Simulador de Reconhecedores de Palavras Isoladas

ALI HUSSEIN SAYED
NORMONDS ALENS

Neste trabalho é apresentado um simulador de reconhecedores de comandos da fala [1]. O similador permite testar vários algoritmos propostos na literatura e variações destes algoritmos.

1. INTRODUÇÃO

A área de reconhecimento de voz preocupa-se em prover meios de comunicação com as máquinas através da fala humana. Estudos recentes mostram que a comunicação através da fala é o meio preferido pelas pessoas e que tenderá a ser uma alternativa barata para a entrada de dados em computadores [2, 3]. O simulador desenvolvido neste trabalho abrange a área de reconhecimento de comandos da fala. O sistema opera com vocabulário limitado e deve ser previamente treinado. Testes realizados com o auxílio do simulador desenvolvido, para uma configuração freqüente na literatura, (ver seção 19), forneceram taxas de acerto entre 95% e 98%.

2. APARELHO FONADOR HUMANO

O primeiro passo no processamento do sinal de voz consiste em modelar o aparelho fonador humano. O aparelho fonador constitui-se de um tubo acústico que se estende entre a glote e os lábios. A cavidade nasal pode ser acoplada ou não à cavidade oral através da movimentação da úvula. O tubo acústico possui características e dimensões variáveis em função da pessoa e do som emitido [11]. Os sons classificam-se em sons vocálicos (ex. vogais), sons oclusivos (/p/, /t/), sons fricativos (/s/, /f/) e sons nasais (/n/, /m/) [11, 12].

Ali Hussein Sayed é professor assistente, EPUSP e Normonds Alens professor titular, EPUSP
DEE – EPUSP – CP 8174 – SÃO PAULO, SP – 05508 – Julho 1989

A figura 1 mostra o sinal acústico do dígito "oito", onde pode-se observar as regiões vocálicas e as regiões de "silêncio" da palavra. O termo "silêncio" refere-se ao ruído de fundo do ambiente acústico do reconhecedor.

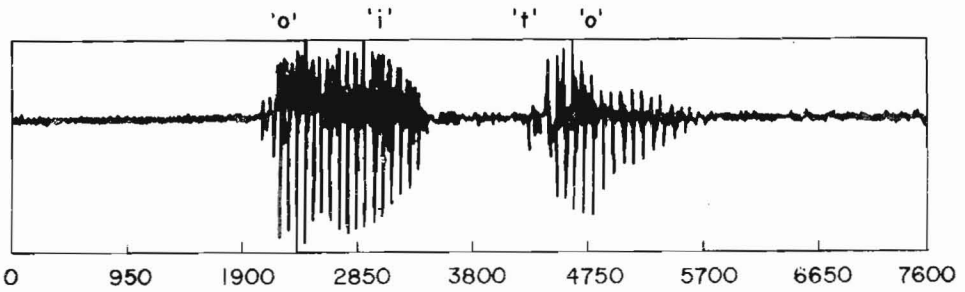

Figura 1. Sinal acústico do dígito "oito". São 7600 amostras relativas a uma taxa de amostragem de 8 KHz.

Resultados experimentais mostram que as características do sistema de geração do sinal de voz mudam lentamente. A mudança ocorre em períodos em torno de 10 a 30ms [8, 12, 13, 14, 15]. A **Figura 2** mostra um intervalo do som "oi" do dígito "oito". Percebe-se que o sinal não apresenta mudanças bruscas de comportamento. Pode-se, então, modelar o aparelho fonador humano por um sistema *linear e lentamente variável* com o tempo [12, 13], que pode ser excitado tanto por um trem de impulsos quasi-periódicos (no caso de sons vocálicos) ou por ruído branco (no caso de sons não-vocálicos), conforme **Figura 3**. Filtrando o sinal de voz proveniente do microfone com um filtro L(z) do tipo:

$$L(z) = 1 - \mu z^{-1}$$

obtém-se um sistema global V(z), com os efeitos da radiação nos lábios e da variação da área da glote reduzidos. A este processo de tratamento do sinal de voz, dá-se o nome de *pré-ênfase*, conforme **Figura 4**. O parâmetro μ é denominado *fator de pré-ênfase* e o seu valor pode ser programado pelo SIR PI (SImulador de Reconhecedores de Palavras Isoladas) (valores típicos de μ são próximos de 1,0).

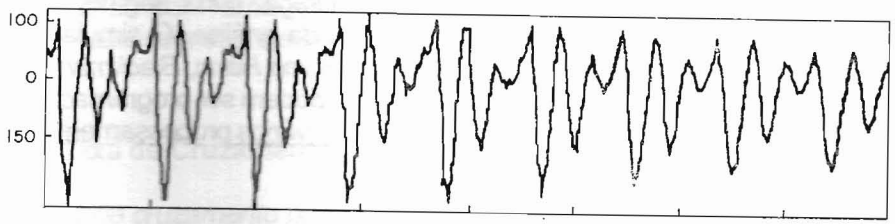

Figura 2. Intervalo do som "oi" do dígito "oito"

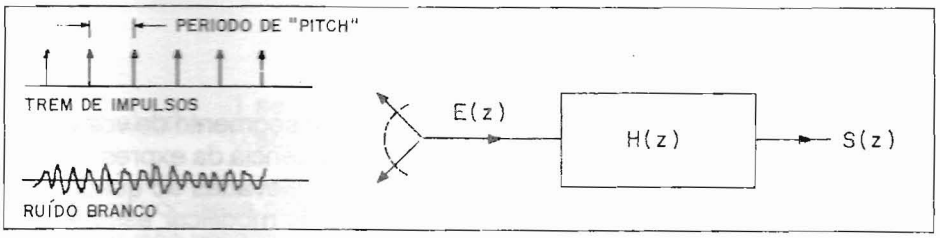

Figura 3. Modelo linear e lentamente variável (variável quadro a quadro).

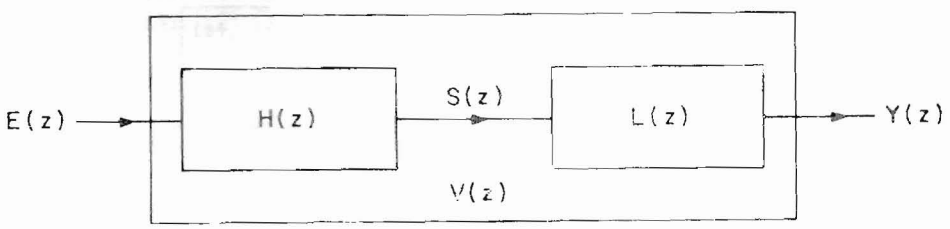

Figura 4. Modelo com pré-ênfase

Tal modelo, adotado neste trabalho, tem-se mostrado eficiente e funciona bem na prática [8].

3. PROCESSAMENTO TEMPORAL DO SINAL DE VOZ

O sinal de voz pode ser analisado em intervalos de duração curta, segmentando-o utilizando janelas de duração igual à duração da análise. O simulador permite escolher entre as janelas do tipo *Retangular, von Hann, Blackmann e Bartlett* [16]. A largura N e o deslocamento D da janela podem ser programados. Após o janelamento, o segmento de voz é submetido a vários processamentos:

3.1. Medida de Energia de Tempo Curto

A amplitude dos sons não-vocálicos é bem menor que a amplitude dos sons vocálicos. A "*energia" quadrática de tempo curto,* E(n), fornece uma boa representação desta variação de amplitude [12, 15]:

$$E(n) = \sum_{m=0}^{N-1} | y(n-m)\, w(m) |^2 \qquad (2)$$

onde w(m) é uma janela de largura N e y(n - m) é um segmento de voz entre n - N + 1 e n para $0 \leq m \leq N-1$. A principal inconveniência da expressão (2) é o destaque dado às grandes amplitudes (que são elevadas ao quadrado) em relação aos níveis mais baixos. Uma maneira de modificar esse efeito seria utilizar a medida de "*energia" absoluta de tempo curto,* M(n):

$$M(n) = \sum_{m=0}^{N-1} | y(n-m)\, w(m) | \qquad (3)$$

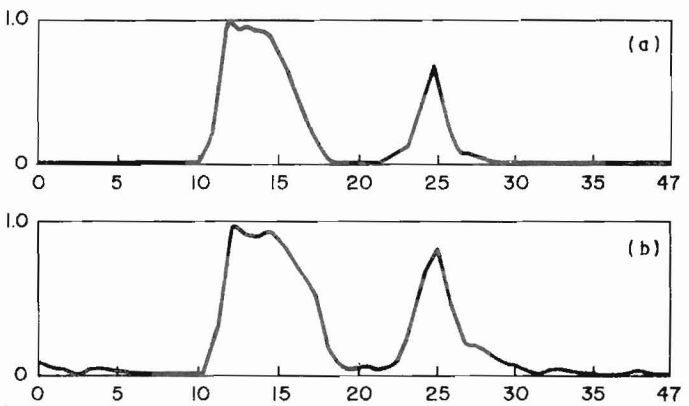

Figura 5. "Energias" quadrática (a) e absoluta (b) do dígito "oito" em função de segmentos de 20ms (N = 160 amostras).

O simulador permite escolher entre a medida absoluta e a medida quadrática. A **Figura 5** mostra as "energias" absoluta e quadrática do dígito "oito", calculadas com uma janela retangular de largura N = 160 amostras e deslocada sem sobreposições (D = 160).

3.2. Taxa de Cruzamento por Zero de Tempo Curto

A *taxa de cruzamento por zero,* z(n), ao longo do sinal de voz, fornece uma idéia robusta do conteúdo espectral do sinal [12, 15, 17]. Z(n) é definida por:

$$Z(n) = \sum_{m=0}^{N-1} | \text{sinal}\{y(n-m)\} - \text{sinal}\{y(n-m-1)\} | \, w(m) \qquad (4a)$$

onde

$$\text{Sinal}(x) = \begin{cases} 1 \text{ se } x \geq 0 \\ 0 \text{ se } x < 0 \end{cases} \qquad (4b)$$

As regiões vocálicas possuem taxas de cruzamento por zero menores que as regiões não vocálicas. A **Figura 6** mostra a taxa de cruzamento de zero do dígito "oito", calculada com uma janela retangular de largura N = 160 amostras e deslocada sem sobreposições.

Deve-se mencionar que o nível DC e interferências de 60 Hz prejudicam o cálculo de Z(n) [12]. O simulador SIR PI permite controlar o nível DC do sinal de voz.

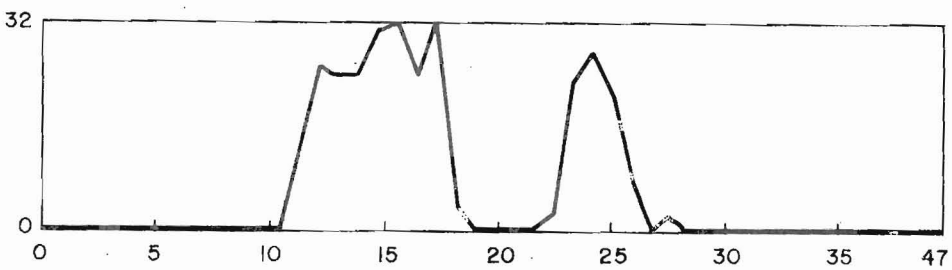

Figura 6. Taxa de cruzamento de zero do dígito "oito" em função de segmentos de 20ms (N = 160 amostras)

3.3. Função de Autocorrelação de Tempo Curto

A função de autocorrelação de tempo curto, R(k), é utilizada na análise do sinal de voz pela técnica de *predição linear*. O *coeficiente de autocorrelação de índice k* é dado por [12].

$$v_n(m) = y(n - m)\, w(m), \quad 0 \le m \le N - 1 \tag{5a}$$

$$R(k) = \sum_{m=0}^{N-1-k} v_n(m)\, v_n(m + k) \tag{5b}$$

onde $v_n(m)$ é um segmento janelado do sinal de voz, de lagura N. As amostras que compõem $v_n(m)$ vão de y(n - N + 1) a y(n).

4. PONTOS EXTREMOS DA PALAVRA

A localização dos pontos extremos do comando pronunciado, é um fator importante que permite reduzir o número de amostras a serem analisadas pelo sistema e *que afeta consideravelmente a sua taxa de acerto de reconhecimento.* O algoritmo empregado foi sugerido por Rabiner e Sambur [18] e caracteriza-se pela sua simplicidade. O algoritmo emprega dois parâmetros: a "energia" de tempo curto, E(n) ou M(n), e a taxa de cruzamento por zero, Z(n), do sinal de voz. Estes parâmetros são utilizados na determinação de limiares que controlam o processo de decisão. São definidos dois limiares de energia L_{es} e L_{ei} (superior e inferior respectivamente) e um limiar de cruzamento de zero L_{zcr}. Uma vez determinados os limiares, o algoritmo parte para a localização dos extremos. O algoritmo realiza a busca a partir do início da palavra segundo o critério de energia. O segmento N1, em que a energia excede o limiar inferior L_{ei} e continua subindo até exceder L_{es}, é adotado como estimativa inicial para o começo da palavra. O mesmo procedimento é repetido para a licalização da estimativa N2 do final da palavra. As estimativas N1 e N2 são então corrigidas de acordo com o critério da taxa de cruzamento de zero. Neste caso, parte-se de N1 em direção ao início da palavra e determina-se o número de vezes em que Z(n) ultrapassa L_{zcr} em α segmentos anteriores a N1. Se este número exceder 3, a estimativa inicial N1 é deslocada até o primeiro ponto, *no tempo*, em que Z(n) ultrapassa L_{zcr}. O mesmo procedimento é repetido para β segmentos posteriores à estimativa N2. O número de segmentos α e β devem ser escolhidos adequadamente (valores típicos são $\alpha = \beta = 25$). O simulador SIR PI permite controlar o cálculo dos

limiares e portanto, permite controlar a precisão do algoritmo. Aplicando o algoritmo ao dígito "oito" (janela retangular com N = 160 e deslocamento sem sobreposição), resultam os pontos extremos (N1 = 10 e N2 = 30) indicados na figura 7.

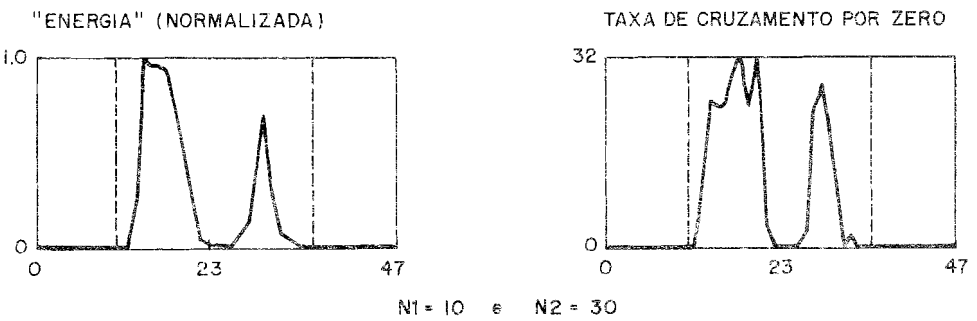

Figura 7. Pontos extremos do dígito "oito"

5. PREDIÇÃO LINEAR

A análise da voz por *predição linear*, baseia-se na hipótese de que uma amostra do sinal de voz pode ser aproximada por uma combinação linear das amostras anteriores ou das amostras futuras [14]. Uma predição de ordem P, procura estimar a amostra $y_w(n)$, do instante n, a partir de P amostras anteriores. O índice w indica janelamento da amostra y(n). Pode-se, também, estimar a amostra $y_w(n - P)$, do instante n - P, a partir de P amostras futuras:

$$\bar{y}(n) = \sum_{k=1}^{P} \bar{a}_k y_w (n - k) \qquad (6a)$$

$$\bar{\bar{y}} (n - P) = \sum_{k=1}^{P} \bar{\bar{a}}_k y_w(n - P + k) \qquad (6b)$$

As expressões (6a) e (6b) permitem definir os *erros de predição futura*, $f_p(n)$, e de *predição passada*, $b_p(n)$, de origem P [12, 14]:

$$f_p(n) = y_w(n) - \sum_{k=1}^{P} \bar{a}_k\, y_w(n-k) \tag{7a}$$

$$b_p(n) = y_w(n-P) - \sum_{k=1}^{P} \bar{\bar{a}}_k\, y_w(n-P+k) \tag{7b}$$

Os *erros quadráticos médios totais de predição futura e de predição passada*, de ordem P, ξ_f^p e ξ_b^p respectivamemte, em um intervalo de largura N, são definidos pelas expressões abaixo:

$$\xi_f^p = \sum_{n=0}^{N-1} f_p^2(n) \tag{8a}$$

$$\xi_b^p = \sum_{n=0}^{N-1} b_p^2(n) \tag{8b}$$

A idéia da predição linear é obter o conjunto de coeficientes $\{\bar{a}_k\}$ ou $\{\bar{\bar{a}}_k\}$ que sirva como estimativa dos coeficientes $\{a_k\}$ do sistema V(z), que modela a geração do sinal de voz. Convém mencionar que V(z) pode ser representado, adequadamente, por um modelo composto apenas por pólos [13]:

$$V(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{9}$$

onde G é um fator de ganho. A presença de zeros em (9) melhoraria a representação de sons nasais. O denominador da expressão (9) é denominado *filtro inverso* relativo aos coeficientes $\{a_k\}$. Existem vários algoritmos propostos para a obtenção das estimativas $\{\bar{a}_k\}$ ou $\{\bar{\bar{a}}_k\}$. O critério utilizado é o de minimização do erro quadrático médio total. Pode-se mostrar que os coeficientes $\{\bar{\bar{a}}_k\}$ são iguais aos coeficiente $\{\bar{a}_k\}$, mas com a ordem invertida [14]. Por isso, os coeficientes serão referenciados por $\{\bar{a}_k\}$ no decorrer deste trabalho e serão denominados de coeficientes LPC ("Linear Prediction Coefficients"). As opções de cálculo dos coeficientes $\{\bar{a}_k\}$, oferecidas pelo SIR PI, conforme **Figura 8**, são:

Método da Autocorrelação – Algoritmo de Durbin

Métodos Lattice:

Método "Forward" – Minimização do Erro de Predição Futura

Método "Backward" – Minimização do Erro de Predição Passada

Método de Itakura

Método do Mínimo

Método de Burg

5.1. Método da Autocorrelação – Algoritmo de Durbin

O algoritmo baseia-se na minimização do erro quadrático total de predição futura ξ_f^P. Derivando ξ_f^P em relação aos coeficientes $\{\bar{a}_k\}$ e igualando a zero, resulta um sistema linear de equações, de ordem P:

$$\sum_{k=1}^{P} \bar{a}_k R(i-k) = R(i) \quad i = 1, 2, \ldots, P \tag{10}$$

onde R(i) é o coeficiente de autocorrelação de ordem i. O *algoritmo de Durbin* permite uma solução iterativa deste sistema de equações [12]:

$$\xi_{fn}^0 = R(0)$$

$$k_i = \frac{R(i) - \sum_{j=1}^{i-1} \bar{a}_j^{(i-1)} R(i-j)}{\xi_{fn}^{i-1}}, \quad 1 \le i \le P \tag{11a}$$

$$\bar{a}_i^{(i)} = k_i \tag{11b}$$

$$\bar{a}_j^{(i)} = \bar{a}_j^{(i-1)} - k_i \bar{a}_{i-j}^{(i-1)}, \quad 1 \le j \le i-1 \tag{11c}$$

$$\xi^{i}_{fn} = (1 - k_i^2)\, \xi^{i-1}_{fn} \qquad (11d)$$

$$G = \sqrt{R(0) - \sum_{k=1}^{P} \bar{a}_k R(k)} \qquad (11e)$$

Os coeficientes $\{ \bar{a}_j^{(i)} \}$, j = 1, 2, ..., i; são os coeficientes do preditor de ordem i. R(i) são os coeficientes de autocorrelação *normalizados* em relação à energia do sinal R(0). Os coeficientes { k_i } são denominados *coeficientes de reflexão*. O termo ξ^{i}_{fn} é o erro quadrático médio total de predição futura *normalizado* em relação a R(0) e G é fator de ganho da expressão (9). Uma condição necessária e suficiente para que o sistema V(z) resultante seja estável é que | k_i | seja menor que 1,0 [11].

5.2. Métodos Lattice

Teoricamente, o método da autocorrelação resulta sempre em um sistema V(z) estável. No entanto, o método sofre do efeito do cálculo com precisão finita, enquanto a *formulação lattice* possui uma sensibilidade muito menor [19]. A formulação lattice baseia-se no fato dos erros de predição futura e passada de um preditor de ordem P, poderem ser calculados, recursivamente, a partir dos erros de predição futura e passada de preditores de ordem menor [12, 19]:

$$f_0(n) = b_0(n) = y(n) \qquad (12a)$$

$$f_{i+1}(n) = f_i(n) - k_{i+1}\, b_i(n-1) \qquad (12b)$$

$$b_{i+1}(n) = -k_{i+1}\, f_i(n) + b_i(n-1) \qquad (12c)$$

No método lattice a solução é calculada diretamente a partir das amostras do sinal de voz. Não existe a necessidade de janelamento e portanto, $y_w(n) = y(n)$. As derivações possíveis são:

5.2.1. Método "Forward" – Minimização do Erro de Predição Futura

Este método é a implementação lattice do método da autocorrelação e baseia-se, portanto, na minimização de ξ_f^p. Substituindo (12b) em (8a) e derivando em relação aos coeficientes de reflexão k_i, resulta:

$$k_i^f = \frac{\sum_{n=0}^{N-1} f_{i-1}(n)\, b_{i-1}(n-1)}{\sum_{n=0}^{N-1} b_{i-1}^2(n-1)} \qquad (13)$$

5.2.2. Método "Backward" – Minimização do Erro de Predição Passada

Neste caso, minimiza-se ξ_b^p. Substituindo (12c) em (8b) e derivando em relação aos coeficientes de reflexão k_i, resulta:

$$k_i^b = \frac{\sum_{n=0}^{N-1} f_{i-1}(n)\, b_{i-1}(n-1)}{\sum_{n=0}^{N-1} f_{i-1}^2(n-1)} \qquad (14)$$

Observa-se que k_i^f e k_i^b possuem o mesmo sinal.

5.2.3. Método de Itakura

Os coeficientes de reflexão calculados pelas expressões (13) e (14) não garantem, necessariamente, um filtro estável V(z)[19]. Itakura propõe tomar a média geométrica de k_i^f e k_i^b [14, 19]:

$$k_i^I = \text{Sinal}\,(k_i^f)\sqrt{k_i^f\, k_i^b} \qquad (15)$$

5.2.4. Método do Mínimo

Neste caso, o mínimo, em valor absoluto, entre k_i^f e k_i^b é escolhido:

$$k_i^M = \text{Sinal}\ (k_i^f)\ \text{mínimo}\ \{\ |\ k_i^f\ |\ ,\ |\ k_i^b\ |\ \} \tag{16}$$

5.2.5. Método de Burg

A solução proposta por Burg minimiza a *soma* dos erros de predição futura e de predição passada, para cada segmento do sinal de voz e resulta sempre, em sistemas V(z) estáveis [14, 19]:

$$\xi_T^i = \xi_f^i + \xi_b^i \tag{17a}$$

Substituindo-se (8b) e (8c) em (17a) e derivando-se em relação a k_i, resulta [14]:

$$k_i^B = \frac{2\ k_i^f\ k_i^b}{k_i^f + k_i^b} \tag{17b}$$

5.3. Ordem da Análise LPC

Atal e Hanauer [13] e Markel e Gray [14] apresentam uma relação entre a ordem P da análise LPC, a freqüência de amostragem F_s, o comprimento do trato vocálico T_v e a velocidade do som no ar c:

$$F_s = \frac{P\ c}{2\ T_v} \tag{18}$$

O simulador SIR PI permite escolher o valor de P. Na prática são utilizados valores entre 8 e 14 [12, 13, 14].

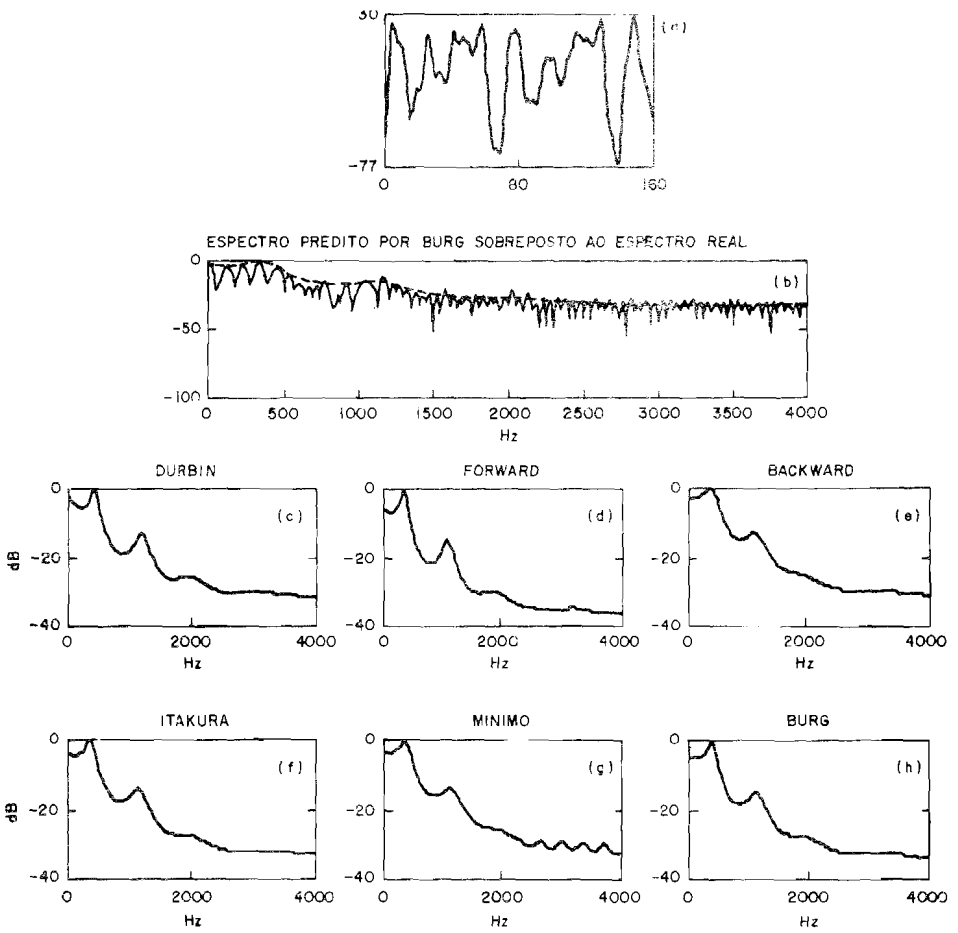

Figura 8. Segmento de 160 amostras (20ms) do som "oi" do dígito "oito" (a), o seu espectro de freqüência (b) e os espectros de V(z) obtidos com os métodos: da autocorrelação (c), "Forward" (d), "Backward" (e), Itakura (f), Mínimo (g) e Burg (h). P = 12.

6. MEDIDAS DE SEMELHANÇA

As medidas de semelhança permitem comparar os parâmetros de dois sinais de voz com o intuito de verificar se podem ser considerados como representantes da mesma palavra. O simulador SIR PI permite escolher entre várias alternativas. Considerando dois segmentos de voz, um de teste, t, e outro de referência, r. tem-se:

6.1. Medida Euclideana

$$d^2 = \sum_{k=1}^{P} (\bar{a}_k^t - \bar{a}_k^r)^2 \qquad (19)$$

6.2. Medida Absoluta

$$d_a = \sum_{k=1}^{P} | a_k^t - a_k^r | \qquad (20)$$

6.3. Medida Cepstral

$$d_c^2 = \sum_{k=1}^{L} (c_k^t - c_k^r)^2 \qquad (21)$$

onde L é o número desejado de coeficientes cepstrais { c_k }, que podem ser obtidos facilmente a partir dos coeficientes LPC [20].

6.4. Medida Cepstral Ponderada

$$d_{cp}^2 = \sum_{k=1}^{L} q(k) (c_k^t - c_k^r)^2 \qquad (22)$$

O simulador SIR PI permite programar a ponderação q(k) e a ordem L.

6.5. Medida da Razão de Verossimilhança

Esta medida, proposta por Itakura [4], é juntamente com as medidas ceptrais, uma das medidas mais bem sucedidas. A sua idéia é simples e baseia-se na comparação das energias dos erros de predição futura $f_p(n)$, nas saídas dos

filtros inversos $A_t(z)$ e $A_r(z)$, de teste e de referência respectivamente, quando o segmento de teste é aplicado a ambos, conforme figura 9:

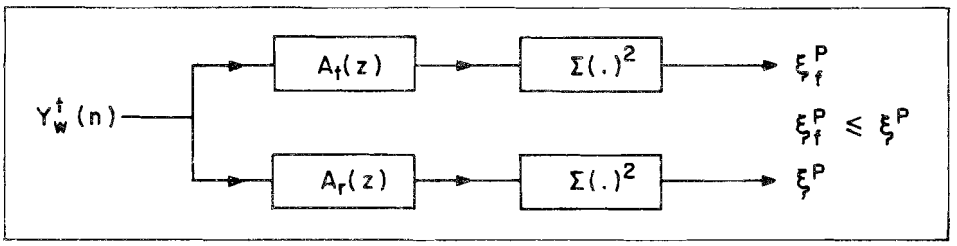

Figura 9. Medida da razão de verossimilhança de Itakura

A medida de distância é dada pela expressão abaixo, onde o índice t indica que os erros correspondem à aplicação do segmento de teste nas entradas dos filtros inversos:

$$d_{rv}^t = 1n\left[\frac{\xi_r^p}{\xi_f^p}\right] \tag{23}$$

Percebe-se que a medida proposta por Itakura não é uma medida simétrica ($d_{rv}^t \neq d_{rv}^r$)

6.6. Medida de Itakura-Saito

A distância de Itakura-Saito é definida pela expressão seguinte:

$$d_{is}^t = \left[\frac{G_t}{G_r}\right]^2 \left[\frac{\xi_r^p}{\xi_f^p}\right]^i - 2\,\iota n\left[\frac{G_t}{G_r}\right] - 1 \tag{24}$$

G_t e G_r são os fatores de ganho dos segmentos de teste e referência, respectivamente (ver expressão (11e)).

Percebe-se, também, que a medida de Itakura-Saito não é simétrica.

6.7. Medida COSH

O objetivo da medida COSH é resolver o problema de assimetria da medida da razão de verossimilhança. A medida COSH é definida por:

$$d_{cosh} = \ln [\ 1 + \Omega + \sqrt{\Omega\,(2 + \Omega)}\ ] \qquad (25)$$

onde Ω é a medida de d_{ts}^{t} e d_{ts}.

6.8. Medida COSH Mínima

Considerando a razão entre os fatores de ganho $(G_t/G_r)^2$ como variável e minimizando Ω em relação a esta variável, resulta [20]:

$$d_{\Omega}\min = \sqrt{\frac{(\xi^p/\xi_f^p)^t}{(\xi^p/\xi_f^p)^r}} - 1 \qquad (26)$$

7. ALINHAMENTO TEMPORAL DINÂMICO

O principal elemento de um sistema reconhecedor de palavras isoladas é o bloco responsável pela comparação da palavra pronunciada com as palavras do dicionário do sistema (palavra de referência). É fato conhecido que o falante não consegue, geralmente, repetir a pronúncia de uma mesma palavra, à mesma taxa e com a mesma duração. Esta variação na pronúncia da palavra provoca flutuações *não lineares* ao longo do eixo do tempo [8, 22]. A solução deste problema sugere realizar o *alinhamento temporal dinâmico* das duas palavras ("Dynamic Time Warping – DIW"). O algoritmo geral de alinhamento temporal dinâmico considera duas palavras, uma de referência e outra de teste e determina a função $\phi(n)$ que mapeia a palavra de referência na palavra de teste ou vice-versa. Deve-se escolher a função que realiza o mapeamento com a *menor distância total* D_T, entre as duas palavras [21, 22, 23]. Em termos gerais, a *distância total* é a soma das *distâncias locais* ao longo da trajetória.

Para tal, a cada ponto (i, j), do plano de mapeamento, atribui-se uma *soma parcial* S_{ij}, que mede a distância acumulada até o ponto (i, j):

$$S_{ij} = d_{ij} + \text{mínimo} \{ S_{pq} \}, \text{ para } p \leq i \text{ e } q \leq j \quad (27a)$$

$$S_{11} = d_{11} \quad (27b)$$

$$D_T = \text{mínimo} \{ S_{pq} \}, \text{ para } p = \text{última coluna} \quad (28)$$

onde d_{ij} representa a distância local entre os segmentos i (eixo horizontal) e j (eixo vertical) do plano (i, j) e S_{pq} representa a distância acumulada nos pontos (p, q) à esquerda e abaixo de (i, j). Para garantir que não ocorram expansões ou compressões excessivas, a inclinação de $\phi(n)$ deve situar-se, preferencialmente, na faixa $[\frac{1}{2}, 2]$ [22]. *Restrições locais* são ainda, impostas para especificar quais são os pontos (p, q) que devem ser considerados na minimização de S_{pq}. Existem vários tipos de restrições locais, admitidas pelo SIR PI, conforme figura 10 [4, 22, 23].

Outra restrição geralmente imposta, é o alinhamento dos segmentos extremos das duas palavras. Supondo, sem perda de generalidade, que os segmentos das palavras dos eixos horizontal e vertical estão indexados de 1 a N_h e de 1 a N_v, respectivamente, então o alinhamento dos extremos implica nas seguintes restrições [21]:

$$1 \leq \Phi(1) \leq \delta + 1 \quad (29a)$$

$$N_v - \delta \leq \Phi(N_h) \leq N_v \quad (29b)$$

As condições (29), em conjunto com a restrição de inclinação da função de mapeamento, restringem a área de procura da trajetória ótima à região hachurada da figura 11. As duas retas paralelas que reduzem a área de procura são uma imposição adicional para diminuir o tempo de processamento necessário.

O fator θ, denominado pelo SIR PI de *fator global*, representa a máxima diferença absoluta permitida entre os segmentos de teste e de referência.

$$|i - j| \leq \theta \quad (30)$$

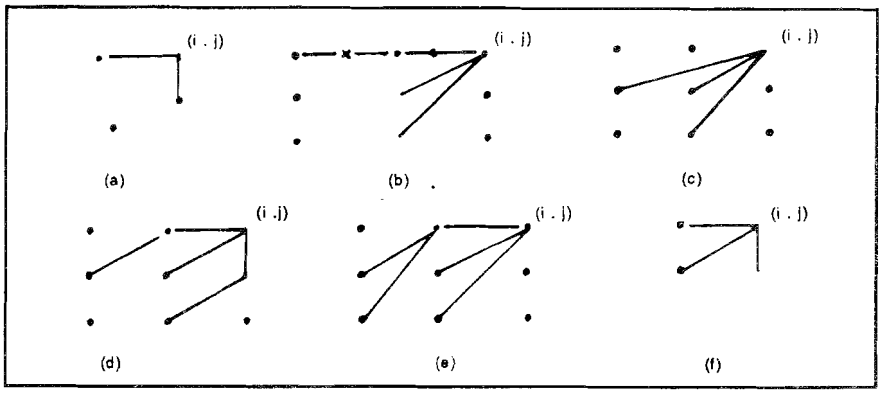

Figura 10. Restrições locais admitidas pelo simulador. Horizontal-Vertical (HV) (a), Itakura (b) 3 Diagonais (c), Horizontal-Vertical-3 Diagonais (HV3D) (d), Horizontal-4 Diagonais (H4D) (e) e Horizontal-Diagonal-Vertical (HV1D) (f).

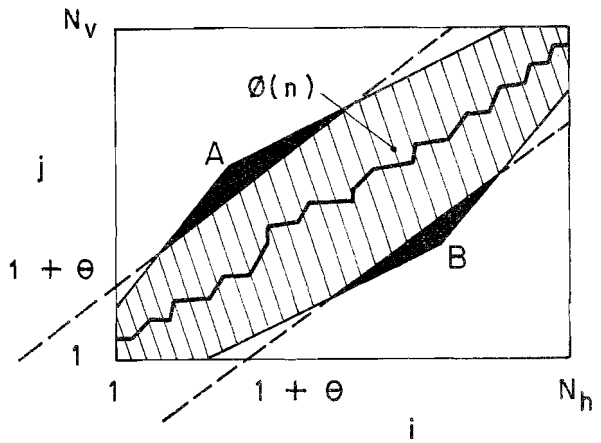

Figura 11. Região de procura da função ϕ (n)

Myers e Rabiner [23] mostram que a redução da área de procura torna-se vantajosa se os comprimentos das duas palavras forem normalizados para um mesmo valor, antes da aplicação do algorítmo de alinhamento temporal dinâmico. Esta normalização pode ser feita por *interpolação* ou por *dizimação*. Outro ponto importante a ser mencionado, refere-se à opção de se alinhar a palavra de teste ao longo do eixo horizontal ou ao longo do eixo

vertical. Sabe-se que a escolha influi no resultado final. Existe, então, a necessidade de se normalizar a distância total com relação ao número de segmentos da palavra que está ao longo do eixo horizontal, para que a distância total represente uma distância média ao longo da trajetória [22, 23]. Esta normalização depende do tipo da função de *ponderação* das condições locais, que se está empregando. Em outras palavras, pode-se atribuir pesos aos ramos de transição das condições locais da **Figura 10**. Existem várias funções de ponderação. O peso de um ramo r, Λ^r, pode ser dado por [22, 23]:

$$\text{Mínimo } (\Delta x, \Delta y) : \Lambda^r = \text{mínimo } (\Delta x^r, \Delta y^r) \tag{31a}$$

$$\text{Máximo } (\Delta x, \Delta y) : \Lambda^r = \text{máximo } (\Delta x^r, \Delta y^r) \tag{31b}$$

$$\Delta x : \Lambda^r = \Delta x^r \tag{31c}$$

$$\Delta x + \Delta y : \Lambda^r = \Delta x^r + \Delta y^r \tag{31d}$$

Os termos Δx^r e Δy^r representam a variação das coordenadas x e y ao longo do ramo r. Costuma-se adotar os seguintes valores para o fator de normalização, NR, em função da função de ponderação:

$$\text{Mínimo } (\Delta x, \Delta y) : NR = N_q \tag{32a}$$

$$\text{Máximo } (\Delta x, \Delta y) : NR = N_q \tag{32b}$$

$$\Delta x : NR = N_q \tag{32c}$$

$$\Delta x + \Delta y : NR = N_q + N_v \tag{32d}$$

onde N_q é o ponto onde a função de mapeamento atinge o valor máximo N_v ($N_q \leq N_h$)[21], isto é:

$$\phi (N_q) = N_v \tag{33}$$

Com isso, a expressão geral da distância total torna-se:

$$D_T = \frac{\Sigma\, d_{ij}\, \Lambda^r}{NR\,(\Lambda)}, \text{ para } (i, j) \text{ e } \phi(n) \tag{34}$$

O simulador SIR PI permite controlar o bloco de alinhamento temporal dinâmico e programar todos os parâmetros de interesse.

8. REGRAS DE DECISÃO

O último passo no sistema de reconhecimento de palavras isoladas é o processo de decisão. Este passo encarrega-se de analisar as distâncias totais, calculadas pelo bloco de DTW, entre a palavra de teste e as palavras de referência, e decidir qual a palavra do dicionário que mais se aproxima do comando falado. As regras de decisão mais populares são a regra NN ("Nearest Neighbour") e a regra KNN ("K Nearest Neighbour") [1, 8, 24]. As duas regras estão implementadas no SIR PI.

9. VISÃO GLOBAL DO RECONHECEDOR

A **Figura 12** apresenta o diagrama em blocos do sistema reconhecedor de palavras isoladas. Todos os blocos foram discutidos nos itens anteriores.

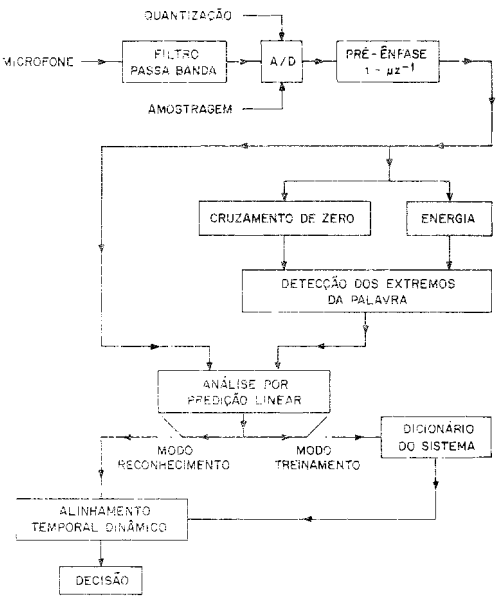

Figura 12. Diagrama em blocos do reconhecedor

Deve-se mencionar, neste ponto, que o reconhecedor possui dois modos de operação: o modo de *treinamento* e o modo de *reconhecimento*. No modo de treinamento, as palavras de referência são pronunciadas e analisadas e os seus coeficientes LPC são gerados e armazenados. No modo de reconhecimento, os coeficientes LPC da palavra pronunciada são gerados e o processo de análise continua até o passo de decisão.

10. TESTE DO SIMULADOR

Neste item, são apresentados os resultados obtidos durante a simulação de *uma configuração* freqüente na literatura. Os parâmetros da configuração testada estão listados a seguir:

LPC: Método de Durbin, janela de Hamming com N = 160, D = 80 e p = 8

DTW: Distância cepstral com L = 8, restrição de Itakura com ponderação Δx, regra NN ou KNN de decisão, palavra de teste ao longo do eixo horizontal, limitação global desativada, pontos extremos vinculados (δ = 0), e normalização de comprimentos ativada.

"Energia": Medida absoluta, janela retangular com N = 80 e D = 80, e fator de pré-ênfase μ = 0,95.

O sinal de voz é filtrado entre 300 e 3400 Hz, amostrado a 8000 Hz e quantizado em 12 bits. As palavras de teste e de referência foram levantadas com o auxílio de 8 falantes designados A, B, C, X1, X2, X3, X4 e X5. Os falantes A, B, X1, X2 e X3 são masculinos, enquanto os falantes C, X4 e X5 são femininos. As palavras utilizadas nos testes são os dígitos de ZERO a NOVE e as palavras MEIA, AJUDA, CANCELE e TERMINE. Cada falante do grupo { A, B, C } pronunciou cada palavra 6 vezes. Portanto, a cada falante A, B ou C correspondem 6 conjuntos de 14 pronúncias, designadas A_i, B_i, e C_i, i = 1, 2, ..., 6.

Cada falante do grupo { X1, X2, X3, X4, X5 } pronunciou cada palavra uma vez. Os testes foram realizados em um microcomputador compatível com o PC da IBM[1] (8 MHz, microprocessador 286 da Intel[2] e com coprocessador). O tempo

1. *IBM* é marca registrada da International Business Machines Corporation.

2. Intel é marca registrada da Intel Corporation.

médio gasto na análise de uma palavra de teste (pré-ênfase, detecção dos pontos extremos e cálculo dos parâmetros LPC) foi de 20 segundos. O tempo médio gasto para alinhar dinamicamente duas palavras foi de 22 segundos. Em média, foram necessários 6 Kbytes de memória, em disco, por palavra de referência. A seguir são apresentados os resultados obtidos:

Dicionários I.A, I.B e I.C:

Os conjuntos A_6, B_6 e C_6 foram escolhidos como referência para gerar três dicionários I.A, I.B e I.C, respectivamente. Os 5 conjuntos restantes, de cada falante, foram utilizados para testar o dicionário correspondente. Os resultados estão listados na tabela 1:

Dicionário	Número de palavras não reconhecidas	Taxa de acerto
I. A	0	100%
I. B	4	94%
I.C	3	95%

Tabela 1. Testes dos dicionários I

Portanto a taxa média de acerto é da ordem de 96%.

Dicionário II:

Os conjuntos A_6, B_6 e C_6 foram escolhidos como referência para gerar o dicionário II. Os 5 conjuntos restantes, de cada falante, foram utilizados para testar o dicionário. (210 palavras de teste). Apenas quatro palavras de teste não foram reconhecidas corretamente, o que indica uma taxa de acerto de 98%. Percebe-se que a taxa de acerto aumentou de 96% para 98% devido ao número maior de referências por comando.

Dicionário III:

Os conjuntos A_6, B_6, C_6, A_4, B_4 e C_4 foram escolhidos como referência para gerar o dicionário III. As pronúncias dos falantes X1 a X5 foram utilizadas nos

testes do dicionário (70 palavras de teste, no total). Foram realizados dois testes: um teste com a regra NN e outro teste com a regra KNN (K=3). Para ambos os casos apenas três palavras de teste não foram reconhecidas corretamente, o que indica uma taxa de acerto da ordem de 95%.

Os testes realizados mostram que a configuração testada fornece bons índices de reconhecimento e confirmam os resultados obtidos por vários pesquisadores [4, 5, 6, 7, 9, 10].

Convém ressaltar que os resultados apresentados neste item referem-se ao teste de *uma* configuração. Outras configurações podem também ser programadas e testadas.

11. INTERFACE DO SIMULADOR COM O USUÁRIO

A operação do simulador é bastante simples e a sua programação é totalmente controlada por janelas ou menus. O simulador [1] foi desenvolvido em um microcomputador compatível com o PC da IBM[1] e escrito em linguagem C (Microsoft[2] versão 5.10). O simulador possui aproximadamente 10.000 linhas de código.

12. CONCLUSÕES E COMENTÁRIOS

Neste trabalho foi discutido o desenvolvimento de um simulador de reconhecedores de comandos da fala. Os parâmetros que controlam a operação do sistema são facilmente programados pelo usuário e a configuração escolhida pode ser facilmente simulada e avaliada. A nova versão do simulador se encontra em desenvolvimento e deverá incluir a geração de dicionários independentes de falantes [6, 8, 25, 26, 27].

O simulador desenvolvido [1] permite testar muitos dos algoritmos apresentados na literatura e representa, portanto, uma contribuição na área de reconhecimento de voz.

1. IBM é marca registrada da International Business Machines
2. Microsoft é marca registrada da Microsoft Corporation

AGRADECIMENTOS

Ao Prof. Dr. Zsolt L. Kovacs pelo incentivo e discussões, ao Prof. Edgard José Casaes pela colaboração na área de linguística, ao Eng. Ivandro Sanches pela troca de idéias e aos engenheirandos Wander O. Cesário, Alex D. Zyrianoff e Roberto Martinelli pela colaboração.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] A.H. Sayed, "Simulador de Reconhecedores de Palavras Isoladas – Sir PI," *Dissertação de Mestrado. Departamento de Engenharia de Eletricidade da EPUSP*, Julho 1989.

[2] D.R. Reddy, "Speech recognition by Machine: A Review" *Proceedings of the IEEE*, vol. 64, pp. 501-531, April 1976.

[3] G.M.White, "Speech Recognition: A Tutorial Overview", *Computer*, vol. 9, pp. 40-53, May 1976.

[4] F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-23, n 1, pp. 67-72, February 1975.

[5] M.R. Sambur and L.R. Rabiner, "A Speaker Independent Digit Recognition System", *The Bell System Technical Journal*, vol. 54, n 1, pp. 91-103, January 1976.

[6] L.R. Rabiner, "On creating Reference Templates for Speaker Independent Recognition of Isolated Words", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-26, n 1, pp. 34-41, February 1978.

[7] L.R. Rabiner, J.G. Wilpon and A.E. Rosenberg, "A Voice Controlled Repertory Dialer System", *The Bell System Technical Journal*, vol. 59, pp. 1153-1163, September 1980.

[8] L.R. Rabiner and S.E. Levinson, "Isolated and Connected Word Recognition – Theory and Selected Applications", *IEEE Transactions on Communications*, vol. COM-29, n 8, pp. 621-658, May 1981.

[9] Y.Tohkura, "A Weighted Cepstral Distance Measure for Speech Recognition", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-35, n 10, pp.1414-1422, October 1987.

[10] J. G. Wilpon, L.R. Rabiner and A. Bergh, "Speaker Independent Isolated Word Recognition Using a 129-Word Airline Vocabulary", *The Jorunal of the Acoustical Society of America*, vol. 72, n 2, pp. 390-396, August 1982.

[11] J.L. Flanagan, Speech Analysis, Synthesis and Perception, *Springer-Verlag*, New York, 1972.

[12] L.R. Rabiner and R.W. Schafer, Digital Processing of Speech Signals, *Prentice Hall*, New Jersey, 1978.

[13] B.S. Atal and S.L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave", *The Journal of the Acoustical Society of America*, vol. 50, n 2, part 2, pp. 637-655, April 1971.

[14] J.D.Markel and A.H. Gray, Jr. Linear Prediction of Speech, *Springer Verlag*, New York, 1976.

[15] R.W. Schafer and L.R. Rabiner, "Digital Representations of Speech Signals", *Proceedings of the IEEE*, vol. 63, pp. 662-677, April 1975.

[16] A.V. Oppenheim and R.W. Schafer, Digital Signal Processing, *Prontice-Hall*, New Jersey, 1975.

[17] A.H. Sayed A. Ferrara e E.T. Taniguchi, "Sistema Reconhecedor de Voz Humana – Programável e com Capacidade de Processamento", Projeto de Formatura, *Departamento de Engenharia de Eletricidade da EPUSP*, 1987.

[18] L.R. Rabiner and M. R. Sambur, "An Algorithm for Determining the Endpoints of Isolated Utterances", *The Bell System Technical Journal*, vol. 54, n 2, pp. 297-315, February 1975.

[19] J. Makhoul, " Stable and Efficient Lattice Methods for Linear Prediction", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-25, n 5, pp. 423-428, October 1977.

[20] A.H. Gray, Jr. and J.D. Markel, "Distance Measures for Speech Processing", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-24, n 5, pp. 380-391, October 1976.

[21] L. Rabiner, A.E. Rosenberg and S.E. Levinson, "Considerations in Dynamic Time Algorithms for Discrete Word Recognition", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-26, n6, pp. 676-592, December 1978.

[22] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-26, pp. 43-49, February 1978.

[23] C. Myers, L.R. Rabiner and A.E. Rosenberg, "Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-29, n 6, pp. 623-635, December 1980.

[24] V.N. Gupta, J.K. Bryan and J.N. Gowdy, "A Speaker-Independent Speech Recognition System Based on Linear Prediction". *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-26, n 1, pp. 27-33, February 1978.

[25] B.H. Juang, D.Y. Wong and A.H. Gray, Jr., "Distortion Performance of Vector Quantization for LPC Voice Coding", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-30, n 2, pp. 294-303, April 1982.

[26] Y. Linde, A. Buzo and R. M. Gray, "An Algorithm for Vector Quantizer Design", *IEEE Transactions on Communications*, vol. COM-28, n 1, pp. 84-95, January 1980.

[27] A. Buzo, A.H. Gray, Jr., R.M. Gray and J.D. Markel, "Speech Coding Based Upon Vector Quantization", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-28, n 5, pp. 562-573, October 1980.