMOR	BSR	INIAPA	
4B	8D	98	NOVODE	BSR	QUANER	Erro > 0 então
4D	DE	90		LDX	TEMP5H	decrementar N.
4F	2C	0A		BGE	DECRE	
FB51	96	95		LDAA	NTEST	
53	27	04		BEQ	ULTDEC	
55	8D	A9		BRS	ATUAPA	
57	20	F2		BRA	NOVODE	
59	8D	A5	ULTDEC	BSR	ATUAPA	
5B	DE	80	DECRE	LDX	APPARH	

5D	0C		CLC	
5E	96	73	LDAA	NL
FB60	A0	03	SUBA	3,X
62	97	73	STAA	NL
64	96	72	LDAA	NH
FB66	A2	02	SBCA	2,X
68	97	72	STAA	NH
6A	20	9D	BRA	CMPPRG

*Subrotina COM2BY

*Calcula o comp.'2 do conteúdo do reg.X.

FB6C	DF	86	COM2BY	STX	CMPBYH	
6E	96	87		LDAA	CMPBYL	
FB70	26	05		BNE	COMP2	
72	70	0086		NEG	CMPBYH	
75	20	06		BRA	C2	
77	70	0087	COMP2	NEG	CMPBYL	
7A	73	0086		COM	CMPBYH	
7D	DE	86	C2	LDX	CMPBYH	X ← comp.'2
7F	39			RTS		de X.

*Rotina INTER

*Atende os pedidos de interrupção.

FB80	B6	D000		LDAA	ACIACS	
83	85	80		BITA	#80	
85	26	23		BNE	ENTCAR	Entrada de
87	B6	800B		LDAA	PIACRB	caráter?
8A	85	40		BITA	#40	
8C	27	16		BEQ	INTREX	Sobrecorrente?
8E	BD	FA04		JSR	ESPERA	Atraso p/verif.
FB91	B6	800A		LDAA	PIAORB	se sobrecorren-
94	B6	8008		LDAA	PIAORA	te é transitô-
97	85	08		BITA	#08	ria.
99	26	50		BNE	RETORN	
9B	CE	FDCC		LDX	#MSG3	
9E	BD	FA81		JSR	DSPLAY	
FBA1	7E	FA8C	STOP1	JMP	STOP	

A4	B6	800A	INTREX	LDA	PIAORB	Se interrup.ex- terna,então des- viar p/Prog. de
A7	7E	F8C6		JMP	CONTR1	
AA	85	01	ENTCAR	BITA	#01	Apoio.
AC	26	13		BNE	RECCAR	
FBAE	DE	82		LDX	INIMEN	
FBB0	08			INX		
B1	DF	82		STX	INIMEN	
B3	A6	00		LDA	0,X	
B5	27	04		BEQ	FIMMEN	Fim de mensagem?
B7	B7	D001		STAA	ACIAD	Imprimir um cará- ter.
BA	3B			RTI		
BB	86	91	FIMMEN	LDA	#91	Interrup.na en- trada de caracte- res.
BD	B7	D000		STAA	ACIACS	
FBC0	3B			RTI		
C1	B6	D001	RECCAR	LDA	ACIAD	Ler um caráter.
C4	84	7F		ANDA	#7F	
C6	DE	7E		LDX	APBUFH	
C8	A7	00		STAA	0,X	Guardar caract. no "buffer" de escrita.
CA	16			TAB		
CB	C1	56		CMPB	#'V	
CD	26	1D		BNE	TESTS	
CF	96	93		LDA	KONT	
FBD1	26	18		BNE	RETONR	KONT ≠ 0 indica que caract. an- terior é hex.
D3	F7	D001		STAB	ACIAD	
D6	BD	F893		JSR	OUTS	
D9	CE	0070		LDX	#ENDRPM	Imprimir Vatuál e N.
DC	BD	F886		JSR	OUT2H	
DF	BD	F891		JSR	OUT2HS	
FBE2	BD	F886		JSR	OUT2H	
E5	BD	F886		JSR	OUT2H	
E8	BD	F87D		JSR	SLFCR	
EB	3B		RETURN	RTI		

EC	C1	53	TESTS	CMPB	#'S	
EE	26	0C		BNE	TESTT	
FBF0	96	93		LDAA	KONT	
F2	26	F7		BNE	RETORN	
F4	F7	D001		STAB	ACIAD	
FBF7	BD	F87D		JSR	SLFCR	
FA	20	A5		BRA	STOP1	Parar o motor.
FC	C1	54	TESTT	CMPB	#'T	
FE	26	1E		BNE	TESTHX	
FC00	96	93		LDAA	KONT	
02	26	E7		BNE	RETORN	
04	F7	D001		STAB	ACIAD	
07	BD	F87D		JSR	SLFCR	
0A	7F	0092		CLR	KEY	
0D	DE	00		LDX	VREFH	
0F	DF	88		STX	TEMP1H	
FC11	DF	7C		STX	VELINH	
13	BD	F969		JSR	PREPAD	
16	BD	FAB8		JSR	DSACTL	Desacel. até
19	BD	FAB3		JSR	DECFIN	parar.
1C	20	83		BRA	STOP1	
1E	C1	30	TESTHX	CMPB	#30	Seq. p/verif.
FC20	2B	3C		BMI	TESTN	se caract. é
22	C1	39		CMPB	#39	hex.
24	2F	0A		BLE	ASCHEX	
26	C1	41		CMPB	#'A	
28	2B	34		BMI	TESTN	
2A	C1	46		CMPB	#'F	
2C	2E	30		BGT	TESTN	
2E	C0	07		SUBB	#07	
FC30	C4	0F	ASCHEX	ANDB	#0F	
32	7C	0093		INC	KONT	
35	96	93		LDAA	KONT	
37	81	04		CMPA	#04	
39	2E	22		BGT	RETINT	

3B	DE	7E		LDX	APBUFH	
3D	A6	00		LDAA	0,X	
3F	B7	D001		STAA	ACIAD	
FC42	E7	10		STAB	10,X	
44	08			INX		
45	DF	7E		STX	APBUFH	
47	96	93		LDAA	KONT	
49	81	04		CMPA	#04	Mais de 4 ca-
4B	26	10		BNE	RETINT	ract.hex.,en-
4D	BD	F893		JSR	OUTS	tão RTI.
FC50	BD	FCE6		JSR	INIBUF	
53	BD	FCEC		JSR	OBBYTE	Seq. p/guar-
56	97	78		STAA	VRNOVH	dar nova Vref.
58	BD	FCEC		JSR	OBBYTE	
5B	97	79		STAA	VRNOVL	
5D	3B		RETINT	RTI		
5E	C1	4E	TESTN	CMPB	#'N	Seq. p/variar
FC60	26	0C		BNE	TESTU	velocidade a
62	96	93		LDAA	KONT	través do al-
64	81	04		CMPA	#04	gor. de com -
66	2D	F5		BLT	RETINT	pensação.
68	8D	6E		BSR	TROVEL	
6A	BD	F87D		JSR	SLFCR	
6D	3B			RTI		
6E	C1	55	TESTU	CMPB	#'U	Variar Vel. uni-
FC70	26	5C		BNE	TESTI	formemente.
72	96	93		LDAA	KONT	
74	81	04		CMPA	#04	
76	2D	E5		BLT	RETINT	
78	DE	00		LDX	VREFH	
7A	DF	7C		STX	VELINH	
7C	8D	5A		BSR	TROVEL	
7E	8D	F87D		JSR	SLFCR	
FC81	7C	0098		INC	INDINT	

FC84	0C			CLC		
85	96	79		LDAA	VRNOVL	
87	90	7D		SUBA	VELINL	
89	97	89		STAA	TEMP1L	
8B	96	78		LDAA	VRNOVH	
8D	92	7C		SBCA	VELINH	
8F	97	88		STAA	TEMP1H	
FC91	96	0B		LDAA	NDIV	
93	4C			INCA		
94	BD	FAAC		JSR	SHIFTB	
97	96	88		LDAA	TEMP1H	
99	26	0D		BNE	VARСУF	
9B	D1	89		CMPB	TEMP1L	
9D	2F	09		BLE	VARСУF	Variação de
9F	CE	F0DB		LDX	#END2	velc. é sufi-
FCA2	DF	82		STX	INIMEN	ciente?
A4	8D	5B		BSR	PRICAR	
A6	20	17		BRA	RETCMP	
A8	DE	88	VARСУF	LDX	TEMP1H	
AA	2C	17		BGE	ACL	Acel. ou desa
AC	7C	0092		INC	KEY	cel.?
AF	BD	FB6C		JSR	COM2BY	
FCB2	DF	88		STX	TEMP1H	
B4	BD	F969		JSR	PREPAD	
B7	BD	FAB8		JSR	DSACTL	
BA	BD	FAB3		JSR	DECFIN	
BD	8D	3D	MENSAG	BSR	HABTRM	Vel.final atin
BF	7F	0098	RETCMP	CLR	INDINT	gida.
FCC2	3B			RTI		
C3	BD	F969	ACL	JSR	PREPAD	
C6	BD	F9A0		JSR	ACEL	
FCC9	BD	F974		JSR	INCFIN	

CC	20	EF		BRA	MENSAG	
CE	C1	49	TESTI	CMPB	#'I	Invalidar en-
FCD0	26	8B		BNE	RETINT	trada de ca-
D2	8D	08		BSR	INVCAR	ract. hex.
D4	BD	F87D		JSR	SLFCR	
D7	3B			RTI		

*Subrotinas TROVEL, INVCAR e INIBUF

FCD8	DE	78	TROVEL	LDX	VRNOVH	Trocar Vref.
DA	DF	00		STX	VREFH	
DC	4F		INVCAR	CLRA		
DD	97	93		STAA	KONT	
DF	97	96		STAA	NCTMAX	Inicializar o
FCE1	97	97		STAA	NCTMIN	nº de ajustes
						em N.
E3	F7	D001		STAB	ACIAD	Imprimir umca
						râter.
E6	CE	0600	INIBUF	LDX	#INICBF	Inicialize a
E9	DF	7E		STX	APBUFH	pontador do
EB	39			RTS		"buffer" de
						escrita.

*Subrotina OBBYTE
 *Obtém byte do "buffer" de
 *escrita.

FCEC	A6	10	OBBYTE	LDAA	10,X
EE	8D	06		BSR	BYTESQ
FCF0	08			INX	
F1	A6	10		LDAA	10,X
F3	1B			ABA	
F4	08			INX	
F5	39			RTS	

*Subrotina BYTESQ

FCF6	48		BYTESQ	ASLA	
------	----	--	--------	------	--

F7	48	ASLA
F8	48	ASLA
F9	48	ASLA
FA	16	TAB
FB	39	RTS

*Subrotina HABTRM

*Habilita interrupção na transmissão de
*caracteres.

FCFC	CE	FD5F	HABTRM	LDX	#END1
FF	DF	82		STX	INIMEN
FD01	86	31	PRICAR	LDAA	#31
03	B7	D000		STAA	ACIACS
06	86	0A		LDAA	#0A
08	B7	D001		STAA	ACIAD
0B	39			RTS	

*Rotina de SWI

*Efetua ajustes no valor de N.

FD0C	BD	FA04		JSR	ESPERA	
0F	96	94		LDAA	INDICA	
FD11	26	25		BNE	INCN	
13	7C	0096		INC	NCTMAX	Incrm. nº de
16	96	98		LDAA	INDINT	ajustes p/NMAX.
18	27	0E		BEQ	CMP1	INDINT = 0 in
1A	0C			CLC		dica ajustes
1B	96	73		LDAA	NL	durante com -
1D	90	77		SUBA	INCRNL	pensação.
1F	97	73		STAA	NL	
FD21	96	72		LDAA	NH	
23	92	76		SBCA	INCRNH	
25	97	72		STAA	NH	
27	3B			RTI		
28	DE	80	CMP1	LDX	APPARH	
2A	0C			CLC		
2B	96	73		LDAA	NL	

2D	A0	03		SUBA	3,X
2F	97	73		STAA	NL
FD31	96	72		LDAA	NH
33	A2	02		SBCA	2,X
35	97	72		STAA	NH
37	3B			RTI	

*Seq. p/ajustar N se NMIN>N.

38	7C	0097	INCN	INC	NCTMIN
3B	96	98		LDAA	INDINT
3D	27	0E		BEQ	CMP2
3F	0C			CLC	
FD40	96	73		LDAA	NL
42	9B	77		ADDA	INCRNL
44	97	73		STAA	NL
46	96	72		LDAA	NH
48	99	76		ADCA	INCRNH
4A	97	72		STAA	NH
4C	3B			RTI	
FD4D	DE	80	CMP2	LDX	APPARH
4F	0C			CLC	
FD50	96	73		LDAA	NL
52	AB	03		ADDA	3,X
54	97	73		STAA	NL
56	96	72		LDAA	NH
58	A9	02		ADCA	2,X
5A	97	72		STAA	NH
5C	3B			RTI	

APÊNDICE E

RESULTADOS EXPERIMENTAIS

Os resultados apresentados a seguir foram obtidos aplicando-se degraus de tensão à armadura do motor, acoplado a um gerador, e registrando-se os valores de velocidade a cada 200m seg. Os degraus foram gerados a partir da variação do número N aplicado à entrada do gerador de pulsos de gatilhamento do chaveador. Nos testes, motor e gerador foram excitados com uma corrente de campo igual a 0,5A cada um.

$\Delta N - 0$ a 1.000

Valores de velocidade em rpm

1) 27	9) 373	17) 673	25) 894	33) 1064
2) 61	10) 426	18) 701	26) 911	34) 1084
3) 103	11) 464	19) 734	27) 938	35) 1103
4) 152	12) 513	20) 773	28) 963	36) 1120
5) 199	13) 563	21) 784	29) 980	37) 1141
6) 254	14) 569	22) 813	30) 1005	38) 1160
7) 289	15) 594	23) 842	31) 1022	39) 1184
8) 352	16) 639	24) 868	32) 1044	40) 1229

Vfinal = 1282 rpm

 $\Delta N - 2000$ a 3280

Valores de velocidade em rpm

1) 1334	9) 1467	17) 1554	25) 1627	33) 1624
2) 1367	10) 1478	18) 1564	26) 1626	34) 1624
3) 1391	11) 1490	19) 1575	27) 1626	35) 1624
4) 1405	12) 1501	20) 1587	28) 1626	36) 1624
5) 1418	13) 1513	21) 1598	29) 1626	37) 1624
6) 1430	14) 1523	22) 1616	30) 1625	38) 1625
7) 1443	15) 1533	23) 1628	31) 1625	39) 1624
8) 1454	16) 1545	24) 1627	32) 1624	40) 1624

Vfinal = 1624 rpm

REFERÊNCIAS BIBLIOGRÁFICAS

- |1| KRISHNAN, THADIAPPAN & RAMASWAMI, BELLAMKONDA - "A Fast-Response DC Motor Speed Control System", IEEE Transactions on Industry Applications, Vol. IA-10, N° 5, Sep/Oct 1974, pp. 643-651.
- |2| MALONEY, T. J. & ALVARADO, F.L - "A Digital Method for DC Motor Speed Control", IEEE Transactions on IECI, Vol. IECI-23, N° 1, Feb 1976, pp. 44-46.
- |3| BOSE, B. K. & JENTZEN, K. J. - "Digital Speed Control of a DC Motor with Phase-Locked Loop Regulation", IEEE Transactions on IECI, Vol. IECI-25, N° 1, Feb 1978, pp. 10-13.
- |4| EAPEN, K. & VENKATESAN, K. - "Phase-Locked Loop DC Motor Drive with Improved Transient Performance", IEEE Transactions on IECI, Vol. IECI, N° 4, Nov 1981, pp. 347-352.
- |5| LIN, A. K. & KDEPSEL, W.W. - "A Microprocessor Speed Control System", IEEE Transactions on IECI, Vol. IECI-24, N° 3, Aug 1977, pp. 241-247.
- |6| KONISHI, T. KAMIYAMA, K. & OHMAE, T. - "A Performance Analysis of Microprocessor - Based Control Systems Applied to Adjustable Speed Motor Drives", IEEE Transactions on Industry Applications, Vol. IA-16, N° 3, May/Jun 1980, pp. 378-387.

- [7] ATHANI, V.V. & DESHPANDE, S.M - "Microprocessor Control of a Three-Phase Inverter used in Induction Motor Speed Control System", in IEEE-IECI '80 Proc., March 1980, pp.45-52.
- [8] PALANICHAMY, S. & PURUSHOTHAMAN, K. V. - "Microprocessor-Controlled Multimotor DC Drive System", in IEEE-IECI '81 Proc., Nov 1981, pp. 258-263.
- [9] _____ - "Experimental Investigations on a Microprocessor Controlled Thyristorized DC Drive System", Proceedings of the International Conference on Microcomputer Application to Industrial Control, Calcutta, Feb 14-16, 1981, pp. 157-163.
- [10] OHMAE, T., MATSUDA, T., SUZUKI, T., AZUSAWA, N., KAMIYAMA, K. & KONISHI, T. - "A Microprocessor - Controlled Fast-Response Speed Regulator with Dual Mode Current Loop for DCM Drives", IEEE Trans. on Ind. Applic., Vol. IA-16, N° 3, May/June 1980, pp. 388-394.
- [11] DEMERLE, M. FROMONT, J. - "Speed Measure and Speed Control with a Multi-microprocessor System on DC Motors", in IEEE-IECI '80 Proc., March 1980, pp. 40-44.
- [12] HARASHIMA, F., KOYAMA, M., KONDO, S. & NAITOH, H. - "Performance Improvement of Microprocessor-Based Digital PLL Speed Control System", in IEEE-IECI '80 Proc., March 1980, pp. 53-58.

- [13] LE-HUY, H. - "A Synchronous Thyristorized DC Motor Drive", IEEE Trans. on Ind. Applic. Vol. IA-15, Mar/Apr 1979, pp. 183-188
- [14] WANG, H. C. - "Sampling Period and Stability Analysis for the Microcomputer-Based Motor Control Systems", IEEE Trans. on IECT, Vol. IECI-28, N° 2, May 2981, pp. 98-102.
- [15] MUKHERJEE, D. S. - "Microprocessor Based Ward Leonard Control", Proc. of the International Conference on Microcomputer Applications to Industrial Control, Calcutta, Feb 14-16, 1981, pp. 151-156.
- [16] OHMAE, T. MATSUDA, T., KAMIYAMA, K. TACHIKAWA, M. - "A Microprocessor-Controlled High-Accuracy Wide-Range Speed Regulator for Motor Drives", in IEEE-IECI '81 Proc., Nov 1981, pp. 381-386.
- [17] DEEP, G. S., CAVALCANTI, J. H. F. & ALVES, R.N.C. - "Microprocessor Based D.C. Motor Speed Control", Proc. of the International Conference on Microcomputer Applications to Industrial Control, Calcutta, Feb 14-16, 1981, pp.133-137.
- [18] SHIMOKAWA, Y., IANO, K. & MATSUDAIRA, T. - "A Microprocessor-Based High Level Programmable Controller", in IEEE-IECI '80 Proc., March 1980, pp. 64-70.
- [19] WOUK, V. - "High Efficiency, High Power, Load Insentive DC Chopper for Electronic Automobile Speed Control", IEEE Conference Record of Fourth Annual Meeting of Industry and General Applications Group", 1969, pp. 393-402.

- [20] ORHUN, E. - "The Programming of Microprocessors for the Control of Gas Turbine Engines" - Proc. of the International Conference on Microcomputer Applic. to Ind. Control, Calcutta, Feb 14-16, 1981, pp. 203-209.
- [21] ORTEGA, R. - "Experimental Evaluation of Four Microprocessor-Based Advanced Control Algorithms", Holland Publishing Company-Microprocessing & Microprogramming 10, 1981, pp. 229-245.
- [22] M6800 Microprocessor Applications Manual-Motorola Inc.1975.
- [23] CAVALCANTI, J. H. F., DEEP, G.S. & ALVES, R.N.C. -"Sistema de Desenvolvimento de Software de Microprocessadores para Aplicações em Tempo Real" - 8º PANEL EXPODATA - Buenos Aires, ARG., Mar 1981.
- [24] DEEP, G.S., CAVALCANTI, J. H.F. & ALVES, R.N.C. -"GPCT-5- Um Gerador Programável de Pulsos de Gatilhamento de Tiristores", Anais do 3º Congresso Brasileiro de Automática, 16 a 19 set 1980, Rio de Janeiro,BR., pp. 423-428.
- [25] CAVALCANTI, J. H. F. & DEEP, G.S. - "Scheduler Misto para Controle de Processos Industriais em Tempo Real usando Microcomputadores", XIV Congresso Nacional de Informática,S. Paulo, 1981.
- [26] _____ - "Um KERNEL para Controle de Processos Industriais em Tempo Real usando Microcomputadores", Anais do VII Seminário Integrado de Hardware e Software,Campinas, Julho de 1980.

- [27] CAVALCANTI, J. H. F.; DEEP, G. S.; ROCHA NETO, J. S. & SILVA, I.S.S. - "Multimicroprocessador para aplicações em Tempo Real (MATER)", Anais do Primeiro Simpósio em Controle de Processos por Computador (Primeiro SICOP), Rio de Janeiro, 1981, pp. 207-209.
- [28] ROCHA NETO, J.S. - "Multiprocessador para Aplicações em Tempo Real (MATER)", Dissertação de Mestrado - DEE - CCT - UFPb, março de 1982.
- [29] CAVALCANTI, J.H.F.; PINTO, A.V. & DEEP, G. S. - "Sistema de Ligação entre PDP-11 e Microcomputadores-SLPM", PANEL EXPODATA , Lima, Peru, 12-16 julho de 1982.
- [30] "Assembly Instructions for the Motorola M6800 Design Evaluation Kit", MOTOROLA INC., 1975