

Boletim Técnico da Escola Politécnica da USP

Departamento de Engenharia Eletrônica

BT/PEE/94-07

**Identificação de Pulsos
Decádicos em Linhas
Telefônicas**

**Antonio P. Timoszczuk
Márcio A. Mathias
Euvaldo F. Cabral Jr.**

São Paulo - 1994

Timoszczuk, Antonio P

Identificação de pulsos decádicos em linhas telefônicas / A.P. Timoszczuk, M.A. Mathias, E.F. Cabral Júnior. -- São Paulo : EPUSP, 1994.

22p. -- (Boletim Técnico da Escola Politécnica da USP, Departamento de Engenharia Eletrônica, BT/PEE/94-07)

1. Telefonia 2. Sinais - Processamento digital
I. Mathias, Márcio A II. Cabral Júnior, Euvaldo F
III. Universidade de São Paulo. Escola Politécnica.
Departamento de Engenharia Eletrônica IV. Título V.
Série

CDD 621.385
621.3822

Índice

Resumo	2
Abstract	2
1. Introdução	3
2. Objetivos	3
3. Descrição seqüencial	3
3.1 Descrição do sistema para aquisição de dados	4
3.2 Coleta das amostras	5
3.3 Pré-processamento	5
3.4 Tratamento dos pulsos	5
3.4.1 Evento fechamento de contato	7
3.4.2 Evento abertura de contato	8
3.5 Análise das amostras pelo método de "clusters"	8
3.5.1 Cálculo das distâncias Euclidianas	9
3.5.2 Determinação dos clusters e cálculo do centróide de cada "cluster"	9
3.4.3 Identificação dos pulsos	10
4. Resultados	10
5. Conclusão	14
6. Bibliografia	14

IDENTIFICAÇÃO DE PULSOS DECÁDICOS EM LINHAS TELEFÔNICAS

Antonio P. Timoszczuk, Márcio A. Mathias e Euvaldo F. Cabral Jr.

Resumo

Neste trabalho é apresentado um método de detecção de pulsos telefônicos decádicos verdadeiros de outros pulsos. É feita uma descrição sucinta do problema e apresentado o meio utilizado para obtenção das amostras. O método escolhido para identificação dos números discados, é o da análise espectral, associado a clusterização das amostras semelhantes. Por último são mostrados os resultados obtidos, que demonstram a viabilidade do método empregado e são apresentadas algumas conclusões.

IDENTIFICATION OF DECADIC PULSES ON TELEPHONE LINES

Antonio P. Timoszczuk, Márcio A. Mathias e Euvaldo F. Cabral Jr.

Abstract

This work presents a method for automatic detection of true decadic telephone pulses. After a brief description of the problem, the way the pulses were obtained is described followed by an explanation of the chosen method to perform the identification. Practical results showed that the method performed a successful identification task.

1. Introdução

O segmento de serviços oferecidos aos clientes por parte de empresas tais como bancos, administradoras de cartão de crédito e afins experimentou nos últimos anos um vertiginoso crescimento do grau de informatização, com o objetivo de tornar a prestação do serviço ao cliente mais ágil e eficiente.

A utilização da rede telefônica como meio de comunicação entre o usuário e os serviços (automatizados) depende de dispositivos inteligentes que possuam interfaces homem-máquina capazes de suportar um processo iterativo como, por exemplo, a solicitação por parte da máquina do número da conta corrente do usuário e, após o reconhecimento do número, a senha de acesso do serviço desejado para fins de verificação se o indivíduo está habilitado ou não para utilizar o serviço.

O meio mais simples que o usuário dispõe para acessar estes serviços é o próprio aparelho telefônico que, em sua versão mais comum, é do tipo decádico. Caso o aparelho telefônico seja do tipo multi-frequencial, nenhum problema de entendimento dos dígitos por parte da máquina ocorrerá, pois os números teclados são convertidos em tons (pares de frequências) que por meio de filtros (digitais ou analógicos) são facilmente interpretáveis e a informação é recebida com grande confiabilidade.

A dificuldade ocorre quando o aparelho telefônico é do tipo decádico, que transmite os números discados/teclados por meio de aberturas e fechamentos de enlace ("loops" de corrente). Estes sinais são de difícil detecção pois podem ser facilmente confundidos com ruídos presentes no ambiente ou mesmo no meio de comunicação (rede telefônica, central de comutação), levando o dispositivo detector a uma interpretação errônea da informação fornecida pelo usuário.

Atualmente na rede telefônica brasileira, cerca de 98% dos aparelhos telefônicos são do tipo decádico, fato que motiva a busca de um método eficiente e confiável para o reconhecimento dos dígitos fornecidos pelo usuário através deste tipo de aparelho.

Neste trabalho é verificada a viabilidade de se utilizar um método de análise de frequências, associado a técnicas de "clustering" para a identificação de pulsos decádicos verdadeiros.

2. Objetivos

O objetivo é a partir de amostras de pulsos coletadas, efetuar um processamento para análise em frequência destas amostras. O resultado desta análise em frequência é então mapeado para o plano, sendo buscadas similaridades entre as amostras válidas que permitam identificar os pulsos decádicos verdadeiros, diferenciando-os dos ruídos e dos pulsos causados pelo curto circuito e realimentação da cápsula microfônica.

3. Descrição sequencial

Para a execução desta análise foram efetuados os seguintes passos: aquisição das amostras, pré-processamento para adequação das amostras ao programa MATLAB utilizado para a análise das amostras, processamento das amostras para obtenção das sequências correspondentes em frequência, mapeamento destas sequências no plano para análise.

3.1 Descrição do sistema para aquisição de dados

Para a aquisição de dados foi implementado um "hardware" em montagem "wire-wrap" juntamente com um simulador de linha telefônica, uma unidade CPU padrão baseada no microprocessador MC6809 (MOTOROLA) e um computador pessoal do tipo PC-486.

A Figura 3.1, apresenta o hardware de aquisição através de um diagrama em blocos :

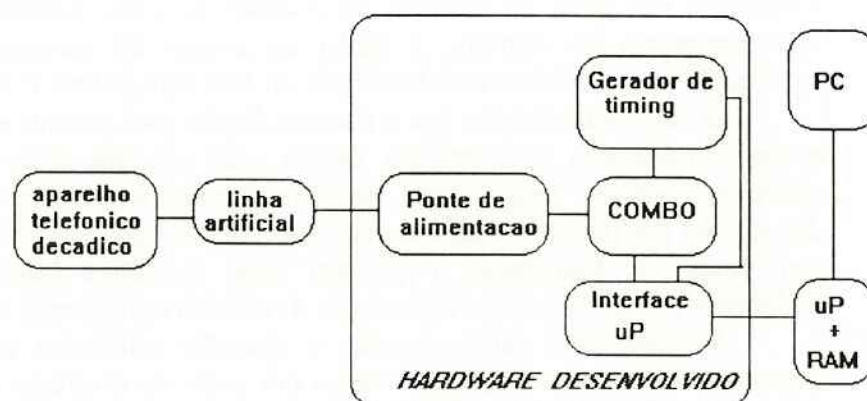


Figura 3.1 - Sistema de aquisição.

Bloco 1 - Ponte de Alimentação: este bloco é formado por uma ponte de alimentação de -48 Volts, responsável pela alimentação do aparelho telefônico e acoplamento dos sinais discados com o COMBO (ver abaixo);

Bloco 2 - Gerador de Timing: baseado em um oscilador a cristal e um conjunto de contadores lógicos (divisores), é responsável pelo sincronismo e fornecimento da frequência de operação do circuito;

Bloco 3 - COMBO (Codificador/decodificador + filtro): é responsável pela amostragem dos pulsos provenientes do telefone, codificando as amostras na forma PCM segundo a lei A de formação. A taxa de amostragem do sinal é de 8 KHz, gerando portanto uma palavra PCM na forma serial (8 bits a cada 125 us). Possui também um filtro anti-aliasing, segundo as características definidas no CCITT (the International Telegraph and Telephone Consultative Committee).

Bloco 4 - Interface com o microprocessador: efetua a conversão serial/paralela das amostras.

Os pulsos (dígitos) do telefone são transmitidos ao "hardware" de aquisição através de uma linha artificial (simulando 3 Km de linha telefônica 0.4mm²), o sinal analógico correspondente é então convertido para a forma digital segundo a codificação PCM-8 bits, obedecendo a lei A de formação.

A CPU por meio de um processo de varredura temporal lê as amostras (cada amostra = 1byte) que são armazenadas em uma memória (RAM), num total de 10 kbytes por evento. As amostras então sofrem o primeiro processamento pelo CPU e cada byte armazenado tem seus bits pares invertidos (isto ocorre para compensar uma pré inversão destes bits, realizada pelo circuito COMBO). Os dados são então enviados para um computador tipo IBM-PC via interface serial RS-232 e armazenados em arquivos no disco rígido da máquina (um arquivo para cada evento).

3.2 Coleta das amostras

A coleta das amostras foi feita utilizando-se dois aparelhos telefônicos decádicos, um utilizando disco e outro do tipo teclado. De um total de 15 ensaios foram coletados 30 eventos (abertura/fechamento de contato). Variando-se o comprimento da linha (0,5 a 3,5 Km) no simulador (linha artificial) não se notou mudança na amplitude nem no formato dos pulsos.

3.3 Pré-processamento

Cada arquivo de dados foi processado, passando por uma descompressão, de acordo com a lei A. Esta descompressão consiste em uma linearização, uma vez que os dados durante a codificação A/D sofreram um processo de compressão. Este processo é realizado por uma rotina escrita em linguagem C, e ao final da descompressão cada amostra que era representada por um byte passa a ser representada por 12 bits. As amostras foram então armazenadas em arquivos no formato ASCII adequados para serem lidos através do programa "MATLAB".

3.4 Tratamento dos pulsos

Uma emissão de pulsos decádicos é feita através de um dispositivo (disco ou teclado) do telefone que provoca interrupções na corrente elétrica DC que circula na linha telefônica.

O sinal emitido pelo aparelho telefônico, ao ser pressionada uma tecla (ou acionado o disco), é um trem de pulsos que chaveará a linha um número de vezes igual ao dígito da tecla pressionada, sendo que o dígito zero corresponde a dez pulsos. Os pulsos têm frequência nominal de 10Hz, relação de abertura/fechamento nominal de 2:1, e a tolerância para ambos os parâmetros, qualquer que seja a velocidade com que as teclas sejam pressionadas é a seguinte:

- a) tempo de abertura: entre 58ms e 77ms;
- b) tempo de fechamento: entre 28ms e 40ms.

A pausa interdigital é de no mínimo 700ms, e no máximo de 1.300ms, qualquer que seja a velocidade com que as teclas sejam pressionadas. A Figura 3.2, apresenta o evento dos pulsos decádicos ocorrendo na linha (de forma ideal). O período 1 representa o monofone em repouso (no gancho), e a corrente elétrica circulante na linha é nula. Quando o monofone é removido do gancho começa a circular na linha a corrente elétrica i_1 . Ao se girar o disco para a direita, a cápsula do microfone é curto-circuitada, o que diminui a impedância na linha e provoca portanto uma elevação na corrente circulante na linha, i_2 . Quando o disco é liberado ele tende a retornar a sua posição de repouso e, durante seu percurso de volta, contatos interruptores são abertos tantas vezes quanto o número do dígito que se deseja transmitir.

No aparelho decádico de teclado (também chamado decádico eletrônico ou decádico compatível), o chaveamento da linha ocorre de maneira similar, porém no lugar de contatos mecânicos, são utilizadas chaves semicondutoras.

Estes pulsos possuem componentes espectrais fora da banda do canal telefônico e estas são também fortemente rejeitadas pelo filtro passa banda na entrada do conversor A/D da interface de linha do assinante na central telefônica. Devido a isto uma distorção destes pulsos pode ser verificada quando convertemos as amostras digitalizadas para o

domínio analógico através de um conversor D/A. O sinal da Figura 3.3 apresenta os pulsos, após uma conversão A/D $\rightarrow$ D/A.

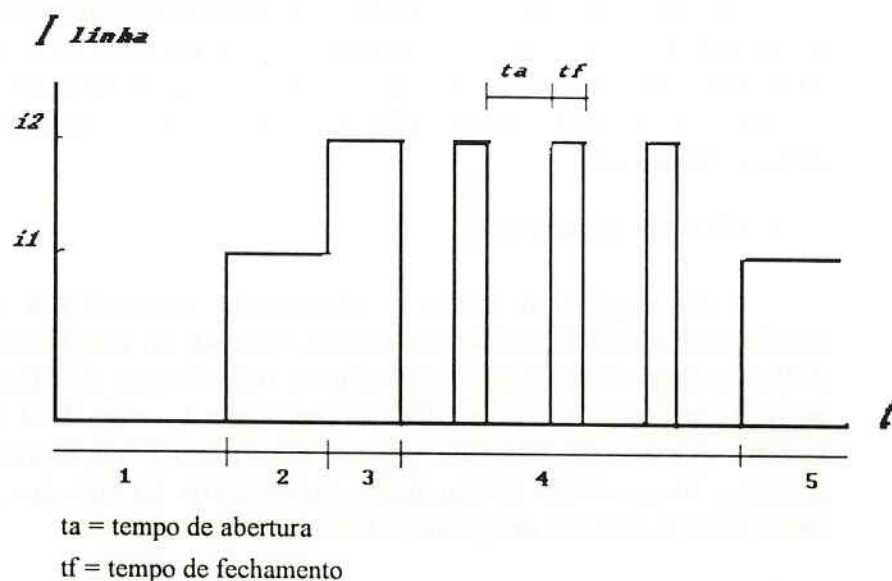


Figura 3.2 - Pulsos decádicos.

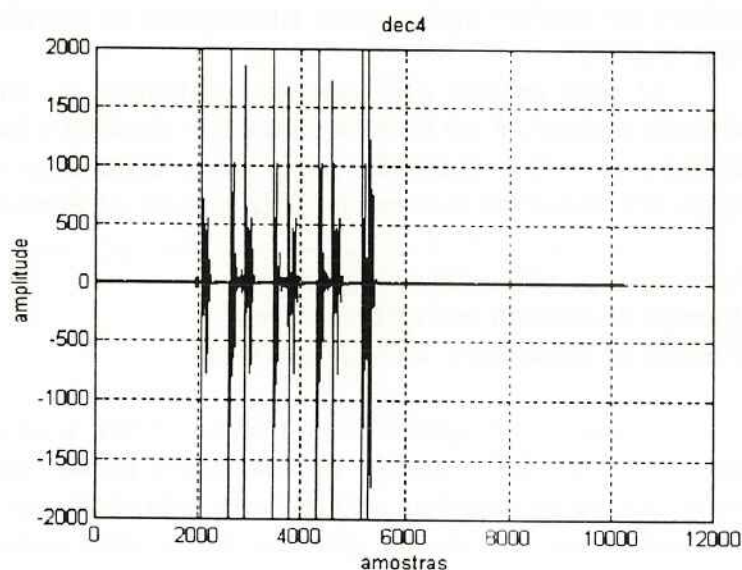


Figura 3.3 - Pulsos correspondentes ao dígito 4.

Através do estudo da composição espectral dos pulsos de abertura e fechamento de contato, é feito um estudo sobre a semelhança entre as amostras obtidas, de forma a caracterizá-las diante de um pulso desconhecido, permitindo identificar pulsos válidos e ruídos.

Os tempos entre dois eventos (abertura e fechamento de contato) bem como os tempos entre os dígitos discados são padronizados, conforme descrito anteriormente, e levando-se em consideração suas respectivas tolerâncias, estes podem ser tratados por um software que faça a medida destes tempos, de maneira trivial, desde que consigamos reconhecer se um dado pulso é válido ou não.

A partir deste momento, vamos concentrar nossa atenção na análise espectral dos eventos, como forma de caracterizar o evento pulso válido.

3.4.1 Evento fechamento de contato

A partir de uma amostra coletada analisou-se isoladamente o evento "fechamento de contato". Para isso foi usada a "janela retangular", visto ser esta apropriada para este tipo de análise pois possui um ponderamento uniforme sobre a amostra no domínio do tempo, preservando assim as amplitudes relativas do sinal analisado.

A Figura 3.4, apresenta uma amostra no domínio do tempo, delimitada pela janela retangular, tendo sido considerado uma janela com o comprimento de 256 unidades.

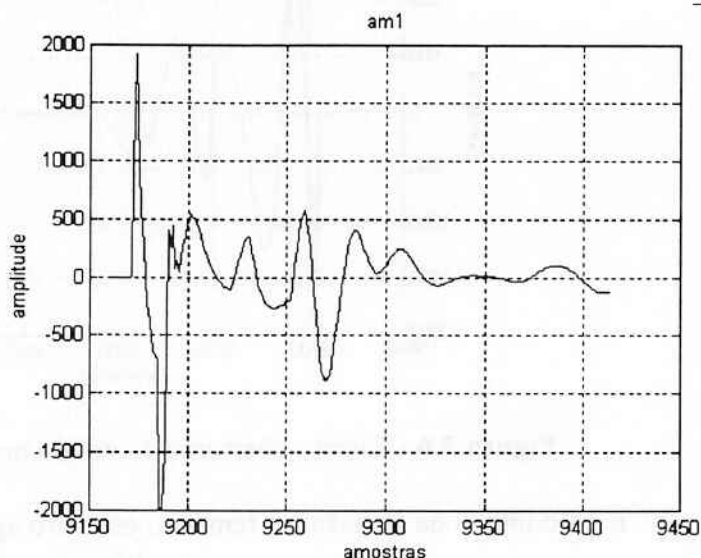


Figura 3.4 - Evento fechamento de contato no domínio do tempo.

Sobre esta sequência no tempo aplicou-se uma DFT ("Discret Fourier Transform") através do algoritmo FFT ("Fast Fourier Transform") disponível no pacote de "software" "MATLAB".

Foi obtida então sua composição espectral, conforme apresentado na Figura 3.5.

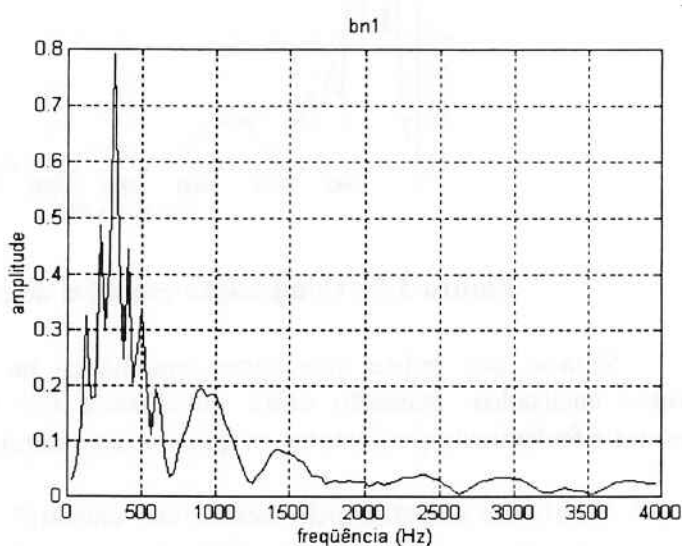


Figura 3.5 - Composição espectral do pulso de fechamento.

3.4.2 Evento abertura de contato

Repetindo o procedimento anterior para analisar agora o evento "abertura de contato" tem-se, no domínio do tempo, o gráfico apresentado na Figura 3.6.

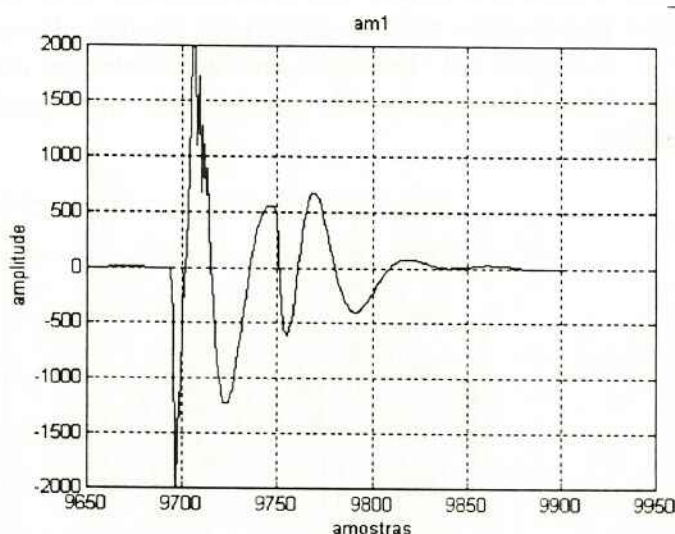


Figura 3.6 - Evento abertura de contato no domínio do tempo.

E no domínio da frequência, tem-se o espectro apresentado na Figura 3.7.

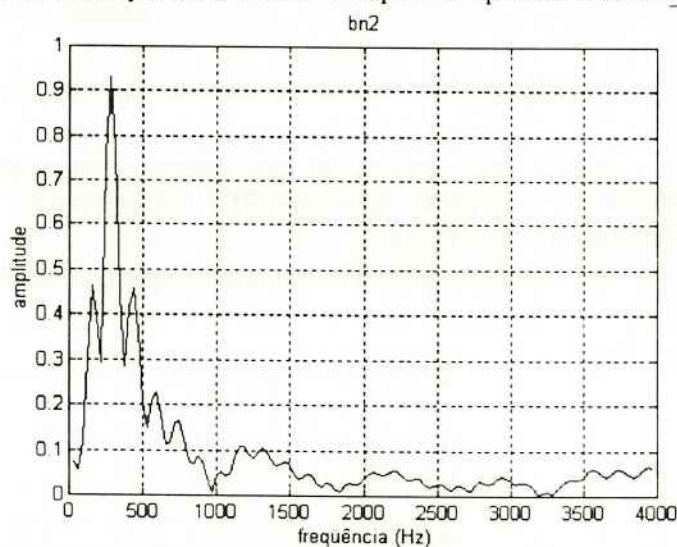


Figura 3.7 - Composição espectral do pulso de abertura.

Nota-se que existe uma forte semelhança na composição espectral dos dois eventos analisados. Baseado nesta semelhança não será feita mais distinção entre abertura e fechamento de contato, os quais serão referenciados como um evento único.

3.5 Análise das amostras pelo método de "clusters"

O método de análise por "clusters" é utilizado quando não se conhece a distribuição espacial das amostras. De uma forma resumida, consiste em efetuar o mapeamento das amostras sobre um plano (no caso bi-dimensional), com o objetivo de

avaliar a distribuição das amostras sobre este plano, buscando assim regiões disjuntas que caracterizem amostras com características em comum. Este mapeamento deve ser escolhido de forma a preservar as características ("features") consideradas relevantes para o problema em questão, e proporcionar uma separação adequada das classes buscadas. Para uma nova amostra que se deseje analisar, efetua-se o cálculo da distância desta amostra à cada um dos centros dos "clusters" (centróides), e comparando-se com a distância máxima a cada centróide, determina-se à qual "cluster" pertence a nova amostra, definindo sua classificação.

Para este problema, foram adotadas como características principais a distribuição espectral e a amplitude relativa entre as amostras. Os vetores representando as amostras, foram obtidos através da aplicação da FFT, e normalizados com relação a amplitude máxima de todas as amostras consideradas (máximo global). Com estes vetores foi calculada a matriz de distâncias entre os vetores, usando a distância Euclidiana, sendo em seguida efetuado o mapeamento destas distâncias sobre o plano bidimensional.

3.5.1 Cálculo das distâncias Euclidianas

Para a visualização da distribuição dos eventos, no espaço Euclidiano, foram calculadas as distâncias de cada amostra em relação às demais amostras. Este cálculo foi efetuado no "MATLAB", utilizando a fórmula :

$$V_1 = (x_1^1, x_2^1, x_3^1, \dots, x_n^1)$$

$$V_2 = (x_1^2, x_2^2, x_3^2, \dots, x_n^2)$$

$$d(1,2) = \sqrt{\sum_{i=1}^n (x_i^1 - x_i^2)^2}$$

com V_1 e V_2 , dois vetores amostra, e $d(1,2)$ igual a distância Euclidiana entre os dois vetores.

Sendo o número de amostras considerado igual a 30, o cálculo destas distâncias, nos conduz ao resultado de uma matriz de distâncias de 30 x 30 elementos.

3.5.2 Determinação dos "clusters" e cálculo do centróide de cada "cluster"

Para a determinação dos "clusters", é necessário traçar o gráfico de distribuição das amostras sobre o plano mencionado. Para efetuar este traçado, foi buscada a maior distância existente na matriz de distâncias calculada anteriormente (maior elemento), e os dois vetores que dão origem a esta distância máxima foram tomados como referência. Tomar a maior distância mostra-se conveniente, pois melhora a precisão da distribuição traçada, permitindo uma melhor separação dos eventuais grupos existentes.

Em função dos vetores tomados como referência, foram traçados os demais pontos por um processo de triangulação, usando as distâncias relativas aos vetores de referência obtendo-se a distribuição das amostras de pulsos válidos.

Observando esta distribuição, foi verificada a existência de agrupamentos ("clusters") de amostras, sendo os componentes de cada grupo facilmente identificáveis, sendo constituídos então os conjuntos de amostras. A seguir para cada conjunto de amostras, foi calculado o centróide do agrupamento.

Para este cálculo, procedeu-se da seguinte maneira. Para um dado "cluster", arbitre-se qualquer amostra, pertencente ao mesmo, como centróide do "cluster". A seguir calcula-se a distância desta amostra às demais amostras do "cluster" em questão. O valor obtido é dividido por N, onde N é o número de eventos que formam o "cluster". A expressão usada para o cálculo da distância média é a seguinte:

$$D_{\text{media}_k} = \frac{1}{N} \sum_{i=1}^N d(k,i) \quad \text{com } N = \text{número de amostras existentes no grupo}$$

e $d(k,i)$ = distância entre as amostras "k" e "i"

Repete-se o processo admitindo uma outra amostra como centro do cluster, e os valores obtidos que representam as distâncias médias entre as amostras são comparados. Caso o valor da média das distâncias obtida no segundo cálculo seja menor do que aquela calculada da primeira vez, a segunda amostra passa a ser o centróide. Este processo é repetido, até que todas as amostras tenham sido tomadas como centro, então quando for terminado o procedimento, a amostra que representar a menor distância média será tomada como centróide do agrupamento.

3.4.3 Identificação dos pulsos

Para cada "cluster", toma-se então a maior distância das amostras do grupo, em relação ao centróide obtido, como raio que define uma região circular em torno deste centróide. A partir para cada amostra que se deseje identificar, é calculada a distância com relação a cada "cluster". Em seguida toma-se a menor distância obtida, e comparamos esta com o raio definido para o "cluster" N, ao qual pertence. Caso a distância seja menor do que o raio da região que define o "cluster", dizemos que a amostra que está sendo analisada pertence ao "cluster" N.

4. Resultados

A seguir apresentamos os resultados obtidos a partir da análise de 30 amostras de pulsos válidos, sendo que 16 correspondem a um aparelho telefônico decádico com sistema de teclas para discagem dos dígitos, e outras 14 amostras correspondem a um aparelho dotado de sistema de disco para discagem dos dígitos.

Para verificação da discriminação de pulsos inválidos, foram usadas 11 amostras correspondentes ao bater de dedos sobre o monofone, e também foram utilizadas 15 amostras correspondendo aos eventos de curto circuito da cápsula e seu retorno, durante a discagem dos dígitos.

A Figura 4.1, apresenta os sinais no tempo correspondentes a discagem do dígito quatro e o seu espectro em frequência - Fig.4.1(a) e (c) - e os sinais no tempo e respectivo espectro em frequência para o bater de dedos sobre o monofone - Fig.4.1(b) e (d). Pode-se verificar através desta figura, a semelhança entre os gráficos que representam os espectros em frequência, embora as amplitudes relativas sejam diferentes (valores normalizados com relação ao máximo global).

Após o processamento das amostras dos pulsos válidos, foi obtida a distribuição de amostras no plano apresentada na Figura 4.2, onde foi verificada a formação de quatro grupos distintos.

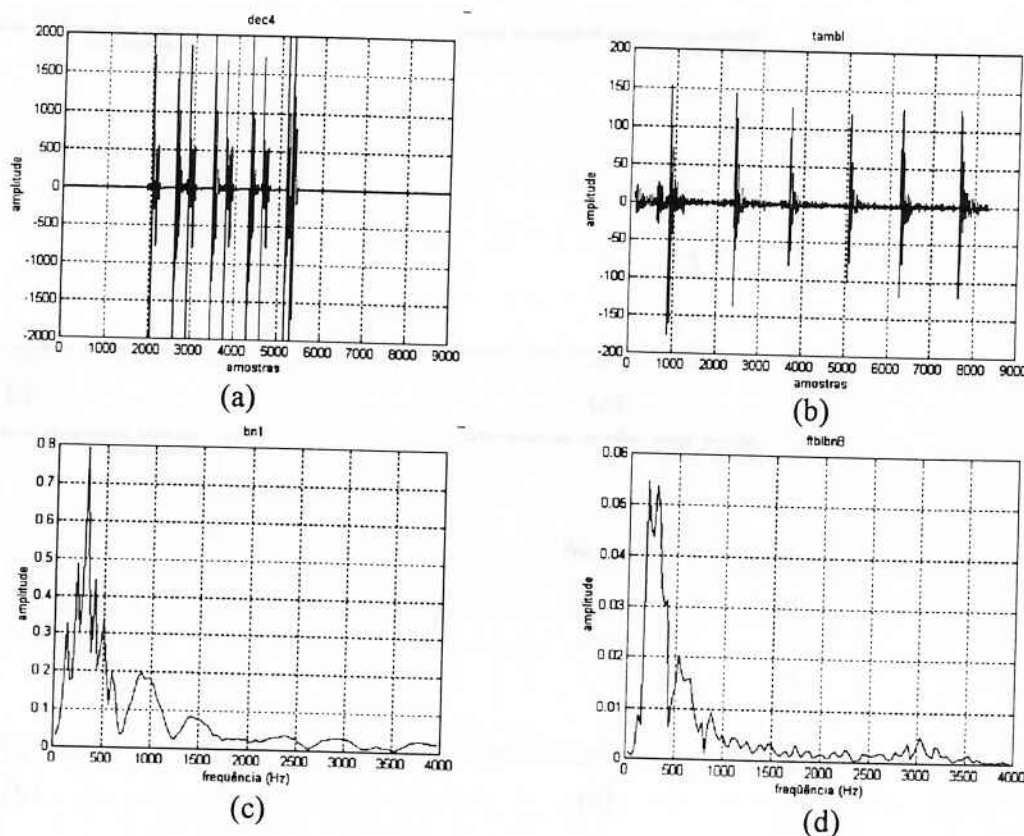


Figura 4.1 - Sinais no tempo e respectivos espectros em frequência, para pulsos válidos e ruído.

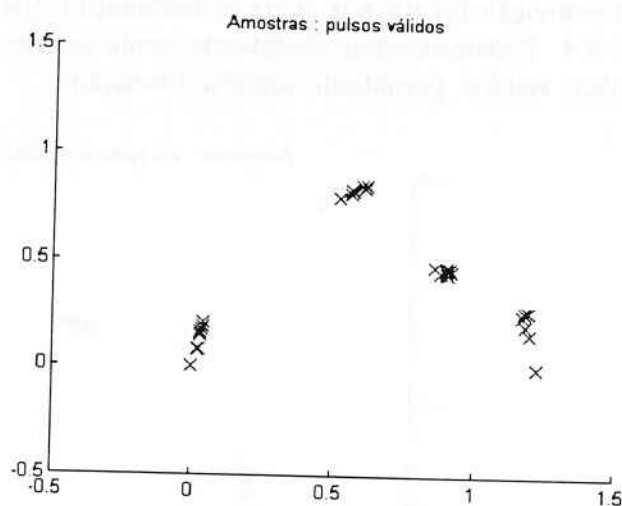


Figura 4.2 - Distribuição das amostras de pulsos válidos no plano.

Efetuada uma verificação das amostras pertencentes a cada agrupamento, foram obtidos os gráficos parciais apresentados na Figura 4.3. Conforme pode-se verificar, os agrupamentos formados obedecem as características do tipo de sistema usado para discagem, e ao tipo de evento que representam, abertura ou fechamento de contato. Assim, apesar de receberem tratamento igual, os eventos foram naturalmente discriminados; para o aparelho dotado de teclado - Fig.4.3(a) e (b) - e para o aparelho dotado de disco - Fig.4.3(c) e (d).

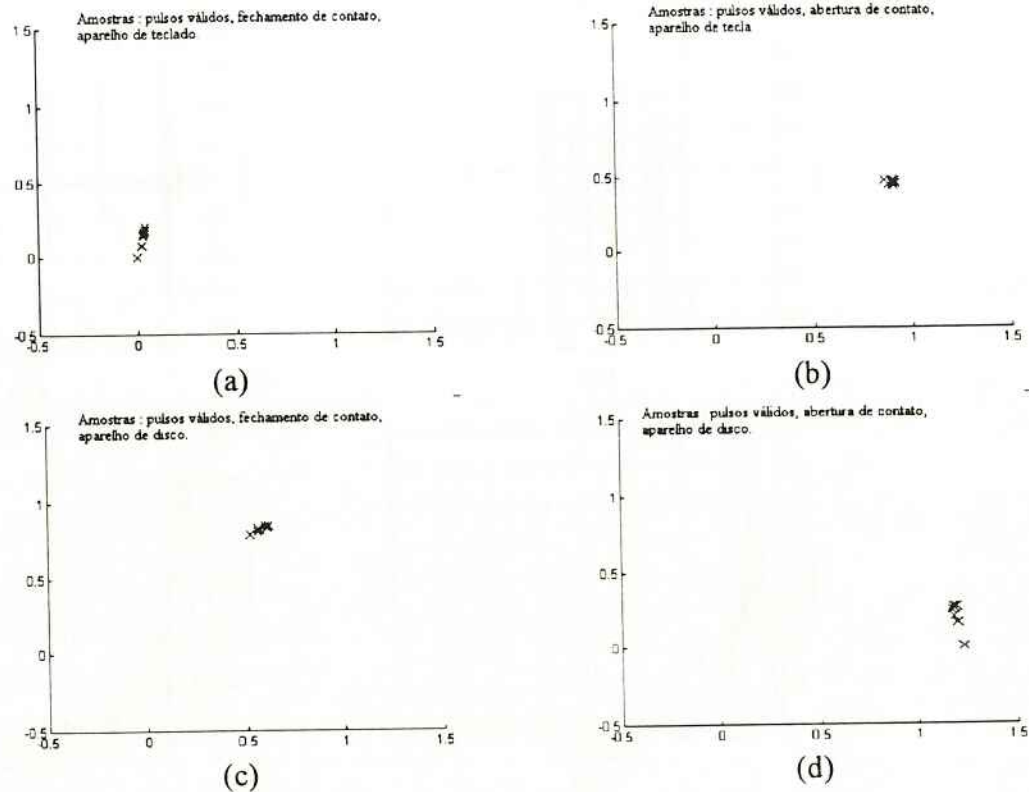


Figura 4.3 - Distribuição parcial das amostras válidas.

A outra distribuição que foi obtida, corresponde aos sinais de ruído considerados. Esta distribuição foi traçada sobre a distribuição dos pulsos válidos e é apresentada na Figura 4.4. Podemos notar claramente como as amostras de ruído situam-se distantes dos pulsos válidos, permitindo sua discriminação.

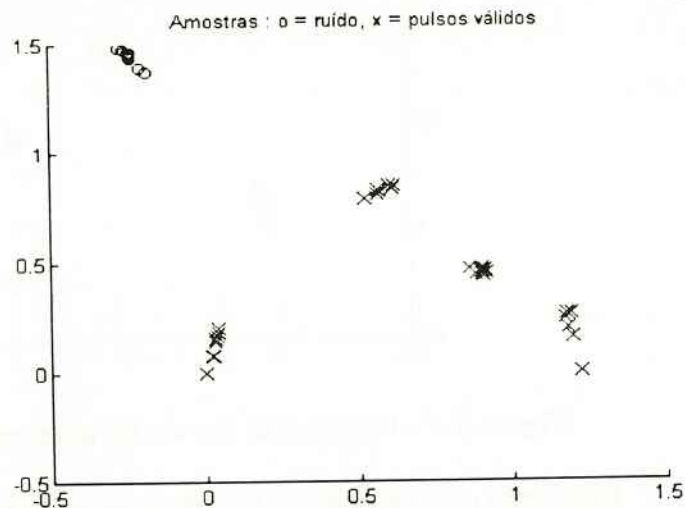


Figura 4.4 - Distribuição do ruído.

Dois outros eventos que também foram considerados, correspondem ao curto circuito e retorno da cápsula microfônica, o que provoca distúrbios sobre a linha. Estes distúrbios são muito semelhantes aos pulsos gerados durante a discagem, e a distribuição

obtida é apresentada na Figura 4.5. O eixo vertical foi escalado para acomodar as distâncias maiores obtidas para os pulsos de retorno da cápsula (* no gráfico).

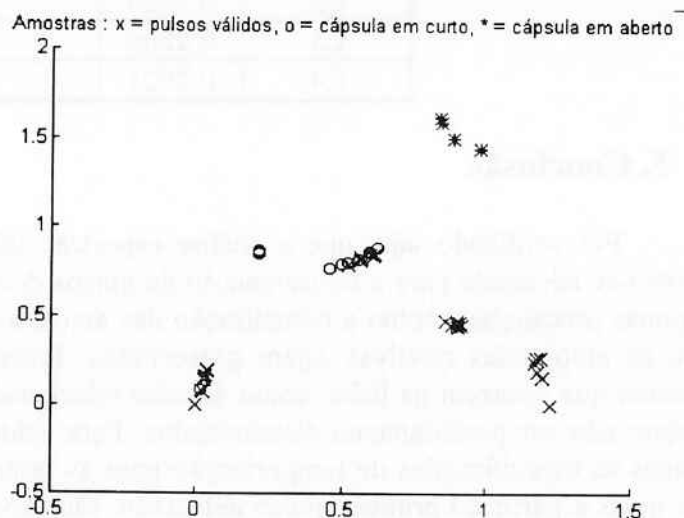


Figura 4.5 - Distribuição dos distúrbios causados pelo curto circuito e retorno da cápsula.

É verificado que os distúrbios causados pelo retorno da cápsula (* no gráfico), são identificados claramente, por estarem a uma boa distância dos pulsos válidos. Por outro lado, os distúrbios causados pelo curto circuito da cápsula do monofone durante a discagem ("o" no gráfico), sobrepõe-se aos pulsos válidos dificultando a sua discriminação. Nota-se também que nestes falsos pulsos não existe uma distinção clara para cada tipo de aparelho telefônico (de teclado ou disco), o que é esperado pois não depende do tipo de mecanismo de discagem, mas apenas das características da cápsula do aparelho e da linha.

A distribuição dos centróides obtidos para os 4 grupos observados, é apresentada na Figura 4.6 e os raios das regiões que delimitam os clusters estão relacionados na tabela.

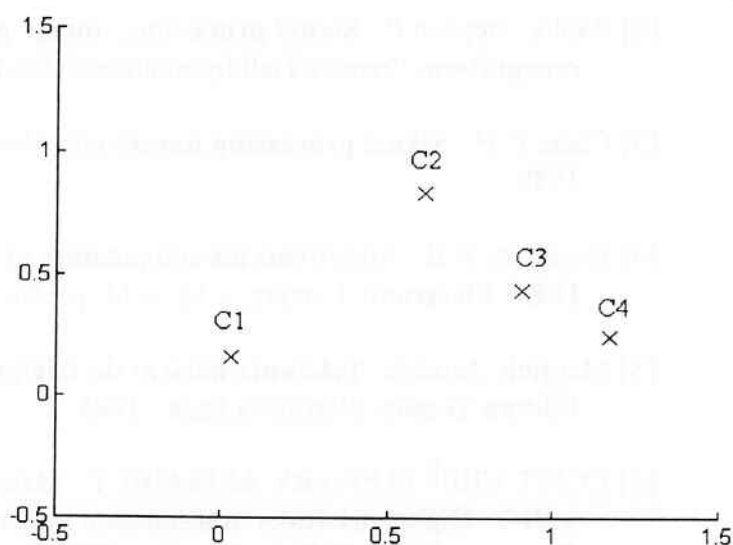


Figura 4.6 - Distribuição dos centróides.

Centróide	Raio	Raio aproximado
C1	0.1735	0.20
C2	0.2026	0.21
C3	0.2266	0.23
C4	0.2921	0.30

5. Conclusão

Foi verificado aqui que a análise espectral, utilizando técnicas de "clustering" mostra-se adequada para a discriminação de pulsos decádicos, desde que sejam tomadas algumas precauções, como a normalização das amostras segundo o máximo global, para que as amplitudes relativas sejam preservadas. Entretanto foi verificado que alguns eventos que ocorrem na linha, como aqueles relacionados com a cápsula do monofone, podem não ser perfeitamente discriminados. Para solucionar este problema, devem ser usadas as especificações de temporização para as sequências de pulsos, criando janelas nas quais a partir do primeiro pulso detectado, são esperados os próximos pulsos e com isso descartar aqueles que ocorram fora da janela de tempo padronizada; poderão também ser usados também métodos de verificação da energia contida nestes pulsos.

Observou-se também que os pulsos válidos apresentam padrões, conforme o tipo de aparelho telefônico utilizado, sendo necessário uma coleta de dados abrangendo o maior número de tipos de aparelho possível, e uma quantidade considerável de amostras de forma a permitir a avaliação estatística da eficácia deste método de detecção.

Finalizando, cabe citar que este trabalho está sendo continuado pelo mesmo grupo envolvido nos estudos aqui apresentados, e o próximo passo consistirá no aperfeiçoamento do projeto utilizando técnicas modernas de análise com o emprego de redes neurais artificiais.

6. Bibliografia

- [1] Oppenheim, A.V.; Schafer, R.W. **Discrete-Time Signal Processing**. New Jersey, Prentice Hall, Inc., 1989.
- [2] Banks, Stephen P. **Signal processing: image processing and pattern recognition**. Prentice Hall International (UK) Ltd, 1990.
- [3] Chen, C.H. **Signal processing handbook**. New York, Marcel Dekker, Inc., 1988.
- [4] Urquhart, R.B. Algorithms for computation of relative neighbourhood graph. **IEEE Electronic Letters**, v.16, n.14, p.556-557, Jul. 1980.
- [5] Megrich, Arnaldo **Telefonia básica: do telefone às microondas**. Editele Editora Técnica Eletrônica Ltda., 1985.
- [6] CCITT VIIIth PLENARY ASSEMBLY, Málaga, 1984. Red Book - v.III, n.III.3, Digital networks: transmission systems and multiplexing equipment. **Recomendation G.711**. Genebra, CCITT, 1985.
- [7] **MATLAB user's guide**. The Mathworks, Inc., Natick, Mass., Feb. 1993.

ANEXOS

A seguir são apresentadas as listagens dos programas utilizados neste trabalho.

I. Programa em "C" para descompressão das amostras, comprimidas segundo a lei A.

```

/*
 * Programa para expansão das amostras PCM comprimido segundo a lei "A"
 *
 * Data: 23/Abr/94
 */

#include <stdio.h>
#include <stdlib.h>
#include <dir.h>
#include <math.h>

int expande(char hexval);          // Protótipo da função

main(void)
{
    char in[MAXPATH], out[MAXPATH];
    FILE *mat, *hex;
    char num;

    printf("H2MAT converte arquivo .HEX em arquivo ASCII .MAT\n\nDigite o nome do arquivo de entrada\n");
    gets( in );

    puts("Digite o nome do arquivo de saída");
    gets( out );

    hex = fopen( in, "rb");          // Abre o arquivo p/leitura em modo binário
    mat = fopen( out, "wt");         // Abre o arquivo p/escrita no modo texto

    while(fread( &num, sizeof(char), 1, hex)) fprintf( mat, "%d ", expande( num));

    fcloseall();

    return 0;
}

int expande(char hexval) {
    int valor;                      // Valor expandido (a retornar)
    int pat;                        // 2^(seg+3)
    int sinal;

    sinal = hexval & 0x80;          // Polaridade
    hexval &= 0x7f;                 // valor absoluto

    pat = hexval >> 4? (int) pow(2, (( hexval >> 4) + 3)): 0;

```

```

        if (pat) valor = pat/16*(hexval & 0x0f);
        else valor = hexval & 0x0f;

        valor += pat;

        if (sinal) return (-valor);
        return valor;
    }

```

II. Programa em "C" para conversão das amostras descomprimidas para arquivo binário, compatível com o MATLAB.

```

/* Programa conversor de formato.
 *
 * Arquivo de entrada: representação ASCII de amostras digitalizadas pelo
 * 29C13, comprimidos segundo a lei "A".
 *
 * Arquivo de saída: representação binária pura das amostras.
 *
 * Data: 22/Abr/94
 */

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <dir.h>

main(int argc, char *argv[])
{
    char in[MAXPATH], out[MAXPATH];
    char drive[MAXDRIVE], dir[MAXDIR];
    char fname[MAXFILE], ext[MAXEXT];
    FILE *txt, *hex;
    unsigned char h[16];

    if (argc < 2) {
        printf("Converte arquivos de amostras .TXT num arquivo absoluto .HEX.\nUso: D2HEX\nfilespec[.TXT]\n\n");
        return 1;
    }

    strcpy( in, argv[1] );
    fnsplit( in, drive, dir, fname, ext );

    if (!strcmp( ext, "")) strcpy( in, ".TXT" );

    strcpy(out, fname);

    strcat(out, ".HEX" );

    if( (txt = fopen( in, "rt")) == NULL ) {
        printf("Não foi possível abrir o arquivo %s\n", in);
        return 2;
    }
}

```

```

if ( (hex = fopen(out, "wb") ) == NULL ) {
    printf("No foi possvel abrir o arquivo %s\n", out);
    return 2;
}

while ( EOF != fscanf( txt, "%X %X %X %X %X %X %X %X %X %X %X %X %X %X %X",
    &h[0], &h[1], &h[2], &h[3],
    &h[4], &h[5], &h[6], &h[7],
    &h[8], &h[9], &h[10], &h[11],
    &h[12], &h[13], &h[14], &h[15]) )

    fwrite(h, sizeof(char), 16, hex);

fcloseall();
return 0;
}

```

III. Rotina em MATLAB para visualizar um arquivo de dados no tempo.

% Rotina para plotar um arquivo de dados, com os respectivos
% comentrios

```

fnameat = input('Nome do arquivo : ','s');
load (fnameat);
inicio = input('Fornea o inicio do intervalo : ');
tamanho = input('Fornea o tamanho do intervalo : ');
final = inicio + tamanho - 1;
x=inicio:1:final;
eval(['plot(x,','fnameat,','(x)')]);
grid
title (fnameat)
xlabel('amostras')
ylabel('amplitude')

```

IV. Rotina em MATLAB para visualizar um arquivo de dados em frequncia.

% Rotina para plotar o resultado da fft dos dados, com os respectivos
% comentrios

```

fnamefft = input('Nome do arquivo : ','s');
load (fnamefft);
inicio = 1;
tamanho = input('Fornea o tamanho do intervalo da FFT : ');
final = ((inicio + tamanho)/2) - 1;
x=inicio:1:final;
xn=(4000/(tamanho/2))*x;
eval(['plot(xn,','fnamefft,','(x)')]);
grid
title (fnamefft)
xlabel('frequncia (Hz)')
ylabel('amplitude')

```

V. Rotina em MATLAB para gerar os arquivos FFT das amostras.

```
% Gera arquivos contendo o espectro de frequências dos arquivos
% de amostras [NAME1..n], salvando nos arquivos [FNAME1..n] (todos
% arquivos são .MAT)

% Efetua a leitura do arquivo com fnamedat=[NAME]

fnamedat = input('Nome do arquivo : ','s');
load (fnamedat);

% Efetua o cálculo da fft de cada arquivo segundo o intervalo fornecido

numarq = 1;
comeco = 0;
fnamefft = input('Nome do(s) arquivo(s) de saída : ','s');
tamanho = input('Forneça o tamanho do intervalo a tratar : ');
while (comeco + tamanho)>0;
    comeco = input('Forneça o inicio do intervalo : ');
    final = comeco + tamanho - 1;
    x = comeco:1:final;
    eval([fnamefft,int2str(numarq),' = abs(fft('',fnamedat,'(x)));']);
    eval(['save ',fnamefft,int2str(numarq),' ',fnamefft,int2str(numarq)]);
    numarq = numarq + 1;
end

% Efetua a leitura sequencial de n arquivos, toma metade dos
% elementos, gerando outro conjunto de arquivos
% Arquivos de entrada an.MAT
% Arquivos de saída bn.MAT

for i = 1:30;
    eval(['load a',int2str(i)]);
end

for i = 1:30;
    for j = 1:128;
        eval(['b',int2str(i),'(',int2str(j),')=','a',int2str(i),'(',int2str(j),')']);
    end
end

for i = 1:30;
    eval(['save b',int2str(i),' ','b',int2str(i),'.']);
end

% Calcula o máximo global das 30 amostras
% Efetua a leitura sequencial de n arquivos, toma o máximo de
% cada vetor, gerando um vetor de máximos.
% Em seguida é calculado o máximo deste vetor, obtendo o máximo global
% Arquivos de entrada bn.MAT
% Gera internamente o vetor "vmax(i)"
% Saída é o valor "maxglobal"

% Carrega os vetores de amostras
```



```

for i = 1:30;
    eval(['load b',int2str(i)]);
end

% Calcula o máximo de cada vetor individualmente
for i = 1:30;
    eval(['bmax',int2str(i),'= max (b',int2str(i),';)']);
end

% Compõe o vetor de máximos, contendo todas os máximos calculados
for i = 1:30;
    eval(['vmax(',int2str(i),') = bmax',int2str(i),';)']);
end

% Apresenta o máximo global do conjunto de vetores
maxglobal = max (vmax)

% Efetua a normalização das 30 amostras, pelo máximo global.
% Efetua a leitura sequencial de n arquivos, e utilizando o valor de
% máximo global, gera um novo conjunto de vetores normalizados
% (amplitudes entre 0 e 1)
% Em seguida salva estes novos vetores.
% Arquivos de entrada bn.MAT com N=1..30
% Utiliza o valor "maxglobal"
% Saída são os vetores bn.MAT com N=1..30

% Carrega os vetores de amostras
for i = 1:30;
    eval(['load b',int2str(i)]);
end

% Efetua a normalização por "maxglobal"
for i = 1:30;
    eval(['bn',int2str(i),'= b',int2str(i),'/maxglobal;']);
end

% Salva os novos vetores normalizados pelo máximo global
for i = 1:30;
    eval(['save bn',int2str(i),' bn',int2str(i)]);
end

```

VI. Rotina em MATLAB para gerar a matriz de distâncias das amostras.

```

% Efetua o cálculo da matriz de distâncias entre as 30 amostras.
% usando a fórmula de distância Euclidiana.
% Efetua a leitura sequencial de n arquivos.
% Calcula cada elemento da matriz de distâncias "d(x,y)".
% Em seguida salva a matriz "d".
% Arquivos de entrada bn.MAT com N=1..30
%

```

```

% Saida é a matriz de distâncias "d".

% Carrega os vetores de amostras

for i = 1:30;
    eval(['load bn',int2str(i)]);
end

% Efetua o cálculo das distâncias entre os vetores de amostras.

for x = 1:30;
    for y = 1:30;
        eval(['m = bn',int2str(x),'- bn',int2str(y),';']);
        n = m.^2;
        d(x,y) = sqrt (sum(n));
    end
end

% Salva a matriz "d", no formato ".MAT".

save d d

```

VII. Rotina em MATLAB para traçar o gráfico de distribuição das amostras no plano.

```

% Programa para plotar a distância relativa entre os vetores.
% sobre o plano.
% A matriz de distâncias deve ter sido previamente calculada.
% e estar carregada em memória.

```

```

hold on          % permite plots sucessivos sem limpar a tela gráfica

```

```

x=-0.5;
y=-0.5;
plot(x,y);
x=1.5;
y=1.5;
plot(x,y);

```

```

deltax = 0;      % inicializa off-set do gráfico no eixo X
deltay = 0;      % inicializa off-set do gráfico no eixo Y

```

```

i=5;    % índice de um dos vetores de distância máxima
j=24;   % índice do outro vetor de distância máxima

```

```

for m=15;

```

```

    XM = (d(i,j)^2 + d(i,m)^2 - d(j,m)^2) / (2 * d(i,j)); % calcula pos. X
    YM = sqrt(d(i,m)^2 - XM^2); % calcula pos. Y
    plot(deltax+XM,deltay+YM,'x');

```

```

end

```

VIII. Rotina para cálculo do centróide de um grupo de amostras.

```

% Estimativa do centro de um cluster (método K-Means).
% 1. Toma a primeira amostra do grupo como centro estimado.

```

```

% 2. Calcula a distância de todos os vetores do grupo com relação ao
% suposto centro.
% 3. Calcula a média das distâncias obtidas M1=média das distâncias.
% 4. Toma a segunda amostra do grupo como centro.
% 5. Repete o passo 2.
% 6. Calcula a média das distâncias obtidas M2.
% 7. Compara M1 com M2, se M1 <= M2 mantém o primeiro centro
% se M1 > M2 adota o segundo como novo centro
% 8. Repete os passos 4, 5, 6 e 7 para todas as amostras.

% Carrega os vetores de amostras

for i = 1:30;
    eval(['load bn',int2str(i)]);
end

% Efetua o cálculo da matriz de distâncias entre as amostras, usando a
% fórmula de distância Euclidiana.
% Calcula cada elemento da matriz de distâncias "D(x,y)".
% Arquivos de entrada bnN.MAT com N=1..30
%
% Saida é o número da amostra considerada como centro e a maior distância
% do conjunto de amostras ao centro considerado.

% Efetua o cálculo das distâncias entre os vetores de amostras.

clear A;
clear G;
clear D;

for x = 1;
    for y = 1:2:16;
        eval(['m = bn',int2str(x),'- bn',int2str(y),'']);
        n = m.^2;
        D(x,y) = sqrt (sum(n));
    end
end

M=(sum (sum(D))) / 7; % calcula a média das distâncias
Y1 = x; % indica qual vetor está sendo considerado centro
clear D;

for x = 1:2:16
    for y = 1:2:16;
        eval(['m = bn',int2str(x),'- bn',int2str(y),'']);
        n = m.^2;
        D(x,y) = sqrt (sum(n));
        A(x,y)=D(x,y);
    end
    M2 = (sum(sum(D))) / 7;
    clear D;
    Y2 = x;
    if M > M2
        M = M2;
        Y1 = Y2;
    end
end

M

```

```
for x=1:2:16
G(x)=A(x,Y1);
end
Y1
R=max(G)
end
```


BOLETINS TÉCNICOS - TEXTOS PUBLICADOS

- BT/PEE/93-01 - Oscilador a HEMT - 10 GHz - FÁTIMA S. CORRERA, EDMAR CAMARGO
- BT/PEE/93-02 - Representação Senoidal da Voz através dos Polos do Filtro Preditor - MARCELO B. JOAQUIM, NORMONDS ALENS
- BT/PEE/93-03 - Blindagens por Grades Conductoras: Cálculo do Campo Próximo - LUIZ CEZAR TRINTINALIA, ANTONIO ROBERTO PANICALI
- BT/PEE/93-04 - Sistema de Otimização e Controle de Produção em Minas de Pequeno e Médio Porte - TSEN CHUNG KANG, VITOR MARQUES PINTO LEITE
- BT/PEE/94-01 - Determinação das Frases de Aplicação Forense para o projeto NESPER e Tese de Mestrado IME/94, com Base em Estudos Fonéticos - MARCONI DOS REIS BEZERRA, EUVALDO F. CABRAL JUNIOR
- BT/PEE/94-02 - Implementação e Teste de uma Rede Neural Artificial do Tipo KSON (Kohonen Self-Organizing Network) com Entradas Bidimensionais - MARCELO YASSUNORI MATUDA, EUVALDO F. CABRAL JR.
- BT/PEE/94-03 - Transformada de Walsh e Haar Aplicadas no Processamento de Voz - ALEXANDRE AUGUSTO OTTATI NOGUEIRA, THIAGO ANTONIO GRANDI DE TOLOSA, EUVALDO F. CABRAL JÚNIOR
- BT/PEE/94-04 - Aplicação de Redes Neurais ao Problema de Reconhecimento de Padrões por um Sonar Ativo - ALEXANDRE RIBEIRO MORRONE, CRISTINA COELHO DE ABREU, EDUARDO KOITI KIUKAWA, EUVALDO F. CABRAL JR.
- BT/PEE/94-05 - Tudo que se Precisa Saber sobre a Prática da FFT - Transformada Rápida de Fourier (Inclui Software) - ROGÉRIO CASAGRANDE, EUVALDO F. CABRAL JR.
- BT/PEE/94-06 - A Survey on Speech Enhancement Techniques of Interest to Speaker Recognition - CELSO S. KURASHIMA, EUVALDO F. CABRAL JR.

