UNIVERSIDADE FEDERAL DE SANTA CATARINA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

# LINGUAGEM HIBRIDA ENTRE UM COMPUTADOR DIGITAL PDP11-40 E UM COMPUTADOR ANALÓGICO RA770

TESE SUBMETIDA À UNIVERSIDADE FEDERAL DE SANTA CATARINA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA.

JORGE EVANDRO DE ARAUJO RODRIGUES

FLORIANÓPOLIS, JANEIRO-1978

LINGUAGEM HÍBRIDA ENTRE UM COMPUTADOR DIGITAL PDP 11-40 E UM COMPUTADOR ANALÓGICO RA 770

JORGE EVANDRO DE ARAÚJO RODRIGUES

ESTA DISSERTAÇÃO FOI JULGADA PARA A OBTENÇÃO DO TÍTULO DE MESTRE EM ENGENHARIA. ESPECIALIDADE ENGENHARIA ELÉTRICA E APROVADA EM SUA FORMA FINAL PELO CURSO DE PÓS-GRADUAÇÃO.

Prof. Rajamani Doraiswami, Ph.D.
ORIENTADOR

Prof. Hans Helmut Zürn, Ph.D.
Coordenador do Curso de Pós-Graduação em Engenharia Elétrica.

APRESENTADA PERANTE A BANCA EXAMINADORA
COMPOSTA PELOS PROFESSORES:

Prof. Rajamani Doraiswami, Ph.D.

Prof. Hans Helmut Zürn, Ph.D.

Prof. Renato Carlson Dr. Ing.

Este trabalho foi financiado pelo Conselho Nacional de Pesquisas

À Ana, minha esposa

Ao meu filho

À meus pais e irmãos

# R E S U M O

Este trabalho descreve a criação de uma linguagem de alto nível de apoio à implementação do hardware de uma interface entre um minicomputador PDP 11/40 da Digital Equipment Corporation e um Computador Analógico RA 770 da Telefunken.

Esta linguagem é composta de cinco Subrotinas em linguagem Assembler inseridas no BASIC-PTS, nominalmente KAD, DAC, ENDE, CALC, POT, com o propósito de converter os dados analógicos em digitais, converter dados digitais em analógicos, endereçar amplificadores e ajustar servopotenciômetros.

# A B S T R A C T

This dissertation describes the development of a high – level language used in connection with the interface ... hardware between a PDP 11/40 minicomputer of the Digital Equipment Corporation and an RA 770 Analog computer of Telefunken.

This language consists of five subroutines in Assembler language inserted in BASIC - PTS. They are called KAD, DAC, ENDE, CALC, POT and their purpose is to convert analog data into digital, digital data into analog, address amplifiers and adjust servo potentiometers.

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

# LINGUAGEM HÍBRIDA ENTRE UM COMPUTADOR DIGITAL PDP 11-40 E UM COMPUTADOR ANALÓGICO RA 770

DISSERTAÇÃO SUBMETIDA A UNIVERSIDADE FEDERAL DE SANTA CATARINA COMO REQUISITO PARCIAL PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA.

JORGE EVANDRO DE ARAÚJO RODRIGUES

Janeiro de 1978

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

# LINGUAGEM HÍBRIDA ENTRE UM COMPUTADOR DIGITAL PDP 11-40 E UM COMPUTADOR ANALÓGICO RA 770

DISSERTAÇÃO SUBMETIDA A UNIVERSIDADE FEDERAL DE SANTA CATARINA COMO REQUISITO PARCIAL PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA.

JORGE EVANDRO DE ARAÚJO RODRIGUES

Janeiro de 1978

# ÍNDICE

Resumo
Abstract

pag.

CAPÍTULO I - Introdução .....02

1.1 - Desenvolvimento Histórico .....02
1.2 - Proposição .....03

CAPÍTULO II - Periféricos Envolvidos no Desenvolvimento da Linguagem Híbrida .....06
2.1 - Introdução .....06
2.2 - Conversor Analógico Digital .....06
2.2.1 - Registro de Estado do Conversor A-D .....06
2.2.2 - Registro de Armazenamento do Conversor A-D .....07
2.3 - Real Time Clock .....08
2.3.1 - Registro de Estado do LPSKW .....08
2.3.2 - Registro de Armazenamento e Registro Contador do LPSKW .....10
2.4 - Conversores Digitais-Analógicos .....10
2.5 - Tratamento dos Modos de Cálculos e Endereçamento Analógico Através do PDP 11-40 .....11
2.5.1 - Modos de Cálculo .....11
2.5.2 - Seleção de Endereços .....14
2.6 - Utilização da Palavra de Deteção .....15
2.7 - Condicionadores de Sinais .....16
2.8 - Configuração do Sistema Híbrido .....16
2.9 - Conclusão .....17

CAPÍTULO III- Descrição da Linguagem Híbrida .....20
3.1 - Introdução .....20
3.2 - Descrição da Subrotina KAD .....20
3.2.1 - Algorítmo para a Subrotina KAD .....20
3.2.2 - Fluxograma da Subrotina KAD .....23
3.2.3 - Listagem da Subrotina KAD .....26
3.3 - Descrição da Subrotina DAC .....28
3.3.1 - Algoritmo da Subrotina DAC .....28
3.3.2 - Fluxograma da Subrotina DAC .....30
3.3.3 - Listagem da Subrotina DAC .....32

pag.

3.4 - Descrição da Subrotina CALC .....34
3.4.1 - Algoritmo da Subrotina CALC .....34
3.4.2 - Fluxograma da Subrotina CALC .....35
3.4.3 - Listagem da Subrotina CALC .....37
3.5 - Descrição da Subrotina ENDE .....38
3.5.1 - Algoritmo da Subrotina ENDE .....38
3.5.2 - Fluxograma da Subrotina ENDE .....39
3.5.3 - Listagem da Subrotina ENDE .....42
3.6 - Descrição da Subrotina POT .....43
3.6.1 - Algoritmo da Subrotina POT .....43
3.6.2 - Fluxograma da Subrotina POT .....45
3.6.3 - Listagem da Subrotina POT .....49
3.7 - Conclusão .....51

CAPÍTULO IV - Aplicação da Linguagem .....53
4.1 - Introdução .....53
4.2 - Formulação do Problema .....53
4.3 - Resultados Obtidos .....56
4.4 - Conclusão .....56

CAPÍTULO V - Conclusão .....62

Anexo I - Cuidados de Operação .....64
Anexo II - Listagem da FTBL e Mapa de Endereço das Subrotinas .....69

# CAPÍTULO I - INTRODUÇÃO

# CAPÍTULO 1 - INTRODUÇÃO

## 1.1 - Desenvolvimento Histórico

Os computadores digitais são bastante usados na análise de problemas de engenharia, enquanto que os computadores analógicos fazem um papel relativamente menor. Ainda que largamente aplicados na indústria, os computadores digitais apresentam deficiência significativa, por serem essencialmente máquinas série as quais requerem, portanto, tempo relativamente grande para solucionar problemas. Isto, significa que os métodos numéricos utilizados na análise de grandes sistemas, requerem tempo de solução e espaço de memória relativamente grandes.

Em 1930, o analisador diferencial mecânico com as características dos computadores analógicos atuais, foi usado para estudar a dinâmica dos sistemas de potência. Este tipo de computador oferece a capacidade de integração em paralelo, uma vantagem que ele ainda mantém sobre o computador digital.

No desenvolvimento tecnológico do computador digital, foram inseridos na sua estrutura, conversores analógicos-digitais e digitais-analógicos que permitiram a comunicação entre os computadores digitais e analógicos. Em meados de 1960, surgiram os computadores híbridos, consistindo basicamente de um computador analógico interligado a um computador digital, combinando a natureza de integração em paralelo, inerentes dos computadores analógicos, com a superior precisão de cálculo e facilidade de programação do computador digital.

A unidade híbrida assim formada, apresenta, entre outras,as vantagens de solucionar problemas não lineares através da simulação do mesmo, no computador analógico; modificação de parâmetros do problema, quando a solução está sendo encontrada;monitoramento do sistema analógico e armazenamento de dados para análise posterior.

A utilidade dos computadores híbridos, contudo, sempre deve ser acompanhada pela parte de programação (software) em linguagem de alto nível, evitando, assim, a necessidade do usuário conhecer programação em linguagem de baixo nível e propiciando ao

mesmo a facilidade de exploração da capacidade completa do sistema, sem ter que gatar muito tempo no desenvolvimento de linguagem de baixo nível. O início da computação híbrida, deu-se com o pareci - mento dos computadores analógicos tipo AD - 256, EAI TR48 e 8800 interligados aos computadores digitais. Da combinação das capacidades dos computadores analógicos e digitais surgiram muitas aplicações novas de importancia para soluções de problema de engen haria. Uma destas, por exemplo, é a simulação do fenômeno de on- das caminhantes em sistema de transmissão (3).

Na década atual, houve um grande progresso na tecnologia utilizada nos computadores híbridos com a aplicação de circuitos integrados lineares monolíticos particularmente amplificadores ope racionais(4). Por exemplo, os amplificadores no EAI 781 analógico híbrido introduzido em 1972, usa amplificadores operacionais IC, dando maior dinamismo a esta configuração, que se torna melhor que os antigos EAI 7800(4).

É ainda desta década, a preocupação de construir linguagens híbridas de apoio em ligações entre computadores específicos. Po de-se citar os trabalhos de Robert Eatock e David Al - Dabass(5), que desenvolveram uma linguagem híbrida-série usando a subrotina EXF da linguagem BASIC, para uma ligação entre um PDP 11-20 e um EAI 580 e o de Sø e Høejberg, que em 1974, na Dinamarca,desenvol veu uma linguagem híbrida usando o computador digital PDP8/I e o computador analógico EAI-680. Esta linguagem se apoiou no compila dor FORTRAN.

Em 1976, foi apresentado na Universidade Federal de Santa Catarina, um trabalho feito por Daniel Augusto Martins(1) que tra tava do hardware da interligação entre um minicomputador PDP11-40 e um computador analógico RA 770.

## 1.2. - Proposição

O trabalho desenvolvido por Martins criou a necessidade de desenvolvimento de uma linguagem híbrida compatível com o sistema

híbrido existente na UFSC.

O presente trabalho propõe-se a criar uma linguagem híbrida de alto nível, inserindo cinco subrotinas Assembler no BASIC/PTS.

Este trabalho apresenta no Capítulo II, os periféricos do PDP 11-40, modos de cálculo e endereçamento assim como placas de interface utilizados no mesmo. O Capítulo III, apresenta cinco subrotinas Assembler. Com o propósito de converter os dados analógicos em digitais, converter dados digitais em analógicos, endereçar amplificadores e ajustar servopotenciômetros, estas cinco subrotinas constituem a linguagem híbrida. No Capítulo IV é apresentado um exemplo de aplicação da linguagem híbrida. No Capítulo V, serão tiradas as conclusões sobre o trabalho.

O ANEXO I, fornece informações sobre os cuidados a serem tomados antes que os computadores analógicos e digitais venham a ser utilizados como unidade híbrida, assim como o método de utilização desta linguagem e a listagem do programa utilizado no Capítulo IV.

No ANEXO II, são apresentadas as tabelas de função do usuário (FTBL) das cinco subrotinas,assim como mapa criado pelo LINK 11-S(8), que mostra os endereços de todos os módulos objetos.

# CAPÍTULO II

## PERIFÉRICOS ENVOLVIDOS NO DESENVOLVIMENTO DA LINGUAGEM HÍBRIDA

# CAPÍTULO 2 - PERIFÉRICOS ENVOLVIDOS NO DESENVOLVIMENTO DA LINGUAGEM HÍBRIDA

## 2.1 - Introdução

Neste capítulo serão descritos, para melhor compreensão deste trabalho, alguns periféricos do computador digital PDP 11/40 sobre o aspecto de programação, assim como os modos de cálculo, endereçamento e algumas placas de interface.

## 2.2 - Conversor Analógico Digital (LPSAD)

O conversor analógico digital situa-se no LPS 11-S (laboratory peripheral system) sendo constituido por um registro de estado e um registro de armazenamento(2).

### 2.2.1 - Registro de Estado do Conversor Analógico Digital

O registro de estado tem como endereço 170400, cuja configuração de bits é como mostra a Fig. 2.1.

| ERR FLAG | NÃO USADOS | CANAIS DE CONVERSÃO | DONE FLAG | INT ENABLE | CLOCK OVERFLOW | SCHMITH TRIGGER | NÃO USADOS | A/D |
|---|---|---|---|---|---|---|---|---|
| 15 | 14 11 | 10 8 | 7 | 6 | 5 | 4 | 3 1 | 0 |

FIGURA 2.1

Nesta figura o bit 15 é o bit de erro. Este bit quando a 1-lógico, indica um erro que pode ocorrer por qualquer dos seguintes motivos:

a) Uma segunda conversão analógica-digital terminou antes que os dados da primeira conversão analógica-digital sejam lidos;

b) O multiplexer mudou durante o primeiro microsegundo da conversão;

c) Uma segunda conversão analógica-digital é iniciada antes que a primeira seja completada.

Os bits de 10 a 8, definem quais os canais que estão sendo utilizados, conforme indica a Tabela 2.1.

| CANAL | CONFIGURAÇÃO BINÁRIA DOS BITS | | |
|---|---|---|---|
| | 10 | 9 | 8 |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 |
| 5 | 1 | 0 | 1 |
| 6 | 1 | 1 | 0 |
| 7 | 1 | 1 | 1 |

TABELA 2.1 - Configuração do bits para a seleção de canais.

O bit 7, quando a 1-lógico, indica que uma conversão analógica digital foi feita. É colocado a zero quando uma interrupção acabou ou quando o registro de dados é lido.

O bit 6, quando a 1-lógico, fará com que o bit 7 cause uma interrupção sempre que uma conversão estiver completa.

O bit 5, quando a 1-lógico, permite que um estouro(overflow) do real time clock reinicie uma conversão analógica digital. Isto permite amonstragens nos canais de zero a sete em intervalos de tempo precisos independentes de software.

O bit 4, quando a 1-lógico, permite que um evento externo dispare o schmitt trigger o qual, por sua vez, inicia uma conversão analógica digital.

O bit Ø, quando a 1-lógico, inicia uma conversão analógica digital. Ele é zerado no fim da conversão.

### 2.2.2 - Registro de Armazenamento do Conversor Analógico-Digital.

O registro de armazenamento do conversor analógico digital, tem por endereço 170402. Sua função é armazenar os valores da conversão A/D em 12 bits ajustados a direita em base octal, conforme mostra a tabela 2.2 e a configuração de bits na Fig. 2.2.

A faixa de tensão de conversão para os canais 4 a 7, está entre Ø e 10 volts. Por isso, a tabela 2.2 apresenta colunas distintas para cada um destes grupos de canais.

| OCTAL | DECIMAL | CANAIS 0/-3 | CANAIS 4-7 |
|---|---|---|---|
| 7777 | 4096 | +1.9995volts | 9.9975 volts |
| 4000 | 2048 | 1.000 volts | 5.000 volts |
| 0000 | 0000 | 0.000 volts | 0.000 volts |

TABELA 2.2 -Faixa de tensão de conversão

| NÃO USADOS | | Armazenamento das conversões | |
|---|---|---|---|
| 15 | 12 | 11 | 0 |

FIGURA 2.2 - Registro de Armazenamento do LPSAD

## 2.3 - Real Time Clock (LPSKW)

O real time clock fica situado no LPS 11-S apresentando registro de estado, registro de armazenamento e registro contador(2).

### 2.3.1 - Registro de Estado do LPSKW

Este registro tem como endereço 170404, apresentando uma configuração de bits conforme mostra a Figura 2.3.

| NÃO UTILIZADOS | | MODO | | INIBE O MODO | INTER-RUPÇÃO | NÃO USADOS | | SELEÇÃO DE FREQUÊNCIA | | INICIA CONTAGEM |
|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 1 | 0 |

FIGURA 2.3 - Configuração de bits do registro de estado do LPSKW

Os bits 9-8 da Fig. 2.3, têm por função a escolha do modo de operação do contador conforme mostra a Tabela 2.3.

| BITS 9 8 | F U N Ç Ã O |
|---|---|
| 0 0 | Modo intervalo único. O contador conta do valor estabelecido até que haja um estouro, então o bit 7 vai a 1 lógico e a contagem é interrompida. |
| 0 1 | Modo intervalo repetido. O contador conta do valor estabelecido até que haja um estouro(overflow), coloca o bit 7 a 1, transfere o conteudo do registro buffer para o contador e inicia novamente um novo ciclo de contagem. |
| 1 0 | Um pulso do ST2 transfere o conteudo do contador para o buffer, coloca o bit 7 a 1 e continua contando. |
| 1 1 | Um pulso do ST2 transfere o conteudo do contador para o buffer e coloca o bit 7 a 1, zera o contador e a contagem inicia de zero |

TABELA 2.3 - Modos de operação do Contador.

O bit 7 é posto a 1 por um dos modos de operação descritos na Tabela 2.3.

Os bits de 3 a 1 têm por função o controle de frequência básica, conforme mostra a Tabela 4.

O bit zero tem por função permitir que o contador conte em uma taxa especificada da frequência básica. Pode ser levado a 1 pelo Schmitt Trigger 1 e é zerado por overflow no modo 1.

| BITS 03 | 02 | 01 | FUNÇÃO |
|---|---|---|---|
| 0 | 0 | 0 | Nenhuma frequência |
| 0 | 0 | 1 | 1 MHz |
| 0 | 1 | 0 | 100 KHz |
| 0 | 1 | 1 | 10 KHz |
| 1 | 0 | 0 | 1 KHz |
| 1 | 0 | 1 | 100 Hz |
| 1 | 1 | 0 | S T 1 |
| 1 | 1 | 1 | Frequência da linha |

TABELA 2.4 - Seleção de Frequências

### 2.3.2 - Registro de Armazenamento e Registro Contador do LPSKW

O registro contador não pode ser carregado diretamente pelo processador mas, sim, pelo registro de armazenamento de endereço 170406. Carregando-se o registro de aramazenamento do LPSKW, quando o bit zero do registro de estado do LPSKW está a 0-lógico,o registro contador também será carregado.

## 2.4 - Conversores Digitais-Analógicos.

O PDP 11-40 apresenta quatro conversores digitais - Analógicos que são responsáveis pela transmissão de dados para o computador analógico. Cada conversor produz uma tensão de saída que se situa na faixa de ± 10 volts. Desses quatro, dois estão situados no LPS 11, tendo registros com endereços 170420(A625-Canal 2) e 170422 (A625-Canal 3), cujos valores convertidos equivalentes são: +4096 para 10 volts e -4096 para -10 volts. Os outros dois conversores situam-se na CPU tendo por endereços 176760(A614-Canal 0) e 176762 (A614-Canal 1), possuindo uma faixa de operação de saída de 2048 correspondente a +10 volts e -2048 a -10 volts.

A figura 2.4 ilustra o esquema de ligação dos canais conversores no painel do computador analógico quando este estiver operando em qualquer dos modos de cálculo que serão descritos posteriormente.

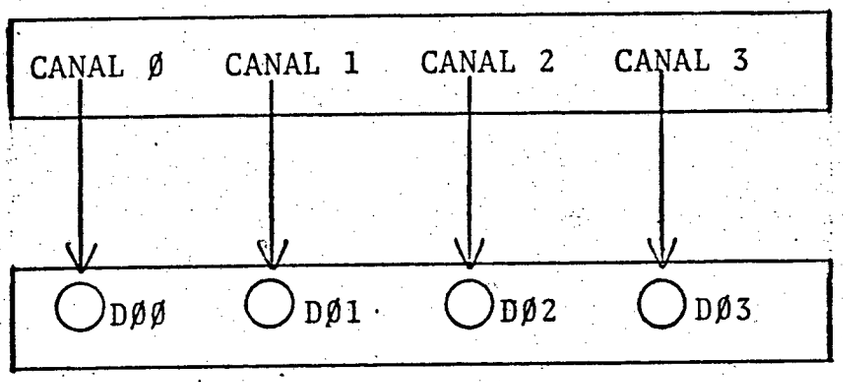



FIGURA 2.4 - Esquema de ligação dos Canais D/A

Deve-se enfatizar que o canal Ø pode ser usado como valor de tensão de referência, na colocação de valores de ajuste de servopotenciômetros, quando o computador analógico estiver no modo de ajuste de servopotenciômetro (1).

## 2.5 - Tratamento dos modos de cálculo e endereçamento do analógico através do PDP 11-40.

Através do kit 11-H M1502 com endereço 164ØØØ situado no computador digital, faz-se controle de cálculo e endereçamento no computador analógico como descrito nos dois parágrafos seguintes.

### 2.5.1 - Modos de cálculo

Utilizando-se 7 bits da palavra 164ØØØ, conforme Fig. 2.5, envia-se sinais de comando de cálculo e endereçamento para o computador analógico (1).

| ENDEREÇAMENTO DAS DEZENAS | | ENDEREÇAMENTO DE UNIDADES | | MODOS DE CÁLCULO | | COMANDO | | PT | POT | P/V | H |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 12 | 11 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

FIGURA 2.5 - Sinais enviados pelo PDP 11-40

Os bits 6 e 7 da Fig. 2.5 selecionam os modos de cálculo conforme Tabela 2.5

| MODOS | BITS | |
|---|---|---|
| | 7 | 6 |
| DAUER | 0 | 0 |
| REPET | 1 | 0 |
| MIT HALT | 0 | 1 |
| IT HAND | 1 | 1 |

TABELA 2.5 - Modos de cálculo.

DAUER: É o modo no qual o computador analógico calcula indefinidamente, sem interrupção.

REPET: É o modo no qual o computador analógico realiza um ciclo de cálculo completo (PAUSA, CÁLCULO e HALT) durante um tempo selecionável pelo relógio da unidade de controle do computador analógico (DBG 771) e continuará fazendo novos ciclos de cálculo, a menos que se force uma parada através dos sinais de comando PAUSE ou HALT.

MIT-HALT: O computador analógico realiza um ciclo completo de cálculo (PAUSA, CÁLCULO e HALT),durante o tempo selecionável pelo relógio da unidade de controle e permanece em HALT, só retomando novamente outro ciclo completo através do comando WEITER.

IT-HAND: O computador analógico realiza um ciclo iterativo completo. Compreende-se por ciclo iterativo completo, dois ciclos de cálculo distinto para integradores selecionáveis no painel de programação analógica do Telefunken, com a característica de que para passar de um ciclo de cálculo para outro, é necessário realizar o comando WEITER:

Em todos os modos descritos acima é necessário que o computador esteja operando em RECHNEN.

Bits 4 a 5 - Selecionam os comandos, conforme mostra a Tabela 6.

| COMANDO | BITS | |
|---|---|---|
| | 5 | 4 |
| RECHNEN | 1 | x |
| PAUSE | 0 | 0 |
| HALT | 0 | 1 |
| WEITER | 1 | ↑ |

TABELA 6 - Bits de sinais de comandos.

Na Tabela 6 o símbolo "x" significa tanto faz se o nível do sinal é zero ou um e, "↑", significa que houve uma transição do estado zero para o estado um.

RECHNEN: É o modo de cálculo do computador analógico. Quando este comando é acionado os integradores são retirados da sua condição e levados a calcular.

PAUSE: É o modo de repouso ou condição inicial.

HALT: É o modo no qual o computador analógico reinicia um novo ciclo de cálculo quando no modo MIT-HALT ou passa para um novo ciclo de iteração quando no modo IT-HAND.

Em qualquer dos comandos descritos acima, o bit 3 deve permanecer sempre no estado zero (1).

Os bits 3, 2 e 1 selecionam os sinais pt, POT e P/V, conforme mostra a Tabela 7.

| SINAIS | BITS | | |
|---|---|---|---|
| | 3 | 2 | 1 |
| pt | 1 | 0 | 0 |
| POT | 1 | ↑ | 1 |
| P/V | 0 | 0 | 1/0 |

TABELA 7 - Seleção de sinais.

BIT 3 - Sinal pt - Corresponde a tecla Pot na unidade de controle do Telefunken. Este modo é utilizado para colocação de valores nos servopotenciômetros e para verificação de valores nos potenciômetros de ajuste manual.

BIT 2 - Sinal Pot - Quando os bits 3 e 1 estão a um lógico e o bit 2 sofre uma transição do estado zero ao estado um lógico, será ativado o funcionamento do mecanismo de ajuste servopotenciômetros.

BIT 1 - Sinal P/V - Quando a um lógico seleciona amplificadores e quando a zero lógico seleciona potenciômetros, substituindo as teclas P e V na unidade de endereçamento do computador analógico.

### 2.5.2 - Seleção de Endereços

O endereçamento de amplificadores e potenciômetros no RA770 foi feito usando-se o kit 11-H M1502 as placas seletoras de endereço e as placas de linhas de endereço(1). Para a seleção de endereços, usou-se nove bits da palavra 164000, conforme Fig. 2.5.

O bit 0 (H) - Este bit quando sofre uma transição de 0-lógico para 1-lógico, permite que seja escolhido um novo endereço no computador analógico.

Os bits 8 a 11 servem para selecionar as unidades na escolha de um endereço no computador analógico.A configuração de bits é feita segundo a Tabela 8. Deve-se notar que esta configuração é BCD excesso seis.

Os bits de 12 a 15 servem para selecionar as dezenas na escolha de um endereço no analógico. Da mesma forma que para o endereçamento de unidades, esta configuração é BCD excesso seis, conforme Tabela 2.8

| UNIDADES OU DEZENAS | 15 ou 11 | 14 ou 10 | 13 ou 9 | 12 ou 8 | EQUIVALENTE BCD |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 6 |
| 1 | 0 | 1 | 1 | 1 | 7 |
| 2 | 1 | 0 | 0 | 0 | 8 |
| 3 | 1 | 0 | 0 | 1 | 9 |
| 4 | 1 | 0 | 1 | 0 | 10 |
| 5 | 1 | 0 | 1 | 1 | 11 |
| 6 | 1 | 1 | 0 | 0 | 12 |
| 7 | 1 | 1 | 0 | 1 | 13 |
| 8 | 1 | 1 | 1 | 0 | 14 |
| 9 | 1 | 1 | 1 | 1 | 15 |

TABELA 2.8 - Configuração dos bits de selecionamento de endereços.

## 2.6 - Utilização da Palavra de Deteção

O circuito de deteção utiliza a palavra de deteção ( Kit 11-H M1501 ) do PDP 11-40 para receber de volta os comandos, endereços e sinais enviados ao computador analógico(1).A esta palavra é atribuída o endereço 164Ø1Ø com configuração de bits,conforme a Fig. 2.6.

| ⟷ | ⟷ | P/V | Tp | TR1 | TH1 | TR2 | TH2 | POTNF | POTF |
|---|---|---|---|---|---|---|---|---|---|
| 15 12 | 11 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

FIGURA 2.6 - Palavra de deteção do PDP 11-40

Os bits 15 a 8 fornecem uma réplica do endereço atingido pelo computador analógico. Deve-se ressaltar que a verificação do endereço atingido no computador analógico não serve para certificar o PDP 11-40 de que o endereço já foi atingido, pois que as mensagens de volta de endereçamento no analógico são retiradas ainda da parte lógica que usa semicondutores. No entanto, esses semicondutores ainda vão chavear os relés de endereçamento que ainda precisam de um certo tempo para se acomodar(1). A estimativa para este tempo, é de 18 m s.

O bit 7 quando a 1-lógico, indica que o endereçamento é de potenciômetro e de amplificador se a 0-lógico.

O bit 6 quando a 1-lógico, indica que o estado de cálculo é PAUSE, qualquer que seja o ciclo de iteração.

O bit 5 quando a 1-lógico, indica que o computador analógico está no modo de cálculo (RECHNEN) do primeiro ciclo de iteração.

O bit 4 quando a 1-lógico, indica o modo HALT do primeiro ciclo de iteração.

O bit 3 quando a 1-lógico, indica o modo de cálculo (RECHNEN) do segundo ciclo de iteração.

O bit 2 quando a 1-lógico, indica o modo HALT do segundo ciclo de iteração.

O bit 1 quando a 0-lógico, indica que o computador analógico não conseguiu ajustar um determinado potenciômetro.

O bit 0 quando a 0-lógico, indica que o ajuste de servopotenciômetro terminou.

## 2.7 - Condicionadores de Sinais

Desde que os conversores analógicos digitais só admitem como entrada tensões que variam entre 0 volts e 10 volts e entre 0 volts e 2 volts, conforme Tabela 2.2, foram criados condicionadores de sinais para compatibilizar os níveis de tensão de entrada dos conversores com o nível de tensão de saída do computador analógico, que varia entre -10 volts e +10 volts.

## 2.8 - Configuração do Sistema Híbrido.

Além dos periféricos descritos neste Capítulo, apresentaremos nas Tabelas 2.9 e 2.10, outros periféricos utilizados na configuração híbrida.

| COMPUTADOR ANALÓGICO |
|---|
| Um Plotter Hewlett Packard |
| Doze Integradores / Somadores |
| Quatro Multiplicadores |
| Dezesseis Somadores / Inversores |
| Dois Geradores de Seno |
| Dois Geradores de Coseno |
| Quatro Geradores de Função ou Inversores |
| Dezesseis Potenciômetros de ajuste manual |
| Vinte Servopotenciômetros |
| Um Voltímetro Digital |

TABELA 2.9 - Relação dos componentes do Computador Analógico.

| COMPUTADOR DIGITAL PDP 11-40 |
|---|
| Software em fita de papel |
| Basic como linguagem de altonível |
| Leitora / perfuradora de fita de papel |
| Terminal decwriter |
| 16 K palavra de Memória |
| Uma unidade aritimética de ponto flutuante |
| Uma unidade aritimética que opera com números inteiros |

TABELA 2.10 - Relação atual dos componentes do PDP 11-40

Deve-se ressaltar que está havendo uma expansão no computador digital, conforme Tabela 2.11.

| |
|---|
| Uma teleimpressora |
| Dual Floppy Disk |
| 8 K de Memória |
| Um terminal de vídeo de caracteres alfanuméricos |

TABELA 2.11 - Relação dos componentes adicionais do Computador PDP 11-40

## 2.9 - Conclusão

Ve-se, assim, a possibilidade de tratar-se os sinais de saída do computador analógico, converter os sinais digitais em analógico, controlar os modos de cálculo, endereçamento de potenciômetro e amplificadores, por intermédio do computador digital, tornando, portanto, o computador analógico um periférico de entrada e saída do computador digital possibilitando a construção de um computador híbrido.

A expansão apresentada na Tabela 2.11, possibilita o armazenamento de um maior número de dados, tornando, assim, a configuração híbrida mais versátil.

No próximo capítulo, serão apresentadas as subrotinas que

compõem a linguagem que faz o tratamento dos sinais entre os dois computadores.

# CAPÍTULO III - DESCRIÇÃO DA LINGUAGEM HÍBRIDA

# 3 - DESCRIÇÃO DA LINGUAGEM HÍBRIDA

## 3.1 - Introdução

No presente Capítulo estão descritos os programas que compõem a linguagem híbrida desenvolvida neste trabalho. Esta linguagem faz o tratamento dos dados transmitidos entre o computador analógico RA 770 e o minicomputador PDP 11-40,utilizando periféricos deste e os circuitos das placas de interface, ambos descritos no Capítulo II. A linguagem compõe-se de cinco subrotinas,nominalmente: KAD; DAC; ENDE; CALC; POT, as quais foram feitas em linguagem Assembler e embutidas numa linguagem de alto nível,o BASIC do PDP 11-40. Para cada uma destas subrotinas, apresenta-se neste Capítulo uma descrição de sua função, um algorítmo, um fluxograma e a listagem correspondente.

## 3.2 - Descrição da Subrotina KAD

Os sinais de saída do computador analógico atingem os canais de conversão analógica digital, através dos circuitos condicionadores. A subrotina KAD tem a função de converter os sinais analógicos para digitais, com amostragens tomadas em uma determinada frequência.

O acesso a esta subrotina é feito através da declaração CALL "KAD" (X1, X2, X3).

Onde a variável:

X1 = qualquer um dos oito canais do conversor A/D
X2 = é a frequência de amostragem
X3 = variável convertida.

### 3.2.1 - Algorítmo para a Subrotina KAD

Descreveremos neste item o algorítmo desenvolvido para a subrotina "KAD". Cada número representa um passo da subrotina.

(1) Verificar o parêntese a esquerda;
(2) Evoluir o primeiro número depois do parântese a esquerda e colocá-lo no armazenador de pt flutuante (FAC);

(3) Transformar o conteúdo do FAC em inteiro e depositá-lo em FAC2;

(4) Mover a variável X1 armazenada em FAC2, para o byte superior do registro de estado do conversor analógico digital, selecionando, desta maneira, o canal com o qual se quer trabalhar;

(5) Verificar a vírgula de separação de variáveis;

(6) Evoluir a segunda variável e colocá-la no FAC;

(7) Transformar o conteúdo do FAC em inteiro e depositá-lo em FAC2;

(8) Mudar o conteúdo do FAC2 por seu complemento a dois;

(9) Mover o conteúdo do FAC2 para o buffer do LPSKW, permitindo a escolha da frequência múltipla da frequência básica;

(10) Verificar a vírgula que separa as variáveis;

(11) Determinar o número de amostragens que devem ser feitas. Somar um a esta variável;

(12) Preparar o endereço de memória que armazenará o resultado das conversões analógicas-digitais;

(13) Verificar o parêntese a direita;

(14) Verificar o fim de linha;

(15) Mover para o registro de estado do LPSKW um número que permita a escolha de uma frequência básica e um modo de contagem;

(16) Comandar as amostragens pelo real time clock;

(17) Verificar se houve conversão;

(18) Se houve, passar para a próxima declaração. Se não, esperar que haja;

(19) Armazenar o resultado da conversão;

(20) Decrementar o número de conversões;

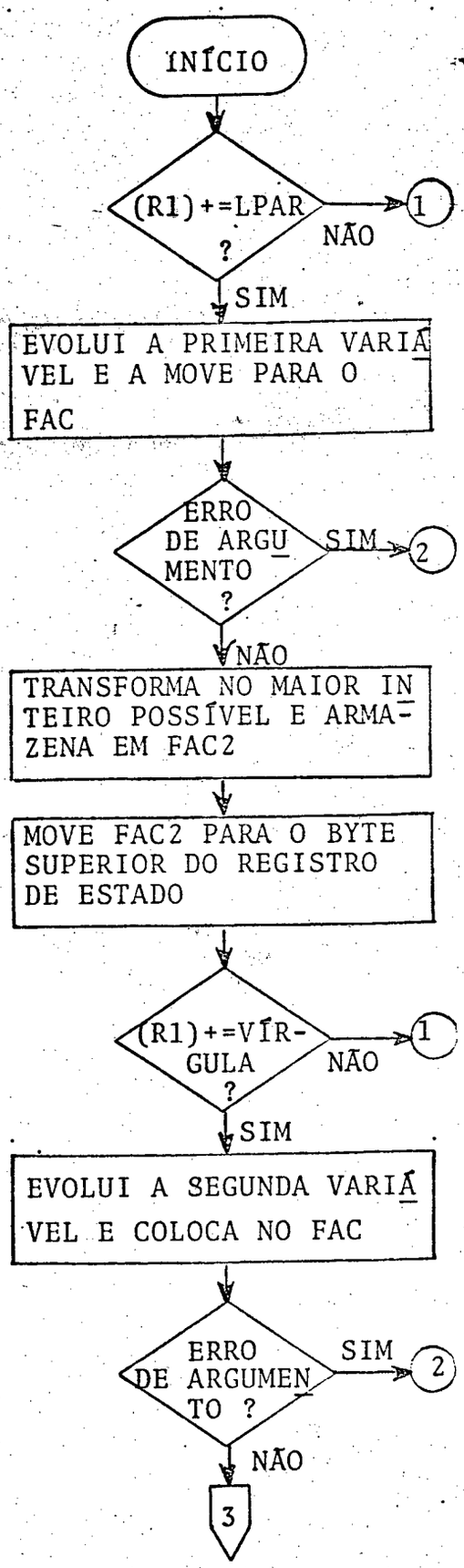
(21) Verificar se as conversões já terminaram;

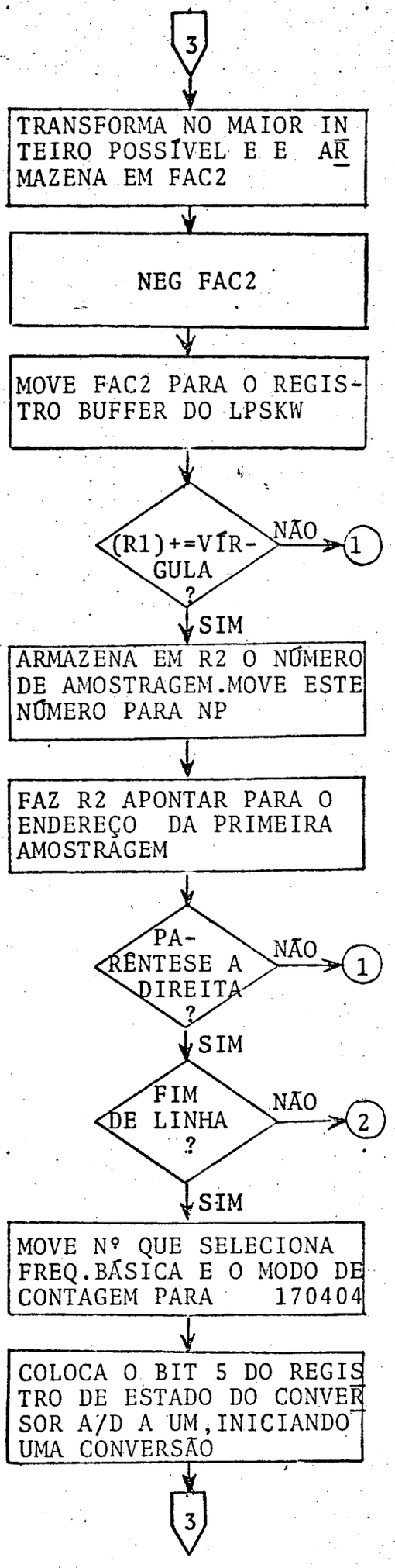
(22) Se não, voltar a fazer conversão;

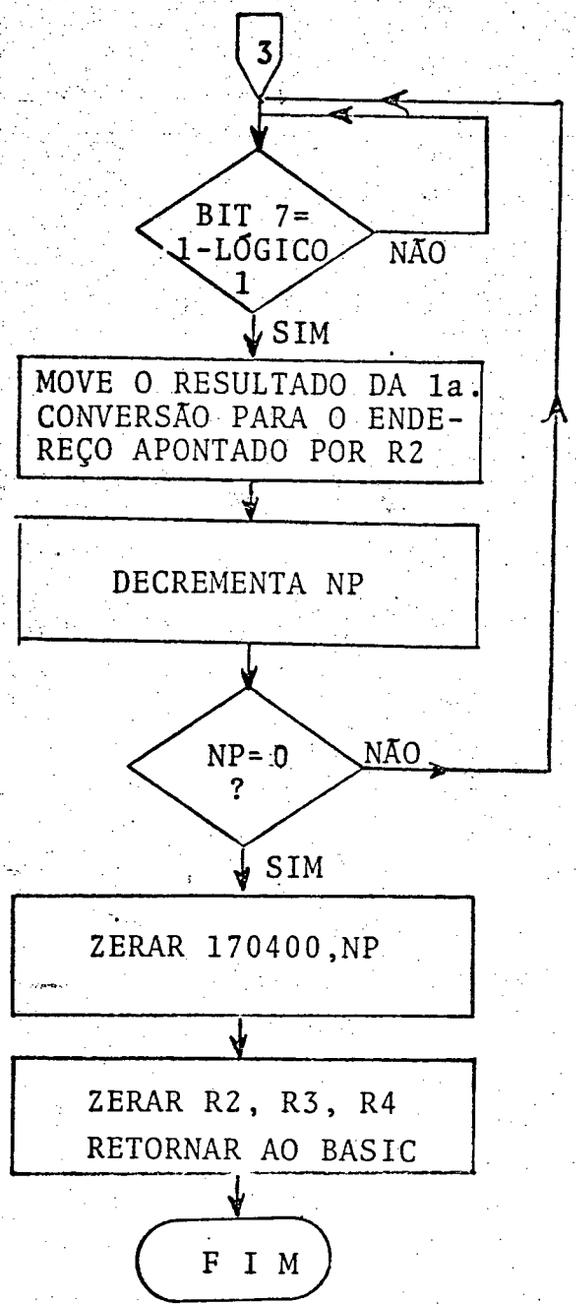
(23) Se foi alcançado, retornar ao BASIC.

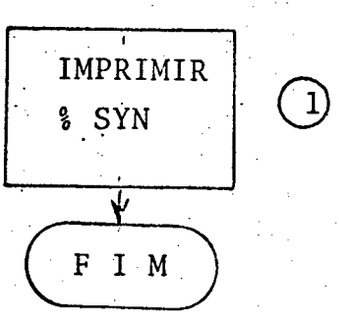
(15) Mover o conteúdo de R3 para 164ØØØ;
(16) Retornar ao BASIC.

### 3.2.2 - Fluxograma da Subrotina KAD

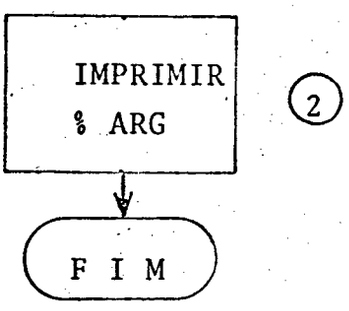



3
TRANSFORMA NO MAIOR INTEIRO POSSÍVEL E E ARMAZENA EM FAC2
NEG FAC2
MOVE FAC2 PARA O REGISTRO BUFFER DO LPSKW
(R1)+=VÍRGULA ?
NÃO
1
SIM
ARMAZENA EM R2 O NÚMERO DE AMOSTRAGEM.MOVE ESTE NÚMERO PARA NP
FAZ R2 APONTAR PARA O ENDEREÇO DA PRIMEIRA AMOSTRAGEM
PARÊNTESE A DIREITA ?
NÃO
1
SIM
FIM DE LINHA ?
NÃO
2
SIM
MOVE Nº QUE SELECIONA FREQ.BÁSICA E O MODO DE CONTAGEM PARA 170404
COLOCA O BIT 5 DO REGISTRO DE ESTADO DO CONVERSOR A/D A UM, INICIANDO UMA CONVERSÃO
3


3
BIT 7= 1-LÓGICO 1
NÃO
SIM
MOVE O RESULTADO DA 1a. CONVERSÃO PARA O ENDEREÇO APONTADO POR R2
DECREMENTA NP
NP= 0 ?
NÃO
SIM
ZERAR 170400,NP
ZERAR R2, R3, R4
RETORNAR AO BASIC
F I M
A


IMPRIMIR % SYN
F I M

(1)


IMPRIMIR % ARG
F I M

(2)

### 3.2.3 - Listagem da Subrotina KAD

```
                                .TITLE  USRFNS
                                .GLOBL KAD,DAC,CALC,ENDE,POT
                                .GLOBL .COMMA,.RPAR,EVAL,GETVAR,STOVAR
                                .GLOBL INT,ERRSYN,FAC1,FAC2,ERRARG
                                .GLOBL .LPAR,.EOL
                 000000         .CSECT
                 000000         R0=%0
                 000001         R1=%1
                 000002         R2=%2
                 000003         R3=%3
                 000004         R4=%4
                 000005         R5=%5
                 000006         R6=%6
                 000007         R7=%7
 000000 122127'KAD:             CMPB (R1)+,#.LPAR
        000000
 000004 001115                  BNE ESYN
 000006 004767'                 JSR R7,EVAL
        000000
 000012 103514                  BCS EARG
 000014 004767'                 JSR R7,INT
        000000
 000020 116537'                 MOVB FAC2(R5),@#170401
        000000
        170401
 000026 122127'                 CMPB(R1)+,#.COMMA
        000000
 000032 001102                  BNE ESYN
 000034 004767'                 JSR R7,EVAL
        000000
 000040 103501                  BCS EARG
 000042 004767'                 JSR R7,INT
        000000
 000046 005465'                 NEG FAC2(R5)
        000000
 000052 016537'                 MOV FAC2(R5),@#170406
        000000
        170406
 000060 122127'                 CMPB (R1)+,#.COMMA
        000000
 000064 001065                  BNE ESYN
 000066 112102                  MOVB (R1)+,R2
 000070 100463                  BMI ESYN
 000072 000302                  SWAB R2
 000074 152102                  BISB (R1)+,R2
 000076 061502                  ADD (R5),R2
 000100 062702                  ADD #4,R2
        000004
 000104 011267                  MOV (R2),NP
        000126
 000110 162702                  SUB #2,R2
        000002
 000114 011292                  MOV (R2),R2
 000116 062702                  ADD #6,R2
        000006
 000122 062767                  ADD #1,NP
        000001
        000106
```

DIGITAL EQUIPMENT COM. IND. LTDA.

DIGITAL EQUIPMENT COM. IND. LTDA.

```
                                        PAGE    001

000130 122127'          CMPB (R1)+,#.RPAR
       000000
000134 001041           BNE ESYN
000136 122127'          CMPB (R1)+,#.EOL
       000000
000142 001036           BNE ESYN
000144 012737           MOV #407,@#170404
       000407
       170404
000152 062737           ADD #40,@#170400
       000040
       170400
000160 105737 CONV:     TSTB @#170400.
       170400
000164 100375           BPL CONV
000166 042737           BIC #200,@#170404
       000200
       170404
000174 013722           MOV @#170402,(R2)+
       170402
000200 005722           TST (R2)+
000202 005367           DEC NP
       000030
000206 026727           CMP NP,#0
       000024
       000000
000214 001361           BNE CONV
000216 005037           CLR @#170400
       170400
000222 005067           CLR NP
       000010
000226 005002           CLR R2
000230 005003           CLR R3
000232 005004           CLR R4
000234 000207 FIM:      RTS R7
000236 000000 NP:       .WORD 0
000240 000167'ESYN:     JMP ERRSYN
       000000
000244 000167'EARG:     JMP ERRARG
       000000
```

## 3.3 - Descrição da Subrotina DAC

A função desta subrotina é selecionar um determinado canal D/A, através do PDP 11-40 e entregar o resultado da conversão D/A ao computador analógico. O acesso a esta subrotina é feito através da declaração:

CALL "DAC" (X1, X2)

X1,representa qualquer um dos quatro canais de conversão digital analógico.

X2,representa o valor que nós queremos converter.

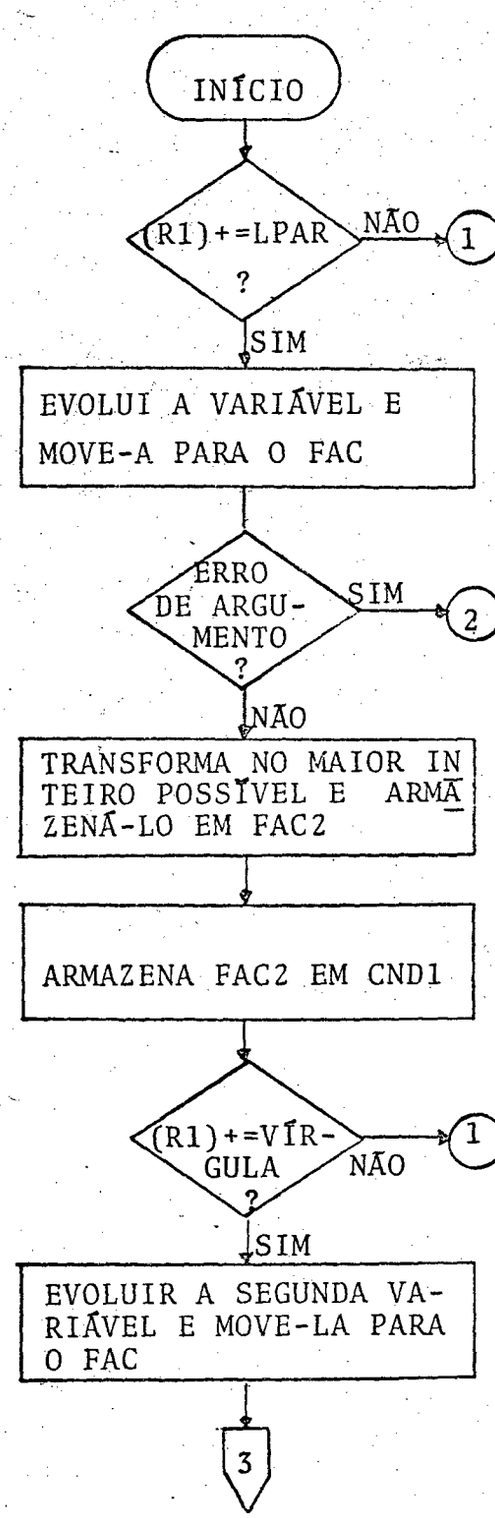
### 3.3.1 - Algorítmo para a Subrotina DAC

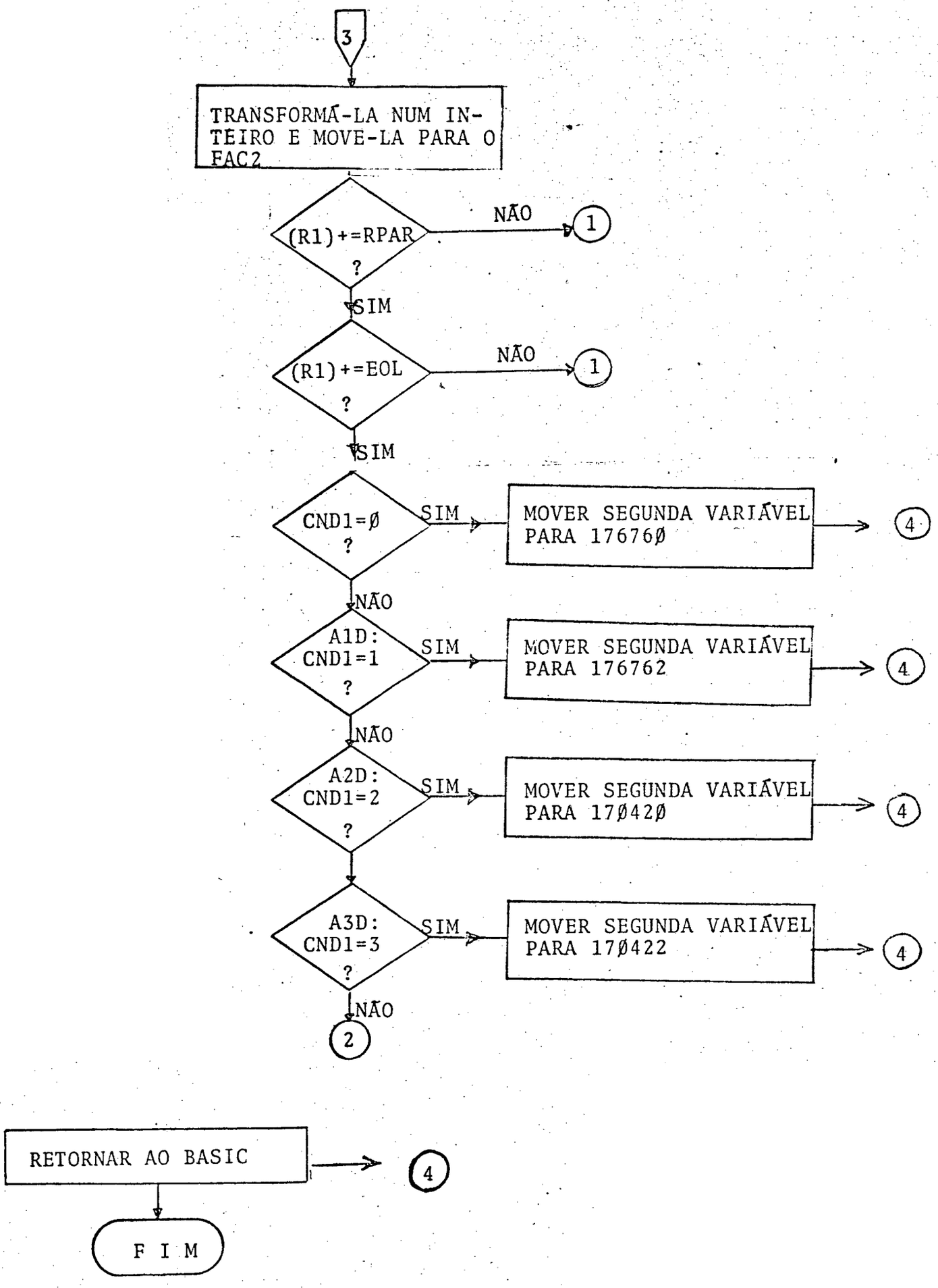
(1) Verificar parênteses a esquerda;
(2) Evoluir a primeira variável e colocá-la no armazenador de ponto flutuante (FAC):
(3) Tornar o conteúdo de FAC um número inteiro e depositá-lo em FAC2;
(4) Armazenar o número do canal em uma palavra (CND1);
(5) Verificar a vírgula que separa as variáveis;
(6) Evoluir a segunda variável e colocá-la no armazenador de ponto flutuante;
(7) Verificar o parênteses a direita;
(8) Verificar fim de linha;
(9) Transformar o conteúdo do FAC em um número inteiro e depositá-lo em FAC2;
(10) Comparar CND1 com canal Ø;
(11) Se for o canal Ø, mover a variável para 17676Ø e retornar ao BASIC;
(12) Caso não seja o canal Ø, entrar na subrotina A1D;
(13) Verificar se é o canal 1;
(14) Se for, mover a variável para 176762 e retornar ao BASIC;
(15) Se não for, desviar para a subrotina A2D;
(16) Comparar CND1 com canal 2;
(17) Se for, mover a variável para 17Ø42Ø e retornar ao BASIC;
(18) Se não for, desviar para a subrotina A3D;
(19) Verificar se é o canal 3;

(20) Se for, mover a variável para 17Ø422 e retornar ao BASIC;

(21) Se não for, vai para a subrotina de erro de argumento.

### 3.3.2 - Fluxograma da Subrotina DAC




3
TRANSFORMÁ-LA NUM INTEIRO E MOVE-LA PARA O FAC2
(R1)+=RPAR ?
NÃO
1
SIM
(R1)+=EOL ?
NÃO
1
SIM
CND1=Ø ?
SIM
MOVER SEGUNDA VARIÁVEL PARA 176760
4
NÃO
A1D: CND1=1 ?
SIM
MOVER SEGUNDA VARIÁVEL PARA 176762
4
NÃO
A2D: CND1=2 ?
SIM
MOVER SEGUNDA VARIÁVEL PARA 170420
4
A3D: CND1=3 ?
SIM
MOVER SEGUNDA VARIÁVEL PARA 170422
4
NÃO
2
RETORNAR AO BASIC
4
F I M

## 3.3.3 - Listagem da Subrotina DAC

```
000250 122127'DAC:      CMPB     (%1)+,#.LPAR
       000000
000254 001371           BNE      ESYN
000256 004767'          JSR      %7,EVAL
       000000
000262 103770           BCS      EARG
000264 004767'          JSR      %7,INT
       000000
000270 016567'          MOV      FAC2(%5),CND1
       000000
       000134
000276 122127'          CMPB     (%1)+,#.COMMA
       000000
000302 001356           BNE      ESYN
000304 004767'          JSR      %7,EVAL
       000000

                        PAGE     002

000310.103755           BCS      EARG
000312 122127'          CMPB (R1)+,#.RPAR
       000000
000316 001350           BNE ESYN
000320 122127'          CMPB (R1)+,#.EOL
       000000
000324 001345           BNE ESYN
000326 004767'          JSR      %7,INT
       000000
000332 026727           CMP      CND1,#0
       000074
       000000
000340 001004           BNE      A1D
000342 016537'          MOV      FAC2(%5),@#176760
       000000
       176760
000350 000207           RTS      %7
000352 026727 A1D:      CMP      CND1,#1
       000054
       000001
000360 001004           BNE      A2D
000362 016537'          MOV      FAC2(%5),@#176762
       000000
       176762
000370 000207           RTS      %7
000372 026727 A2D:      CMP      CND1,#2
       000034
       000002
000400 001004           BNE      A3D
000402 016537'          MOV      FAC2(%5),@#170420
       000000
       170420
000410 000207           RTS      %7
000412 026727 A3D:      CMP CND1,#3
       000014
       000003
```

DIGITAL EQUIPMENT COM. IND. LTDA.

0.247-884-1
Biblioteca Universitária
UFSC

```
000420 001123           BNE ERRG1
000422 016537'          MOV     FAC2(%5),@#170422
       000000
       170422
000430 000207           RTS     %7
000432 000000           CND1:   .WORD   0
```

## 3.4 - Descrição da Subrotina CALC

A finalidade desta subrotina é enviar através dos bits 4 a 7 de 164ØØØ, os comandos e modos de cálculo ao computador analógico.

O acesso a esta subrotina, é feito pela declaração:

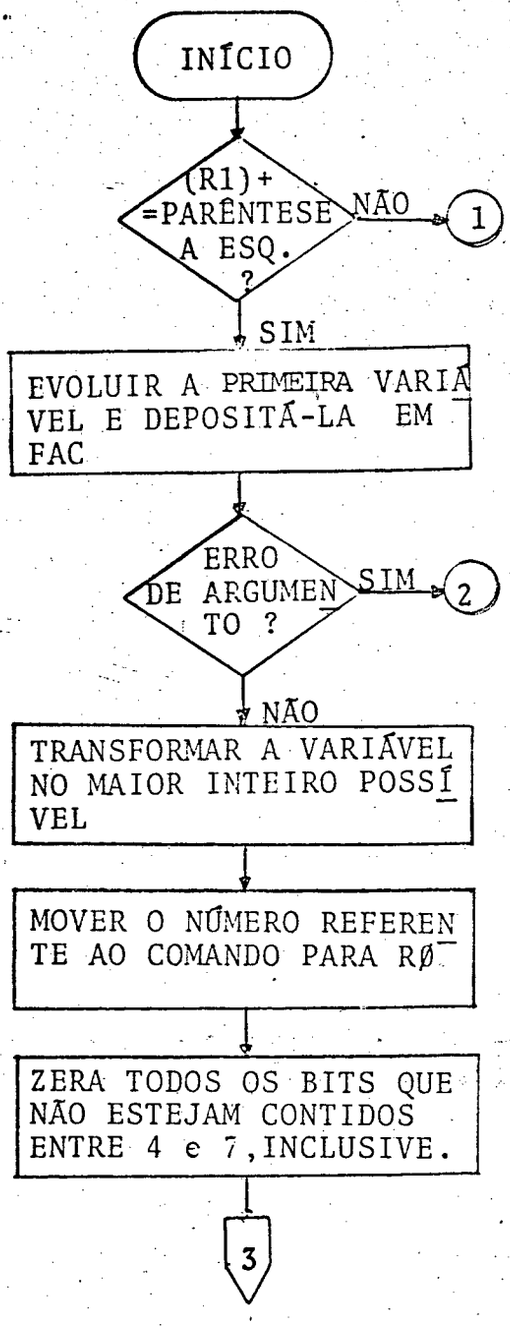
CALL "CALC" (X1)

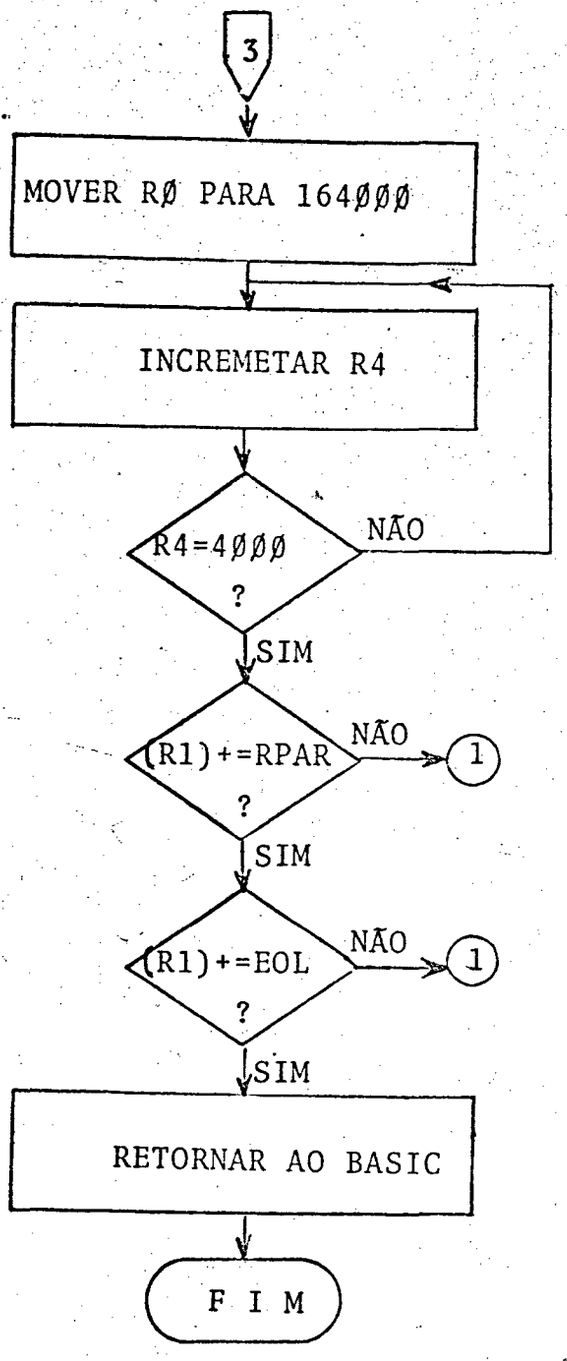
Onde X1 é um número que representa um comando e um modo de cálculo.

### 3.4.1 - Algorítmo da Subrotina CALC

(1) Comparar parêntese a esquerda;
(2) Evoluir a segunda variável e depositá-la em FAC;
(3) Transformá-la no maior inteiro possível e depositá-la em FAC2;
(4) Mover X1 para RØ;
(5) Zerar R4;
(6) Eliminar a possibilidade de endereçamento de outros bits, a não ser de 4 a 7;
(7) Mover RØ para 164ØØØ;
(8) Esperar pelo acomodamento de reles;
(9) Verificar parêntese a direita;
(10) Verificar fim de linha;
(11) Zerar R4;
(12) Retornar ao BASIC.

### 3.4.2 - Fluxograma da Subrotina CALC




3
MOVER RØ PARA 164ØØØ
INCREMETAR R4
R4=4ØØØ ?
NÃO
SIM
(R1)+=RPAR ?
NÃO
1
SIM
(R1)+=EOL ?
NÃO
1
SIM
RETORNAR AO BASIC
F I M

### 3.4.3 - Listagem da Subrotina CALC

```
000434 122127'CALC:   CMPB (%1)+,#.LPAR
       000000
000440 001111         BNE ESYN1
000442 004767'        JSR %7,EVAL
       000000
000446 103510         BCS EARG1
000450 004767'        JSR %7,INT
       000000
000454 016500'        MOV FAC2(%5),R0
       000000
000460 005004         CLR R4
000462 005003         CLR R3
000464 005002         CLR R2
000466 042700         BIC #177417,R0
       177417

                      PAGE    003

000472 010037         MOV R0,@#164000
       164000
000476 005204 VOL:    INC R4
000500 020427         CMP R4,#4000
       004000
000504 001374         BNE VOL
000506 122127'        CMPB (%1)+,#.RPAR
       000000
000512 001064         BNE ESYN1
000514 122127'        CMPB (R1)+,#.EOL
       000000
000520 001061         BNE ESYN1
000522 005004         CLR R4
000524 005002         CLR R2
000526 005003         CLR R3
000530 000207         RTS %7
```

## 3.5 - Descrição da Subrotina ENDE

A subrotina ENDE tem a função de endereçar amplificadores do computador analógico, utilizando-se das placas do interface e da palavra 164ØØØ. Para isso, deve-se fazer uma transição no bit zero desta palavra, de tal maneira que um endereço seja selecionado.

O acesso a esta subrotina é feito através da declaração:

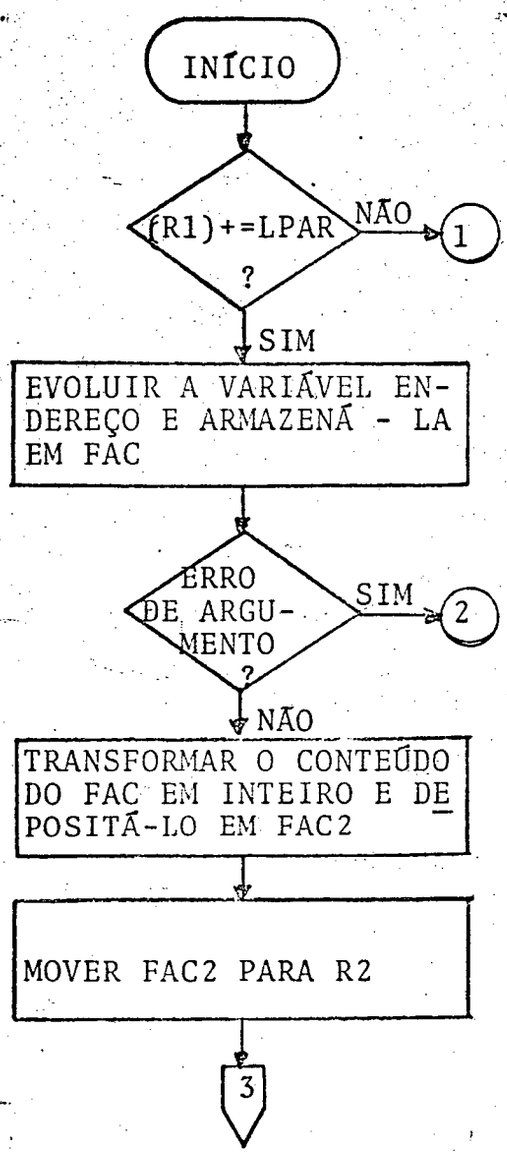
CALL "ENDE" (N)

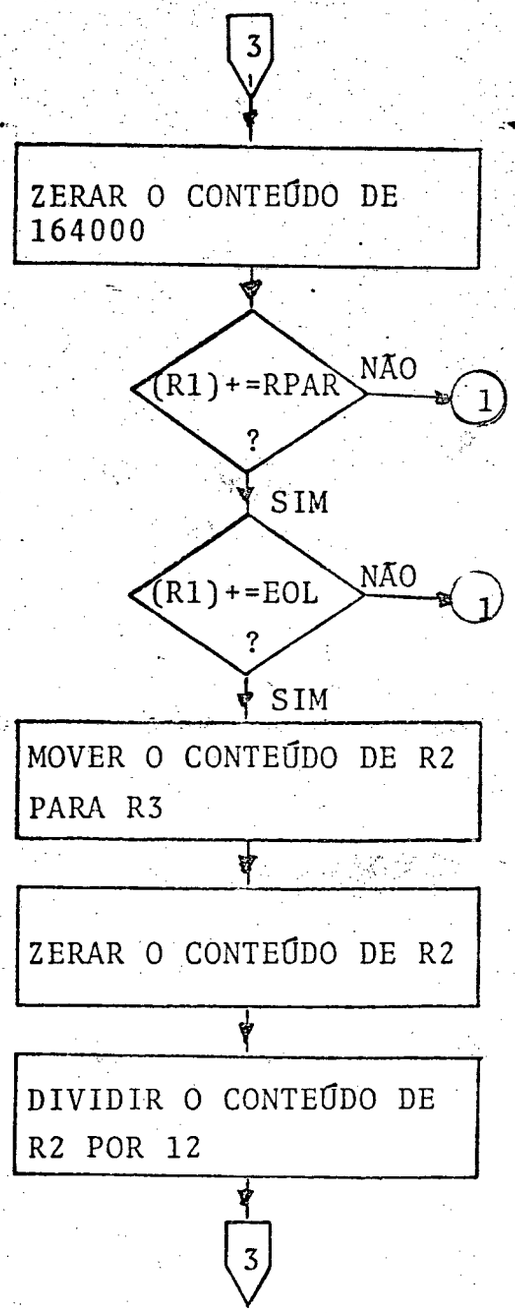
Onde N representa o endereço do amplificador que se quer trabalhar.

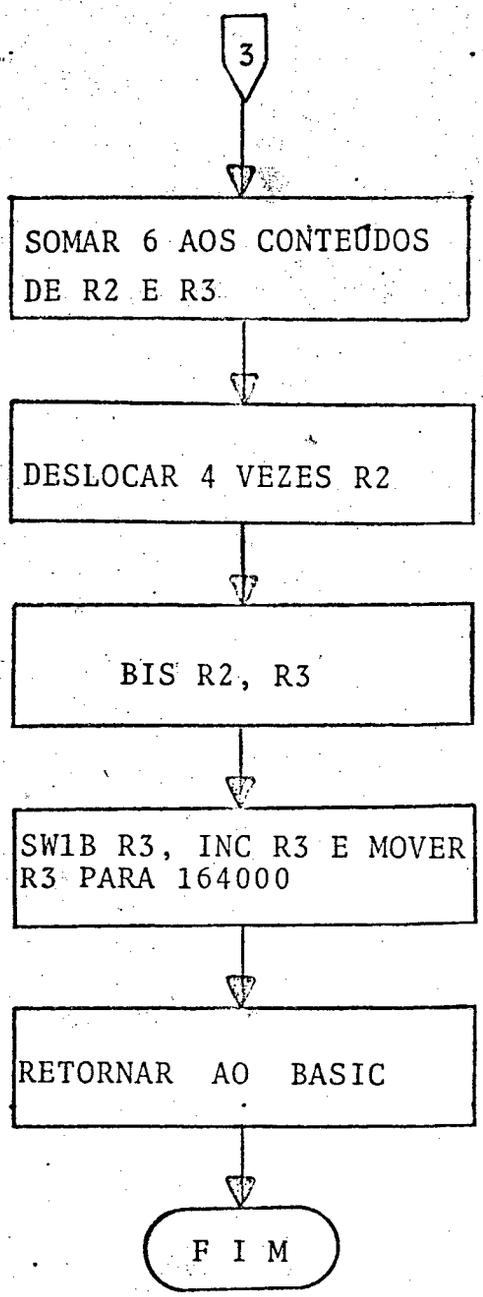
### 3.5.1 - Algorítmo da Subrotina ENDE

(1) Verificar o parêntese a esquerda;
(2) Evoluir N e depositá-lo no armazenador de ponto flutuante;
(3) Transformar o conteúdo do FAC em inteiro e depositá-lo em FAC2;
(4) Depositar o conteúdo do FAC2 em R2, obtendo assim,o endereço em octal;
(5) Zerar o conteúdo de 164ØØØ, R3, R4;
(6) Verificar o parêntese a direita;
(7) Verificar o fim da linha;
(8) Transformar o endereço que é o conteúdo de R2 em decimal, armazenando as unidades em R3 e as dezenas em R2;
(9) Criar o BCD excesso seis, para as unidades;
(10) Criar o BCD excesso seis, para as dezenas;
(11) Preparar o conteúdo de R2 para move-lo para os bits correspondentes da palavra 164ØØØ;
(12) Mover os bits que estão a 1-lógico em R2 para R3,fazendo assim com que R3 contenha o endereço do amplificador que nós queremos selecionar;
(13) Mudar o byte superior de R3 por seu byte inferior,colocando assim, os bits em correspondência com os bits de 164ØØØ;
(14) Colocar o bit 0 de R3 a 1-lógico, para que haja transição no bit 0 de 164ØØØ;

### 3.5.2 - Fluxograma da Subrotina ENDE




3
ZERAR O CONTEÚDO DE 164000
(R1)+=RPAR ?
NÃO
1
SIM
(R1)+=EOL ?
NÃO
1
SIM
MOVER O CONTEÚDO DE R2 PARA R3
ZERAR O CONTEÚDO DE R2
DIVIDIR O CONTEÚDO DE R2 POR 12
3


3
SOMAR 6 AOS CONTEÚDOS DE R2 E R3
DESLOCAR 4 VEZES R2
BIS R2, R3
SW1B R3, INC R3 E MOVER R3 PARA 164000
RETORNAR AO BASIC
F I M

## 3.5.3 - Listagem da Subrotina ENDE

```
000532 122127'ENDE:    CMPB -(%1)+,#.LPAR
       000000
000536 001052          BNE ESYN1
000540 004767'         JSR %7,EVAL
       000000
000544 103451          BCS EARG1
000546 004767'         JSR R7,INT
       000000
000552 016502'         MOV FAC2(%5),R2
       000000
000556 005037          CLR @#164000
       164000
000562 005003          CLR R3
000564 005004          CLR R4
000566 122127'         CMPB (R1)+,#.RPAR
       000000
000572 001034          BNE ESYN1
000574 122127'         CMPB (R1)+,#.EOL
       000000
000600 001031          BNE ESYN1
000602 010203          MOV R2,R3
000604 005002          CLR R2
000606 071227 OUTRA:   .WORD   DIV,12
000610 000012
000612 062702          ADD #6,R2
       000006
000616 062703          ADD #6,R3
       000006
000622 006302          ASL R2
000624 006302          ASL R2
000626 006302          ASL R2
000630 006302          ASL R2
000632 050203          BIS R2,R3
000634 000303          SWAB R3
000636 005203          INC R3
000640 010337          MOV R3,@#164000
       164000
000644 005204 IN:      INC R4
000646 020427          CMP R4,#4000
       004000

                       PAGE    004

000652 001374          BNE IN
000654 005004          CLR R4
000656 005003          CLR R3
000660 005002          CLR R2
000662 000207          RTS R7
       071227 DIV=071227
000664 000167'ESYN1:   JMP ERRSYN
       000000
000670 000167'EARG1:   JMP ERRARG
       000000
```

DIGITAL EQUIPMENT COM. IND. LTDA.

## 3.6 - Descrição da Subrotina POT

Nesta subrotina, a primeira variável utiliza-se da palavra 164ØØØ assim como das placas seletoras de endereço e placas de linha de endereço, para selecionar o endereço do potenciômetro. A segunda variável utiliza-se do canal Ø (17676Ø) para a colocação de valores nos servopotenciômetros e da palavra 164Ø1Ø para verificar se estes valores já foram alcançados.

O acesso a esta subrotina é feito através da declaração:

CALL "POT" (N,M)

Onde a variável:

N = o endereço do potenciômetro

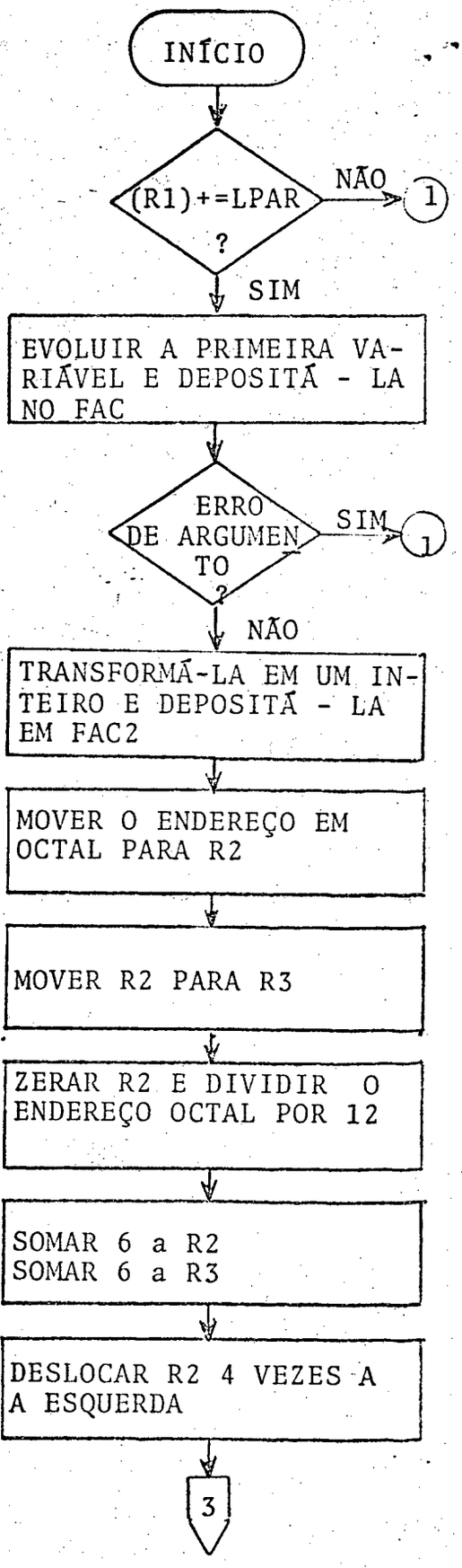
= M = valor a ser ajustado.

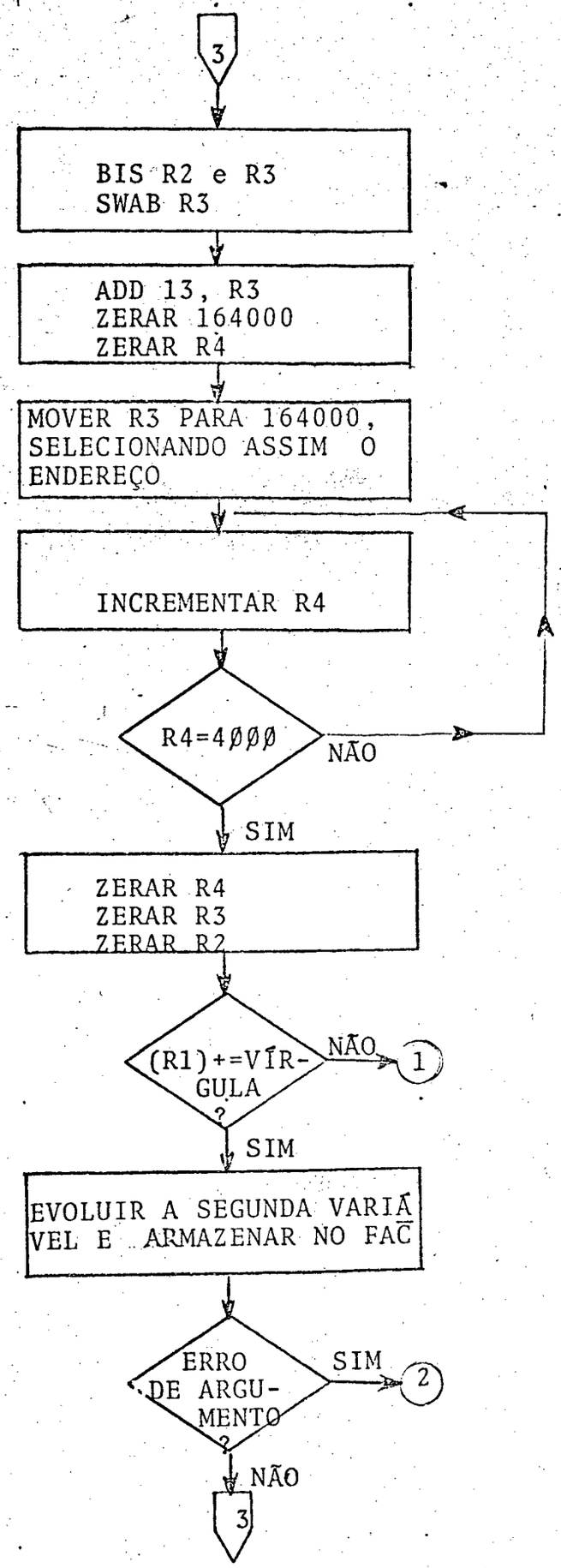
### 3.6.1 - Algorítmo da Subrotina POT

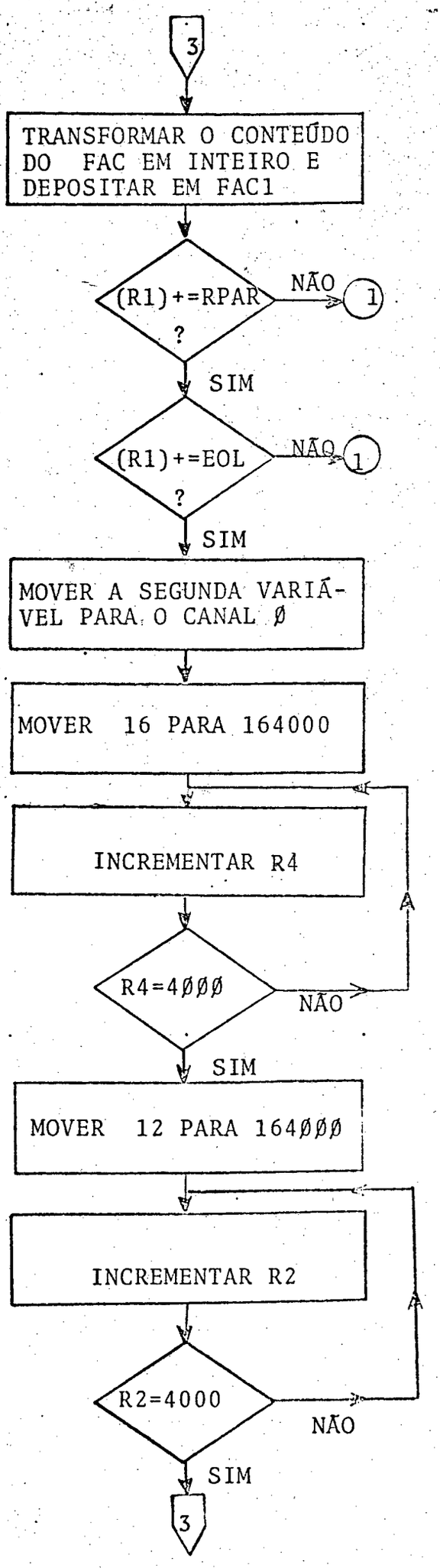
(1) Verificar parêntese a esquerda;
(2) Evoluir N e depositá-lo em FAC;
(3) Transformá-la no maior inteiro possível e depositá-la em FAC2;
(4) Depositar o conteúdo do FAC2 em R2, obtendo o endereço em octal;
(5) Zerar R4, R3;
(6) Transformar o endereço de octal para decimal, armazenando as unidades em R3 e as dezenas em R2;
(7) Criar o BCD excesso seis, para as dezenas;
(8) Criar o BCD excesso seis, para as unidades;
(9) Preparar o conteúdo de R2 para que este esteja ocupando a parte correspondente as dezenas;
(10) Mover os bits que estão a 1-lógico em R2, para R3, fazendo assim, com que R3 contenha o endereço do amplificador que queremos selecionar;
(11) Mudar o byte superior de R3 por um byte inferior, colocando assim, os bits em correspondência com os bits de 164ØØØ;
(12) Adicionar 13 ao conteúdo de R3, colocando os bits 0, 1 e 3 de R3 a 1-lógico;

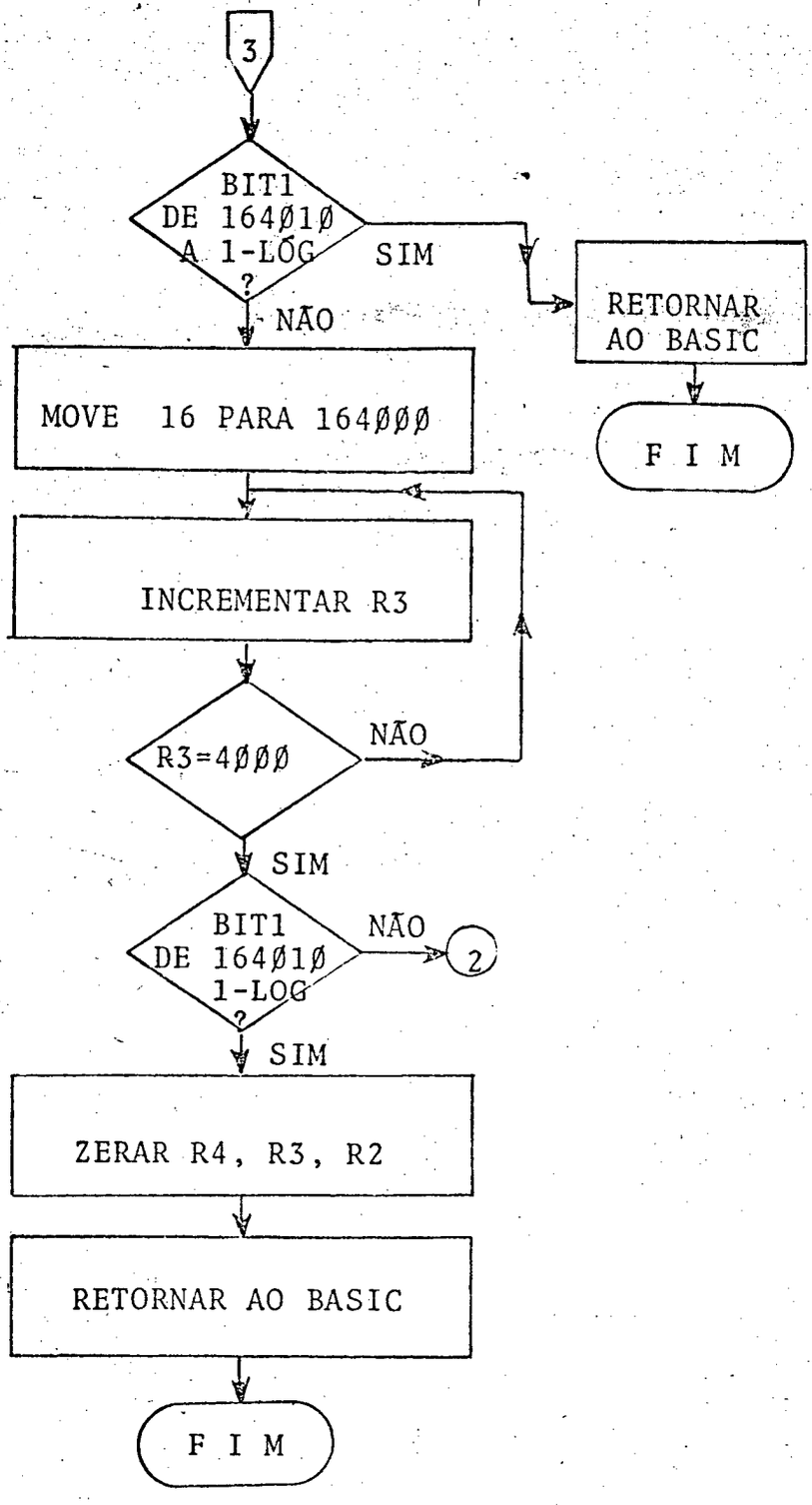
(13) Zerar o conteúdo de 164ØØØ e o conteúdo de R4;
(14) Mover R3 para 164ØØØ, selecionando assim o endereço do potenciômetro e acionando também a tecla pt na unidade de controle do analógico;
(15) Fazer um loop de espera, correspondente ao tempo de acomodamento dos reles no analógico;
(16) Zerar R4, R3 e R2;
(17) Verificar a vírgula de separação de variáveis;
(18) Evoluir a segunda variável e depositá-la no FAC;
(19) Tornar o conteúdo do FAC um inteiro e depositá-lo em FAC2;
(20) Verificar parêntese a direita;
(21) Verificar fim de linha;
(22) Mover a segunda variável para o canal Ø,que servirá como valor de referência para ajuste de servopotenciômetro;
(23) Ajustar servopotenciômetro;
(24) Fazer um loop de espera de ajuste;
(25) Testar o bit 1 de 164Ø1Ø, verificando se o ajuste do servopotenciômetro já terminou;
(26) Se não terminou, tentar ajustar novamente. Se não conseguir, desviar para a subrotina de erro de argumento;
(27) Se já terminou, retornar ao BASIC.

### 3.6.2 - Fluxograma da Subrotina POT.




3
BIS R2 e R3
SWAB R3
ADD 13, R3
ZERAR 164000
ZERAR R4
MOVER R3 PARA 164000, SELECIONANDO ASSIM O ENDEREÇO
INCREMENTAR R4
R4=4ØØØ
NÃO
SIM
ZERAR R4
ZERAR R3
ZERAR R2
(R1)+=VÍR-GULA ?
NÃO
1
SIM
EVOLUIR A SEGUNDA VARIÁVEL E ARMAZENAR NO FAC
ERRO DE ARGU-MENTO ?
SIM
2
NÃO
3


3
TRANSFORMAR O CONTEÚDO DO FAC EM INTEIRO E DEPOSITAR EM FAC1
(R1)+=RPAR ?
NÃO
1
SIM
(R1)+=EOL ?
NÃO
1
SIM
MOVER A SEGUNDA VARIÁVEL PARA O CANAL Ø
MOVER 16 PARA 164000
INCREMENTAR R4
A
R4=4ØØØ
NÃO
SIM
MOVER 12 PARA 164ØØØ
INCREMENTAR R2
A
R2=4000
NÃO
SIM
3


3
BIT1 DE 164Ø1Ø A 1-LÓG ?
SIM
NÃO
RETORNAR AO BASIC
F I M
MOVE 16 PARA 164ØØØ
INCREMENTAR R3
R3=4ØØØ
NÃO
SIM
BIT1 DE 164Ø1Ø 1-LOG ?
NÃO
2
SIM
ZERAR R4, R3, R2
RETORNAR AO BASIC
F I M

## 3.6.3 - Listagem da Subrotina POT

```
000674 122127'POT:    CMPB (%1)+,#.LPAR
       000000
000700 001371         BNE ESYN1
000702 004767'        JSR %7,EVAL
       000000
000706 103770         BCS EARG1
000710 004767'        JSR %7,INT
       000000
000714 016502'        MOV FAC2(R5),R2
       000000
000720 005004         CLR R4
000722 005003         CLR R3
000724 010203         MOV R2,R3
000726 005002         CLR R2
000730 071227 DIVI:   .WORD DIV,12
000732 000012
000734 062702         ADD #6,R2
       000006
000740 062703         ADD #6,R3
       000006
000744 006302         ASL R2
000746 006302         ASL R2
000750 006302         ASL R2
000752 006302         ASL R2
000754 050203         BIS R2,R3
000756 000303         SWAB R3
000760 062703         ADD #13,R3
       000013
000764 005037         CLR @#164000
       164000
000770 005004         CLR R4
000772 010337 MOVE:   MOV R3,@#164000
       164000
000776 032737         BIT #103,@#164010
       000103
       164010
001004 001772         BEQ MOVE
001006 005204 INCR:   INC R4
001010 020427         CMP R4,#4000
       004000
001014 001374         BNE INCR
001016 005004         CLR R4
001020 005003         CLR R3
001022 005002         CLR R2
001024 122127'        CMPB (R1)+,#.COMMA
       000000

                      PAGE    005

001030 001315         BNE ESYN1
001032 004767'        JSR R7,EVAL
       000000
001036 103714         BCS EARG1
001040 004767'        JSR R7,INT
       000000
```

DIGITAL EQUIPMENT COM. IND. LTDA.

```
001044 122127'          CMPB (R1)+,#.RPAR
       000000
001050 001305           BNE ESYN1
001052 122127'          CMPB (R1)+,#.EOL
       000000
001056 001302           BNE ESYN1
001060 016537'          MOV FAC2(R5),@#176760
       000000
       176760
001066 012737           MOV #16,@#164000
       000016
       164000
001074 005204 INCREM:   INC R4
001076 020427           CMP R4,#4000
       004000
001102 001374           BNE INCREM
001104 012737           MOV #12,@#164000
       000012
       164000
001112 005202 VOLTA:    INC R2
001114 020227           CMP R2,#4000
       004000
001120 001374           BNE VOLTA
001122 032737 AJUSTE:   BIT #2,@#164010
       000002
       164010
001130 001016           BNE FINE
001132 012737           MOV #16,@#164000
       000016
       164000
001140 005203 INCRI:    INC R3
001142 020327           CMP R3,#4000
       004000
001146 001374           BNE INCRI
001150 032737           BIT #2,@#164010
       000002
       164010
001156 001644           BEQ EARG1
001160 005004           CLR R4
001162 005003           CLR R3
001164 005002           CLR R2
001166 000207 FINE:     RTS %7

EOF ?
```

DIGITAL EQUIPMENT COM. IND. LTDA.

## 3.7 - Conclusão

A linguagem híbrida, constituida pelas subrotinas descritas neste Capítulo, tornou possível ao usuário que não conhece linguagem assembler, a utilização do computador analógico por meio do computador digital, dando-lhe, entre outras, as seguintes vantagens adicionais:

- Utilizar além das funções normais do compilador BASIC, os modos de cálculo e endereçamento do computador analógico;

- Obtenção dos sinais de saída dos amplificadores operacionais, dando uma maior versatilidade ao computador digital em termos de espaço de memória e tempo de programação, já que a integração numérica requer programações que tanto ocupam espaço de memória quanto tempo de programação;

- Eliminar as dificuldades encontradas em hardware, como as de esperar um tempo relativo de 18 m s para que haja acomodamento dos relés de endereçamento de amplificadores e potenciômetros.

No próximo Capítulo, aplica-se esta linguagem através da simulação de um motor dc no computador analógico e fazendo o controle por meio do computador digital.

# CAPÍTULO IV - APLICAÇÃO DA LINGUAGEM

# CAPÍTULO 4 - APLICAÇÃO DA LINGUAGEM

## 4.1 - Introdução

Neste Capítulo testaremos a linguagem híbrida, fazendo o controle de um sistema cujas equações de estado serão simuladas no computador analógico e as equações de controle, simuladas no computador digital. Os testes realizados no sistema, foram inicialmente feitos com o sistema em malha aberta e, posteriormente, foi aplicado o controle ao sistema. Os resultados deste teste são apresentados em forma gráfica, no final do Capítulo, assim como a listagem do programa de controle encontra-se no Anexo I.

## 4.2 - Formulação do Problema

A Fig. 4.1 representa um motor de corrente contínua, com corrente de armadura constante, cujo modelo matemático é representado pela Equação 4.1

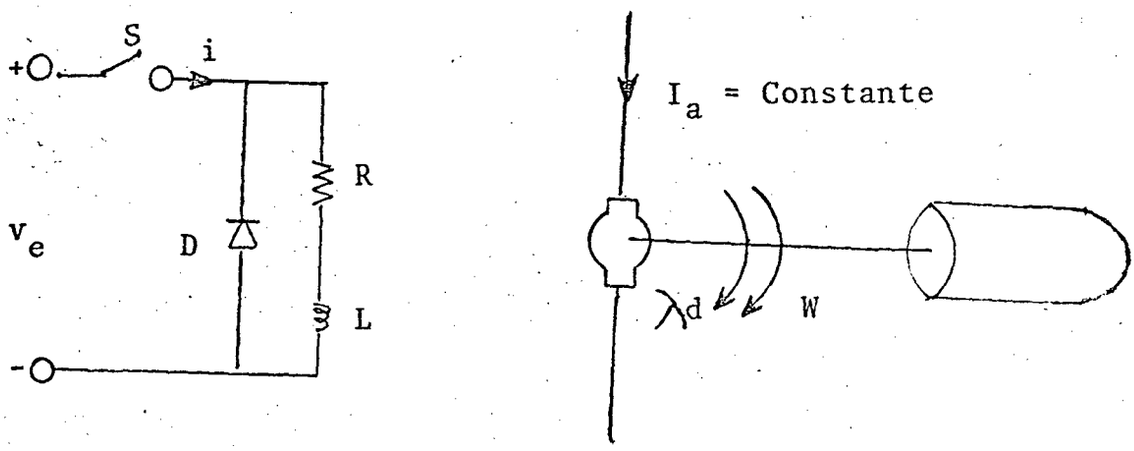



FIGURA 4.1 - Motor de Corrente Contínua

$$\begin{bmatrix} \frac{di\ (t)}{dt} \\ \frac{dw\ (t)}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & 0 \\ \frac{K}{I} & 0 \end{bmatrix} x \begin{bmatrix} i\ (t) \\ w\ (t) \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} v_e(t) \qquad (4.1)$$

onde: R resistência do circuito de campo.

L indutância do circuito de campo.

$\lambda d(t) = K\ i(t)$ torque instantâneo desenvolvido.

$K = K_t\ I_a$ é o produto do constante de torque pela corrente de armadura.

I é o momento angular de inércia.

A lei de controle ótimo que leva o sistema à origem minimizando a medida de desempenho do sistema,

$$J = \frac{1}{2} \int_o^{\infty} \left[ i^2(t) + w^2(t) + v_e^2(t) \right] dt , \qquad (4.2.)$$

é dado pela equação abaixo

$$U^* = \left[ K_1\ i(t) + K_2\ w(t) \right] , \qquad (4.3.)$$

Onde as constantes $K_1, K_2$ são definidas conforme Equações 4.4a à 4.4c.

$$K_1 = - \frac{a + \sqrt{a^2 + b^2/r\ (2c\ \sqrt{rq_2}/b + q_1)}}{b^2/r} , \qquad (4.4.a)$$

$$K_2 = \frac{\sqrt{rq_2}}{b} , \qquad (4.4.b)$$

$$a = \frac{R}{L}, \quad b = \frac{1}{L}, \quad c = \frac{K}{I}. \qquad (4.4c)$$

Deseja-se fazer o controle do sistema através de uma simulação híbrida onde o processo é simulado no computador analógico e o controlador no computador digital. As saídas do processo são convertidas para sinais digitais e injetadas no controlador que os processa e os devolve ao computador analógico, conforme Fig. 4.2.

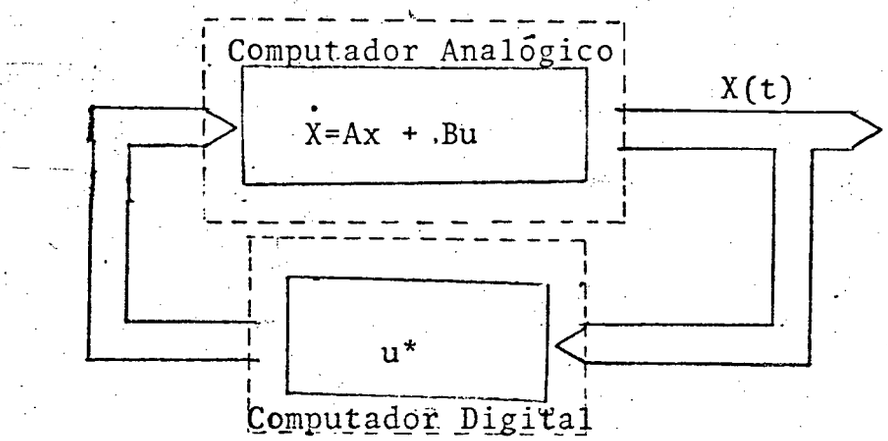



FIGURA 4.2 - Simulação Híbrida do Modelo

O esquema do modelo é apresentado na Fig. 4.3.

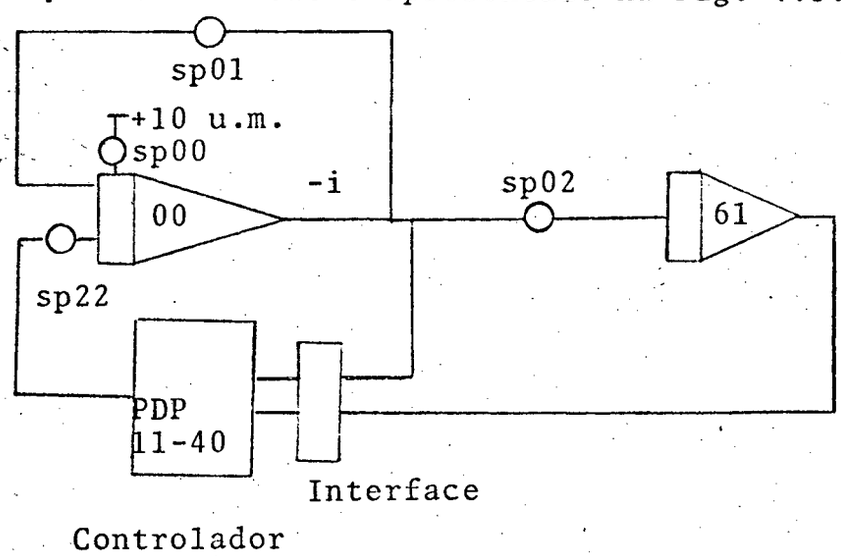



FIGURA 4.3 - Sistema Híbrido

Os valores atribuídos aos servopotenciômetros, foram:

a = SPO1 = 1
b = SP22 = 1
c = SPO2 = 1
SPOO = .2

## 4.3 - Resultados obtidos

Foi feita uma comparação entre o sistema em malha aberta e o sistema em malha fechada de tal maneira, à se fazer sentir a influência do controlador. No caso do sistema em malha aberta, em que o motor é alimentado tanto pela armadura quanto pelo indutor, o rotor que estava inicialmente bloqueado é liberado no instante $t = 0^+$ simultaneamente com a abertura da chave "S". Com isto, a energia armazenada na indutância "L" vai se descarregar através do diodo "D", o que equivale na Equação 4.1a anular $v_c$ e o motor vai girar. Tendo em vista a ausência de atrito e a alimentação da armadura por uma corrente constante o motor vai atingir uma velocidade de regime, conforme mostra a Fig. 4.4.

Na Fig. 4.5, é apresentada a resposta de velocidade do sistema, com este incialmente em malha aberta e depois de um certo intervalo de tempo é aplicado o comando U* de controle do sistema.

Na Fig. 4.6, é apresentada a resposta de corrente quando o sistema está em malha aberta e, na Fig. 4.7, é apresentada a curva da corrente com o sistema incialmente em malha aberta e, após um determinado instante de tempo, é aplicado o comando U* de controle do sistema.

## 4.4 - Conclusão

Como tínhamos proposto no início deste Capítulo, fez-se o controle de um sistema simulado no computador analógico por meio de um controle U*, simulado no computador digital.

Pelos resultados obtidos nas Fig. 4.5 e 4.7, podemos concluir que o controle U* aplicado ao sistema, levou-o para origem em um determinado intervalo de tempo, podendo-se afirmar que a linguagem híbrida série proposta neste trabalho, atingiu seus objetivos.

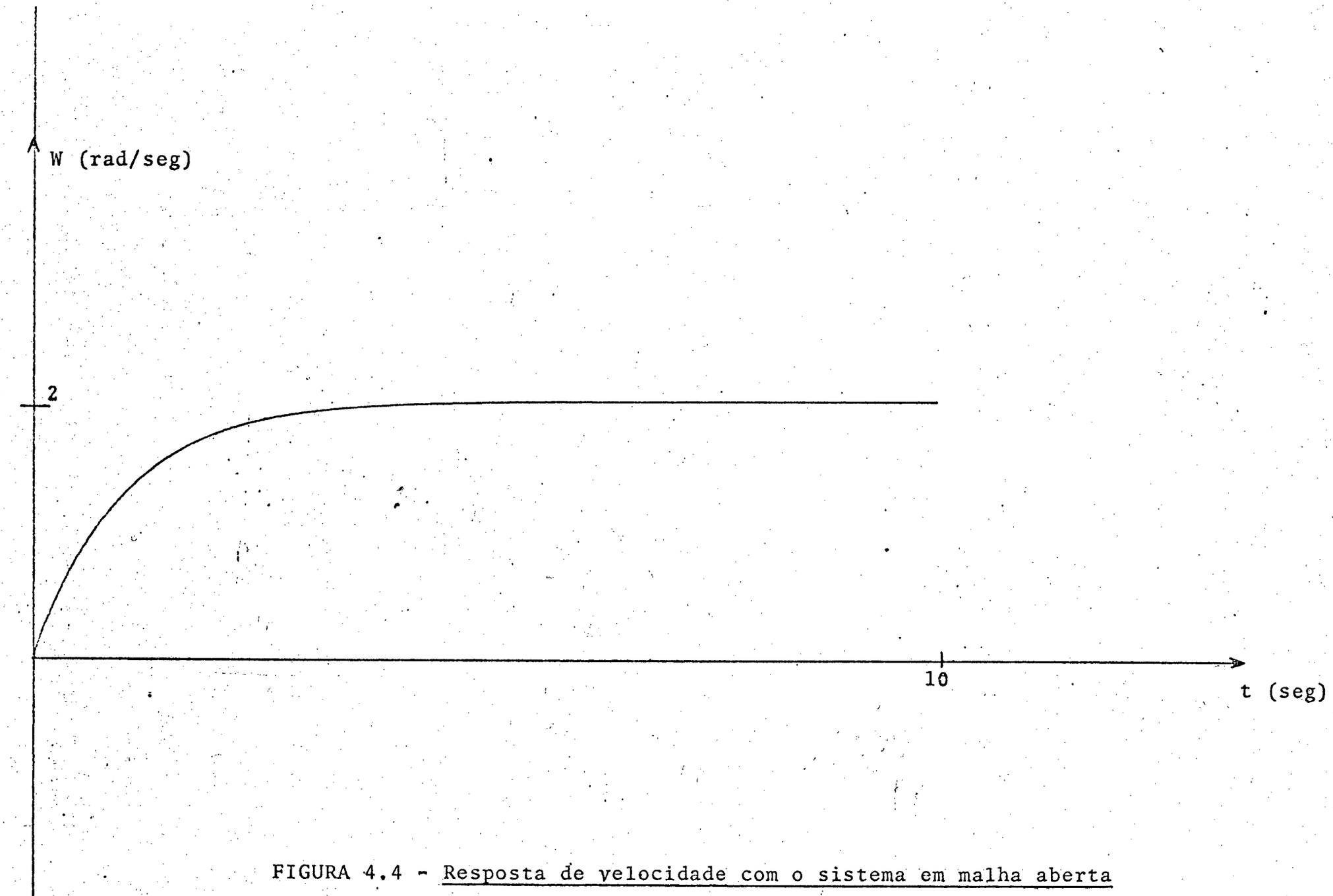



FIGURA 4.4 - Resposta de velocidade com o sistema em malha aberta

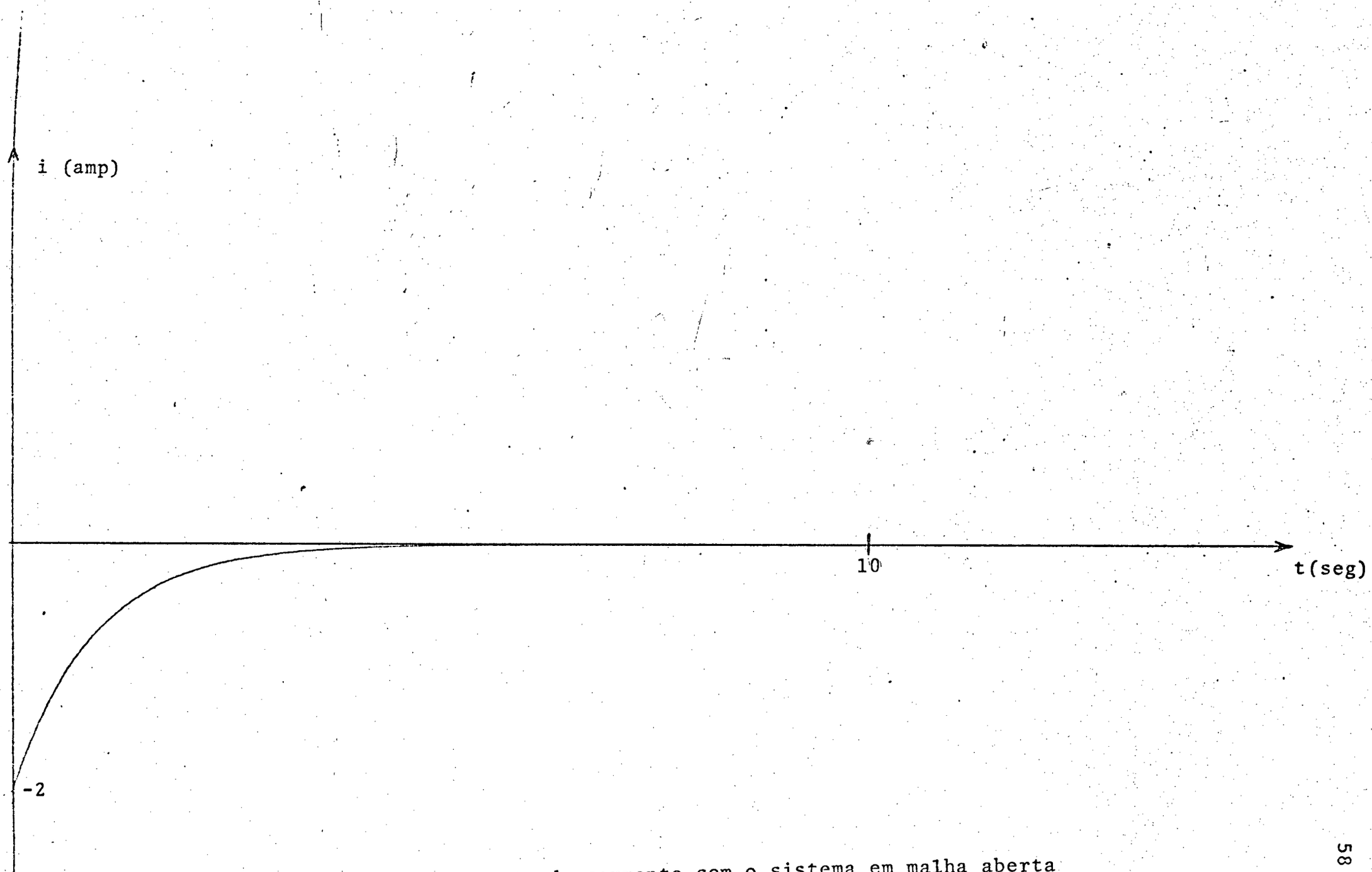



FIGURA 4.6. - Resposta de corrente com o sistema em malha aberta

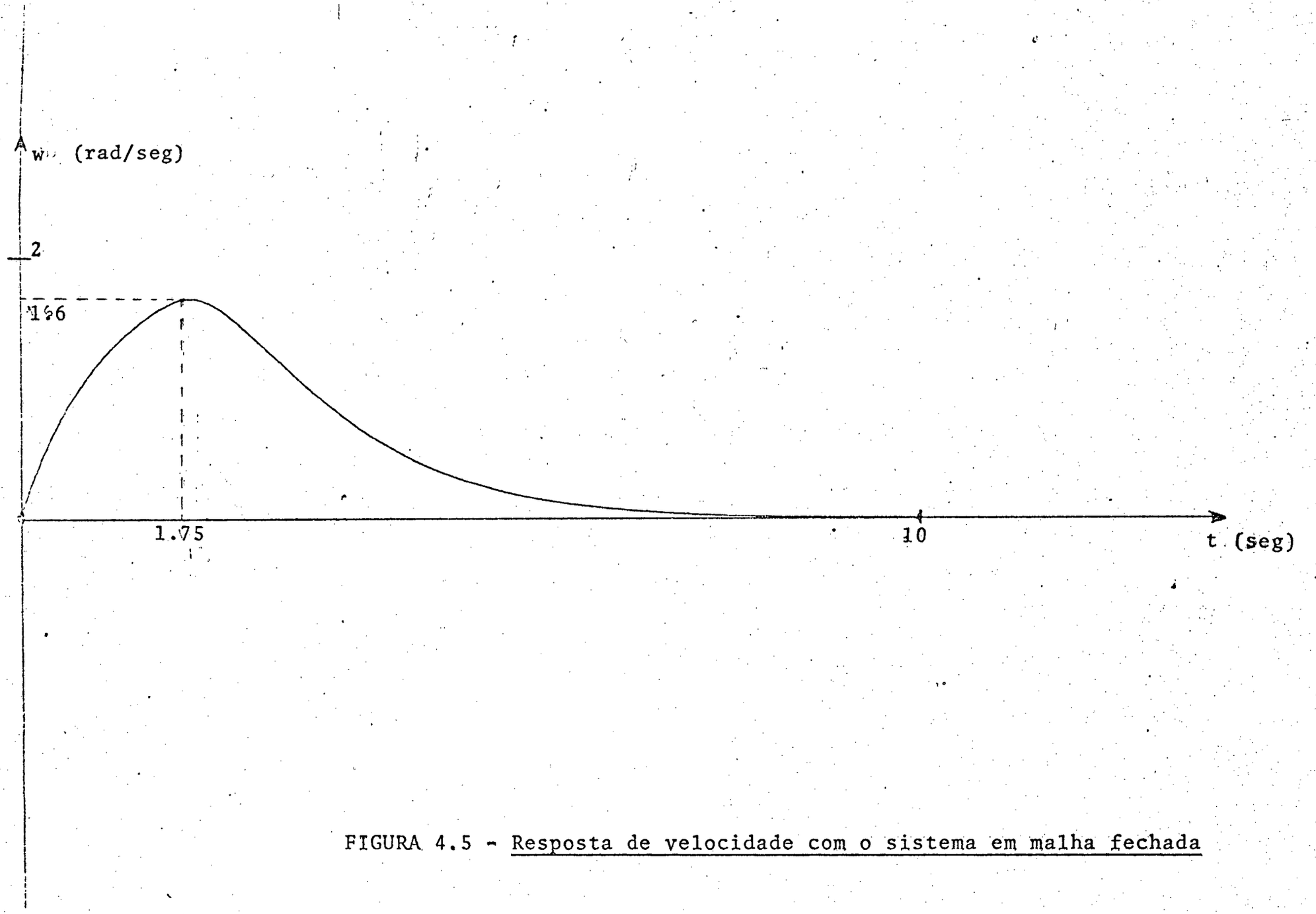



FIGURA 4.5 - Resposta de velocidade com o sistema em malha fechada

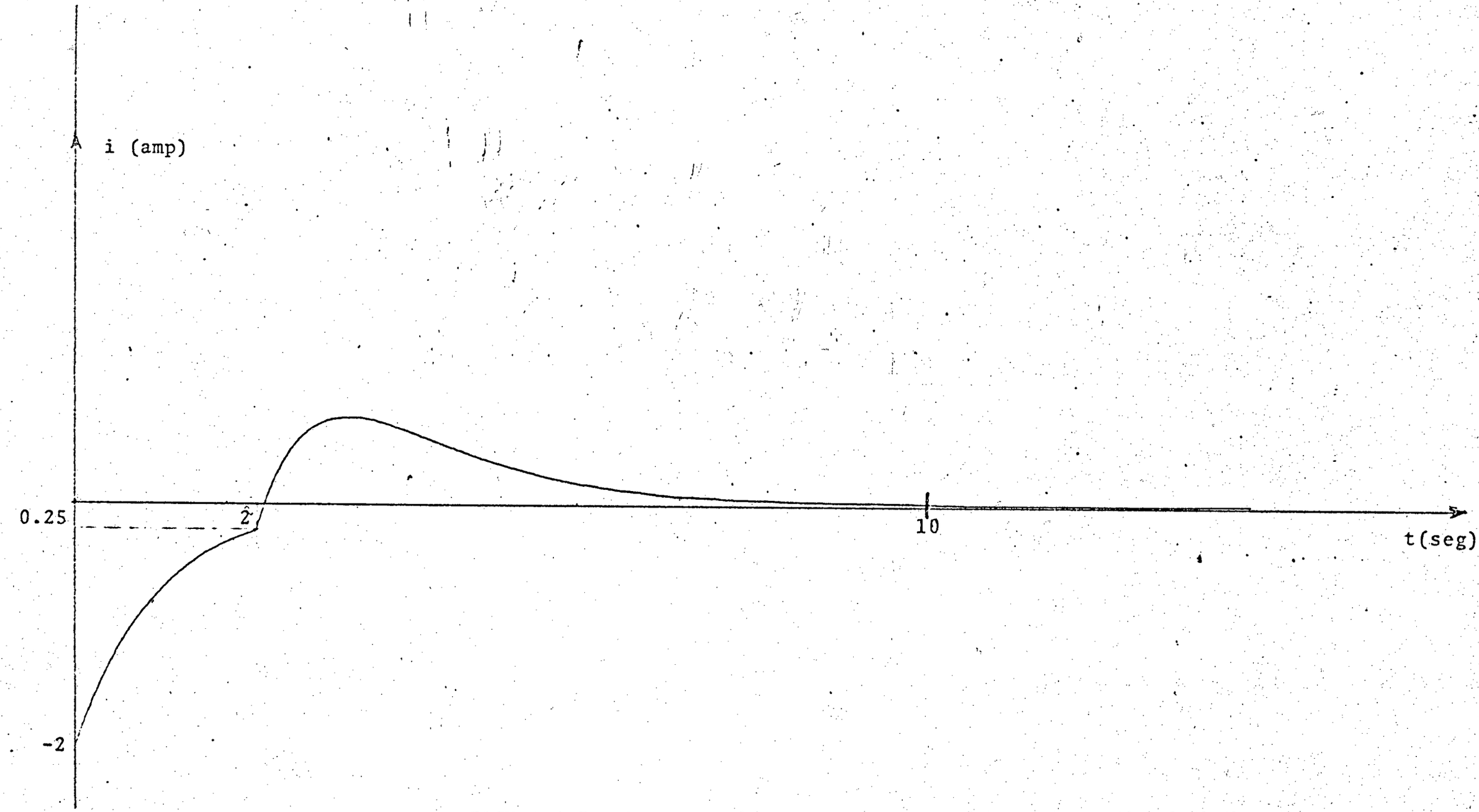



FIGURA 4.7 - Resposta de corrente com o sistema em malha fechada

# CAPÍTULO V - CONCLUSÃO

# CAPÍTULO 5 - CONCLUSÃO

A inclusão de subrotinas em linguagem de baixo nível em uma linguagem de alto nível, dá à linguagem híbrida resultante grande flexibilidade, já que o repertório inteiro de instruções da linguagem de alto nível é mantido e suplementado por um novo conjunto de instruções simples, projetadas para operar com o computador analógico. A linguagem híbrida formada, aumentou a flexibilidade da simulação analógica em termos de acesso a memória e aos controles do computador digital. Criou-se, portanto, uma linguagem híbrida série, já que o interface entre o computador analógico e o computador digital não utiliza o sistema de interrupção do PDP 11-40, permitindo assim, aos usuários, a utilização interpretador BASIC/PTS sem a preocupação do conhecimento de programação Assembler.

As Subrotinas de Amostragem e deamostragem encontram um grande campo de aplicação em controle numerico de processo, alguns trabalhos nêste sentido estão sendo desenvolvidos nesta Universidade, já com a utilização destas subrotinas. Um destes trabalhos é uma tese de mestrado, onde se faz um controle numérico de velocidade de um motor de onde o computador digital PDP 11-40 funciona como o controlador.

A construção de uma linguagem híbrida em tempo real, implica na solução do problema de tempo encontrando para chaveamento dos relés do computador analógico, necessitando-se assim, de que no futuro, se modifique as placas do KIT 11-H para que se possa utilizar o sistema de interrupção do PDP 11-40(1).

No entanto, a linguagem híbrida série foi construída sem qualquer prejuízo, a não ser o tempo de processamento, já que se pode parar o cálculo no computador analógico em qualquer instante, modificar parâmetros e dar continuidade ao mesmo sem nenhuma perda de informações.

Para dar continuidade a este trabalho, sugere-se o desenvolvimento de uma linguagem híbrida em tempo real devendo-se, no entanto, modificar as placas do KIT 11-H para utilizar-se o sistema de interrupção do PDP 11-40.

A aplicação da unidade híbrida formada, fica limitada pelo número de componentes em funcionamento do computador analógico, já que o minicomputador pode endereçar até 200 pontos no painel analógico mas, apenas, um terço destes pontos existem.

# ANEXO I - CUIDADOS DE OPERAÇÃO

# ANEXO I - CUIDADOS DE OPERAÇÃO

## Preparação dos computadores

Para se utilizar o interface, deve-se tomar os seguintes cuidados:

(1) Pressionar a tecla FREMD no painel de teclas da unidade de controle do computador analógico;

(2) Apertar a tecla EXTERN na unidade de endereçamento do computador analógico e pressionar levemente as teclas de endereços para que todas elas fiquem soltas;

(3) Ligar a chave do interface;

(4) Carregar a linguagem híbrida no computador digital.

Feito isto, qualquer uma das subrotinas pode ser chamada. Começaremos pela descrição da utilização da subrotina KAD. Faz-se o acesso a esta subrotina utilizando a declaração:

```
CALL "KAD"  (X1, X2, X3)
```

Sendo: X1 o canal que vai ser utilizado

X2 é a frequência que é um múltiplo da frequência básica que, no caso, é 10 KHertz. Pode-se calcular a frequência utilizando-se a fórmula abaixo:

$$f = \frac{10\ KHz}{X2}$$

X3 é o lugar onde fica armazenada a conversão.

Deve-se observar que em todas as vezes que quisermos tomar o valor de X3 convertido para decimal, deveremos dividir X3 por 409.6. Assim, por exemplo, se quisermos obter o valor de X3 em decimal, executaremos a declaração:

```
PRINT X3/409.6.
```

O acesso à subrotina DAC é feito utilizando-se a seguinte declaração:

```
CALL "DAC"  (X1, X2)
```

Onde a variável X1 pode ser qualquer um dos quatro canais de conversão digitais analógicos e a variável X2 deve ser multiplicada por um determinado fator, dependendo do canal escolhido para se fazer a conversão, conforme mostra a Tabela I.1 abaixo:

| Canal | Fator |
|---|---|
| Ø | 204.8 |
| 1 | 204.8 |
| 2 | 409.6 |
| 3 | 409.6 |

TABELA I.1

Por exemplo, se quizermos fazer conversão no canal Ø, então chamaremos:

CALL "DAC" (Ø, 204.8* X)

O acesso a subrotina ENDE é feito utilizando-se a declaração:

CALL "ENDE" (N)

Onde N representa o endereço de um amplificador.

É necessário que se faça, depois desta subrotina,um loop de espera para que haja acomodamento dos relés de endereçamento.Como exemplo, pode ser vista a listagem feita no final deste ANEXO.

O acesso a subrotina CALC é feito utilizando-se a declaração:

CALL "CALC" (X)

Onde X é o número correspondente a um dos comandos de cálculo, conforme Tabela I.2.

| COMANDO (s) | NÚMERO |
|---|---|
| DAVER, PAUSE | 0 |
| REPET, PAUSE | 128 |
| MIT HALT, PAUSE | 64: |
| IT HAND, PAUSE | 192 |
| DAUER, RECHNEN | 32 |
| REPET, RECHNEN | 16Ø |
| MIT HALT, RECHNEN | 96 |
| IT HAND, RECHNEN | 324 |
| DAVER, HALT | 16 |
| REPET HALT | 144 |
| MIT HALT, HALT | 80 |
| IT HAND, HALT | 200 |

TABELA I.2

Deve-se ressaltar que só se consegue o sinal WEITER quando feita uma transição do bit 4 da palavra 164ØØØ, estando o bit 5 a 1-lógico. Portanto, para que o sinal WETTER seja atingido, é necessário que nos estejamos anteriormente no modo RECHNEN. Este sinal pode ser atingido acessando a subrotina CALC, sendo X o número correspondente a um dos comandos, conforme Tabela I.3.

| COMANDO | NÚMERO |
|---|---|
| DAUER, RECHNEN, WEITER | 48 |
| REPET, RECHNEN, WEITER | 176 |
| MIT HALT, RECHNEN, WEITER | 112 |
| IT HAND, RECHNEN, WEITER | 240 |

TABELA I.3

O acesso à subrotina POT é feito utilizando-se a seguinte declaração:

CALL "POT" (X1, - 204.8X2)

Onde X1 é o endereço do servopotenciômetro que nós desejamos ajustar e, X2, é o valor de ajuste. Por exemplo, se quizermos ajustar o servopotenciômetro 02 com o valor .2, basta chamar:

CALL "POT" (02, - 204.8 x 2).

## Anexo 1 - Listagem do Programa de Teste

```
LIST


0 DIM W1(0),W2(0),X1(1000),X2(1000),U(1000)
10 INPUT N,A,B,C,R,Q1,Q2
15 INPUT U1\CALL "DAC"(1,0)
20 CALL "ENDE"(N)
25 FOR J=1 TO 4000
30 NEXT J
35 CALL "CALC"(32)
40 FOR S=1 TO 7000
45 NEXT S
50 I=0
55 CALL "KAD"(4,1,W1)
60 CALL "KAD"(5,1,W2)
65 I=I+1
70 X1=2*(W1(0)/409.6-5)
75 X2=2*(W2(0)/409.6-5)
80 K2=SQR(R*Q2)/B
85 Y=R/B^2
90 Z=A^2+(2*C*SQR(R*Q2)/B+Q1)/Y
95 K1=Y*(-A+SQR(Z))
100 U=-B/R*(K1*X1+K2*X2)
105 IF ABS(U)<9 THEN 110 \U=U1
110 X1(I)=X1\X2(I)=X2\U(I)=U
115 CALL "DAC"(1,204.8*U)
120 GO TO 55

READY
```

DIGITAL EQUIPMENT COM. IND. LTDA.

# ANEXO II - LISTAGEM DA FTBL E MAPA DE ENDEREÇOS DAS SUBROTINAS

## Anexo 2 - Listagem da FTBL

```
                                ;GERACAO DA TABELA DE FUNCOES
                                .TITLE FTBL
                                .GLOBL KAD,DAC,CALC,ENDE,POT
            000000              .ASECT
            000046              .=46
000046      000000'             .WORD FTBL
            000000              .CSECT
000000          113 FTBL:       .ASCII /KAD/
000001          101
000002          104
000003          000             .BYTE 0
000004      000000'             .WORD KAD
000006          104             .ASCII /DAC/
000007          101
000010          103
000011          000             .BYTE 0
000012      000000'             .WORD DAC
000014          103             .ASCII /CALC/
000015          101
000016          114
000017          103
000020      000000'             .WORD CALC
000022          105             .ASCII /ENDE/
000023          116
000024          104
000025          105
000026      000000'             .WORD ENDE
000030          120             .ASCII /POT/
000031          117
000032          124
000033          000             .BYTE 0
000034      000000'             .WORD POT
000036          000             .BYTE 0
            000040              .EVEN
            000001              .END
```

DIGITAL EQUIPMENT COM. IND. LTDA.

## Anexo 2 - Mapa de Endereços das Subrotinas

```
LINK-11S V002A
*I H

*O H

*M T

*T

*B 0

PASS 1

*

*

*

*

*

* E

LOAD MAP

TRANSFER ADDRESS: 031400
LOW LIMIT: 000000
HIGH LIMIT: 032720
**********
MODULE  BASICL
SECTION ENTRY    ADDRESS SIZE
<. ABS.>         000000  000000
<      >         000000  021426
          ARGB   011650
          ARRAYS 000010
          COLUMN 000034
          ERRARG 007734
          ERRMIX 011650
          ERRPDL 011120
          ERRSYN 011650
          EVAL   010616
          FAC1   000040
          FAC2   000042
          FILLCO 000110
          GETVAR 013604
          INT    014004
          LIMIT  000002
          MAKEST 011650
          MSG    015474
          NORM   015554
          NUMOUT 015746
```

DIGITAL EQUIPMENT COM. IND. LTDA.

```
            NUMSGN   015760
            OPRATO   011272
            PDL      000004
            PDSIZE   000006
            RND1     000064
            RND2     000066
            SAVCHA   011650.
            SOPRAT   011650
            START    001034
            STOVAR   017244
            STPRO    011650
            TABLE5   010756
            TBL5EN   011002
            TKS      177560
            TPB      177566
            T1       000056
            T2       000060
            T3       000062
            VAL      020702
            $STKSZ   000200
            .COMMA   000243
            .EOL     000201
            .LPAR    000255
            .RPAR    000237
      **********
      MODULE   FTBL
      SECTION  ENTRY    ADDRESS  SIZE
      <      >          021426   000037
      **********
      MODULE   USRFNS
      SECTION  ENTRY    ADDRESS  SIZE
      <      >          021466   001170
            CALC     022122
            DAC      021736
            ENDE     022220
            KAD      021466
            POT      022362
      **********
      MODULE   FPMP11
      SECTION  ENTRY    ADDRESS  SIZE
      <      >          022656   004276
            AINT     023774
            ALOG     023420
            ATAN     026270
            COS      025714
            ERRFPU   022660
            EXP      024544
            IFPMP    022656
            SIN      025750
            SQRT     026760
            $ADR     022664
            $DVR     024112
            $ERR     027116
            $ERRA    027126
            $ERVEC   027146
            $INTR    024012
            $IR      025114
            $MLR     025200
            $POLSH   026754
            $POPR3   025564
```

DIGITAL EQUIPMENT COM. IND. LTDA.

```
            $POPR4   025552
            $POPR5   025552
            $RI      025572
            $SBR     022660
            $V20A    026754
**********
MODULE   BASICH
SECTION  ENTRY    ADDRESS  SIZE
<       >         027154   003544
            USRARE   027154

PASS 2
```

# REFERÊNCIAS

(1) MARTINS, Daniel A., Interface Entre um Minicomputador PDP 11-40 e um Computador Analógico RA 770

(2) PDP 11-40 - "Laboratory Peripheral System LPS 11 - User's Guide, Digital Equipmente Corporation, 1973.

(3) KRAUSE, Paul C., LIPO Thomas A., CARROL Dennis P., - "Applications of Analog and Hybrid Computation in Eletric Power System Analysis", Proc. I E E E,7(62): 994, July, 1974

(4) RUBIN, Arthur I., KEENE, David H. - "The Future of Hybrid Computation", Proc. Simulation, 55-59, August 1975.

(5) EATOCK, R; AL-DABASS, D. - "BASIC External Function for the Control of an Analogue Computer System ", Simulation 103-108, April, 1975.

(6) HØEJBER, K. Søe - "ANIBAL - A Hybrid Computer Language for EAI 680 - PDP 8/I, FPP 12, Simulation, 159-163, November, 1974.

(7) KIRK, E. Donald,- "Optimal Control Theory", Prentice-Hall, INC, 1970, Cap. 5.

(8) PDP 11-40 - LINK 11S, Linker Programming Mannual",DEC, 1973.

# B I B L I O G R A F I A

PDP 11-40 - Processos Handbook, DEC, 1972.

PDP 11 - Peripherals Handbook. Programming and Interfacing, DEC, 1972.

PDP 11 - BASIC - User's Guide, DEC, 1972

PDP 11 - Laboratory Peripherals System LPS 11 - Maitenance Manual, DEC, 1973

PDP 11 - PAL 11S Assembler Programming Manual , DEC, 1973