Sistema para Controle de Elevadores por Voz

Adriano Petry[1]
Dante Augusto Couto Barone[2]

Universidade Federal do Rio Grande do Sul - Instituto de Informática
Av. Bento Gonçalves, 9500 bloco IV - Porto Alegre - RS – Brasil

Palavras chave: Interfaces homem – máquina, Informática em Engenharia

Resumo

O sistema desenvolvido e apresentado neste trabalho combina o reconhecimento de comandos discretos com a verificação da identidade do locutor, para o controle de elevadores. Através da identificação do andar desejado, associado à verificação se o locutor possui acesso válido àquele andar, o sistema comanda um elevador. São mostrados os resultados obtidos com a técnica Quantização Vetorial Multisecção (MSVQ), utilizando coeficientes mel-cepstrais, no reconhecimento de comandos e na verificação da identidade de um locutor. O sistema foi implementado em um elevador na empresa Elevadores Sûr S.A., e apresentou taxas para o reconhecimento de comandos superior a 97% e rejeição de impostores superior a 98%, em 4500 tentativas de acesso por intrusos.

1 Introdução

Nos últimos anos, muita pesquisa tem sido feita no sentido de melhorar as técnicas para o reconhecimento automático de voz (RAV) existentes. Tem-se estudado técnicas de RAV desde os anos 50, quando pesquisadores dos laboratórios Bell desenvolveram um sistema que reconhecia dígitos isolados falados por apenas um locutor. Foram feitas pesquisas semelhantes em outros locais, como no Instituto de Tecnologia de Massachusets (MIT) e na universidade de Kioto. Na década de 70 iniciou-se a utilização de técnicas de programação dinâmica para realizar o reconhecimento de voz. Na década de 80 o foco foi o reconhecimento de frases e fala contínua, ao invés de palavras isoladas. Iniciou-se então a utilização de modelos estatísticos (Modelos Ocultos de Markov - HMM) e Redes Neurais Artificiais (RNA). Atualmente procura-se a robustez do sistema a ruídos, variabilidade de pronúncia e diferença de vozes entre locutores, a fim de atingir taxas de reconhecimento mais elevadas.

O controle de equipamentos pela voz oferece muitas vantagens aos usuários. Ao utilizar uma interface homem-máquina por voz, possibilita-se utilizar as mãos em outras tarefas simultaneamente e evita-se o contato físico direto entre o usuário e o equipamento que controlado. Além disso, não há necessidade de treinamento prévio por parte do usuário. Podemos considerar a comunicação vocal uma das mais utilizadas para troca de informações entre pessoas, consequentemente uma interface muito mais natural e amigável, quando comparada a apertos de botões ou manipulação de alavancas.

Uma aplicação para a utilização da interface por voz é o controle de elevadores. Isso acarretaria maior comodidade ao usuário e facilidade de operação. Em elevadores residenciais de acesso restrito onde apenas os moradores devem poder entrar, senhas e cartões magnéticos também poderiam ser substituídos por simples comandos de voz.

O sistema apresentado neste trabalho objetiva controlar um elevador através de comandos discretos. Deseja-se não apenas identificar o andar que o usuário deseja ir, mas também verificar a sua identidade. Se o usuário tiver sido previamente cadastrado, o elevador deverá conduzi-lo corretamente ao andar requerido. Caso contrário, o usuário deve ser informado que seu acesso a tal andar foi negado. O protótipo construído é composto por um computador pessoal conectado ao hardware de controle de um elevador através da porta serial do computador. A entrada de voz é feita através de um microfone unidirecional, conectado a um filtro passa-baixas e desse para uma placa de som conectada ao computador pessoal. Os algoritmos implementados rodam no computador pessoal que informa ao elevador como proceder.

[1] Engenheiro eletricista, Doutorando no Instituto de Informática / UFRGS.
[2] Professor Doutor no Instituto de Informática / UFRGS.

2 Funcionalidade do Sistema

No sistema de controle de elevadores por voz, os usuários cadastrados devem, *a priori*, fornecer amostras de voz em dias e horários variados para o treinamento do sistema.

Para o reconhecimento, o usuário, ao entrar no elevador, deverá falar o andar desejado, como por exemplo "quinto andar". O sistema faz a detecção automática dos limites das palavras avaliando a energia normalizada dos *frames*, separando a palavra correspondente ao andar desejado (palavra "quinto") e a palavra "andar", comum a todos os andares. Após, o sistema identifica o andar desejado analisando a primeira palavra. São selecionadas então as informações dos usuários que têm direito de acesso ao andar desejado. Com a palavra "andar", o sistema valida (ou não) a identidade do locutor como um entre os locutores cadastrados naquele andar e conduz o elevador apropriadamente.

3 Técnicas Utilizadas

O sistema para reconhecimento de comandos e verificação de locutor faz uso de técnicas e algoritmos distintos. Para a identificação das palavras ditas, são utilizadas técnicas de *aquisição do sinal vocal e detecção dos limites das palavras*. Não são utilizadas as amostras de voz para a realização da classificação, mas coeficientes extraídos através de técnicas descritas na etapa de *pré-processamento e extração de características*. De posse desses coeficientes, procedemos a *classificação* do sinal. A etapa de *classificação* poderá também ser separada nas etapas de *geração de codebook*, *quantização* e *comparação*.

3.1 Aquisição do Sinal Vocal e Detecção dos Limites das Palavras

A aquisição do sinal de voz é realizada utilizando-se um microfone conectado a um filtro passa baixas, cuja saída está ligada a uma placa de som instalada no computador pessoal. A freqüência de corte do filtro é de 4,5KHz. A placa de som transforma o sinal analógico filtrado em amostras digitais, a uma taxa de 11025Hz, com resolução de 16 bits por amostra. Tais amostras são processadas por um algoritmo de detecção de limites de palavras [4], que identifica quando uma palavra iniciou e terminou, gravando os dados da palavra em um arquivo.

O parâmetro de medida utilizado pelo algoritmo de detecção de limites de palavras é a energia média contida em um bloco. Um bloco é um conjunto composto por um número fixo de amostras de voz. O cálculo da energia média pode ser visto na equação 1. O início de uma possível palavra é considerado a partir do primeiro bloco onde a energia ultrapassa um limiar pré-determinado. Para que os blocos seguintes constituam realmente um início de palavra, a energia deles deve permanecer acima do limiar durante um período de tempo pré-estabelecido chamado tempo de início. Caso a energia de um bloco caia abaixo do limiar antes do fim deste período, as amostras até então armazenadas são descartadas e a procura pelo início de palavra recomeça. Caso a energia média dos blocos se mantenha superior ao limiar por todo o tempo de início, começamos a procura pelo fim da palavra. O método de detecção do fim de palavra é idêntico ao método de detecção de início, descrito anteriormente. A diferença está na procura de blocos com energia inferior à estabelecida no limiar. Da mesma forma, os blocos devem permanecer com energia inferior ao limiar por um certo período de tempo para que se considere atingido o fim de palavra. Tal período de tempo é chamado tempo de fim. Os blocos que compõem o tempo de fim não são considerados como componentes da palavra. O algoritmo de detecção de limites de palavra é mostrado na figura 1.

$$En = \frac{1}{N}\sum_{k=0}^{N-1}|x(k)| \qquad (1)$$

onde *En* é a energia média de um bloco composto por *N* amostras de voz, e *x(k)* é a amplitude da k-ésima amostra de voz de um bloco.

O sistema faz a detecção dos limites das palavras de forma automática, evitando com isso a necessidade de apertos de botões. Assim, utilizou-se o algoritmo descrito para realizar tal tarefa, que permite ao usuário, a qualquer momento, iniciar uma locução sem a necessidade de "avisar" o sistema.

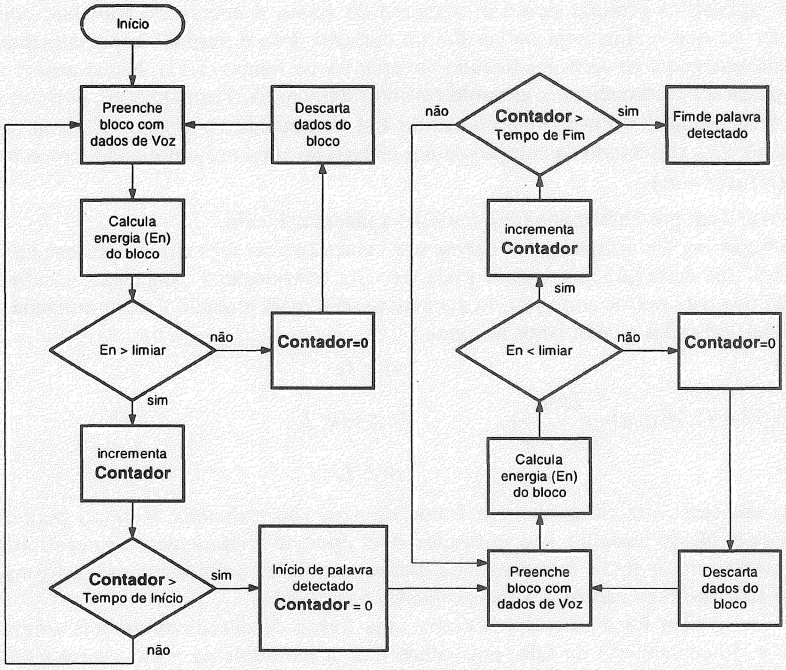

FIGURA 1: Algoritmo para detecção de limites de palavra.

Uma vez detectado o início e o fim da primeira palavra, o algoritmo aguarda um período de tempo pré-determinado esperando pela detecção de início da segunda palavra. Caso não haja a detecção de início da segunda palavra dentro desse período de tempo, a gravação efetuada é ignorada, e isso é informado ao usuário do sistema. Essa proteção adicional garante que as duas palavras sejam captadas corretamente. No exemplo de "quinto andar", as palavras "quinto" e "andar" seriam captadas distintamente. Se o usuário falasse tais palavras sem uma pausa mínima entre elas, o sistema erroneamente captaria a primeira palavra como "quintoandar" (sem pausa entre palavras), mas descartaria tal gravação pois não detectaria o início da segunda palavra. O período de tempo máximo entre as duas palavras foi estabelecido em 3s ou 2067 blocos. Após a realização do reconhecimento do andar que se deseja ir utilizando a primeira palavra, é efetuada a verificação da identidade do locutor, utilizando a segunda palavra.

Foram utilizados blocos de 32 bytes ou 16 amostras de voz, tempo de início de 70ms ou 48 blocos, tempo de fim igual a 150ms ou 103 blocos.

3.2 Pré-processamento e Extração de Características

Após a aquisição do sinal de voz, é realizado um pré-processamento nas amostras a fim de prepará-las para a extração de suas características [3][5][6]. Tais características são utilizadas no algoritmo de reconhecimento de padrões. O pré-processamento é composto pelas etapas de pré-ênfase, divisão do sinal em *frames* e janelamento. A extração de característica fornece um conjunto de coeficientes representantes de cada *frame* do sinal.

A pré-ênfase objetiva eliminar uma tendência espectral de aproximadamente -6dB/oitava na fala irradiada dos lábios. Essa distorção espectral não traz informação adicional e pode ser eliminada através da aplicação de um filtro, de resposta aproximadamente +6dB/oitava, que ocasionaria um nivelamento no espectro. Para um sistema digital, tais pré-ênfases podem ser implementadas como um circuito analógico, o qual precede o filtro e o amostrador, ou como uma operação digital no sinal amostrado, através de um filtro FIR de primeira ordem. O efeito de ascensão de +6dB/oitava pode ser obtido pela diferenciação da entrada. A equação 2 descreve o pré-enfatizamento realizado no sinal amostrado.

$$y(n) = x(n) - a.x(n-1) \qquad \text{para } 1 \le n < M \qquad (2)$$

onde M é o número de amostras do sinal amostrado $x(n)$, $y(n)$ é o sinal pré-enfatizado o parâmetro constante "a" é usualmente escolhido entre 0,9 e 1. Foi utilizado "a" igual a 0,95.

Em todas as aplicações práticas de processamento de sinais, é necessário trabalhar com "pequenas porções" ou *frames* do sinal, a não ser que o sinal seja de curtíssima duração. Isso é verdade especialmente se estivermos utilizando técnicas de análise convencionais de sistemas lineares invariantes no tempo (LTI). Nesse caso é necessário selecionar uma porção de sinal que possa ser razoavelmente assumida como estacionária. Formalmente, definimos um *frame* de voz como sendo o produto de uma janela discreta *w(n)* de tamanho *L* e terminando no tempo "*l*", com relação à seqüência de voz discreta (pré-enfatizada) *y(n)*, resultando na seleção de um pedaço do sinal pré-enfatizado, como mostra a equação 3.

$$f(n) = y(n).w(l-n) \tag{3}$$

onde *f(n)* é um *frame* do sinal pré-enfatizado *y(n)*, e *w(n)* é a janela aplicada.

A janela de hamming foi utilizada, por apresentar características espectrais interessantes e por atenuar a transição entre *frames* adjacentes. Sua descrição matemática pode ser vista na equação 4. As janelas usualmente são sobrepostas entre si, para que a variação dos parâmetros entre janelas sucessivas seja mais gradual. Foram utilizadas janelas de tamanho igual a 330 amostras ou 30ms, aplicadas a cada 10ms de sinal.

$$w(n) = \begin{cases} 0 & n < 0 \\ 0{,}54 - 0{,}46\cos\left(\dfrac{2\pi n}{330-1}\right) & 0 \le n < L \\ 0 & n \ge L \end{cases} \tag{4}$$

A partir das amostras que compõem um frame de voz, são utilizadas técnicas para obtenção dos coeficientes representantes do mesmo. Nesse trabalho, são utilizados dois tipos de coeficientes: os cepstrais e os mel-cepstrais. Esses coeficientes proporcionam uma redução no volume de dados, sem perda significativa de informação útil. Essa redução de dimensão nos fornece sistemas reconhecedores mais robustos e eficientes.

A análise homomórfica foi desenvolvida como uma forma de desconvoluir dois sinais. Análise homomórfica é considerada útil para o processamento da fala, pois oferece uma metodologia para a separação do sinal de excitação da resposta impulsiva do trato vocal. No modelamento matemático para a produção do sinal vocal [1][2], temos que o um *frame f(n)* do sinal vocal (pré-enfatizado) *y(n)* pode ser escrito como o produto da convolução do sinal de excitação *u(n)* com a resposta impulsiva do trato vocal *h(n)*, como é visto na equação 5.

$$f(n) = u(n) \otimes h(n) \tag{5}$$

A representação no domínio freqüência desse processo através da aplicação da transformada de Fourier, transforma a operação de convolução em multiplicação. Aplicando-se a função logarítmica, transformamos a multiplicação na soma (ou sobreposição) de sinais, como mostra a equação 6.

$$\log\left(F\{f(n)\}\right) = \log\left(F\{u(n)\}\right) + \log\left(F\{h(n)\}\right) \tag{6}$$

onde $F\{\bullet\}$ representa a aplicação da transformada discreta de Fourier (DFT).

Aplicando-se a transformada inversa nesse sinal tem-se o *cepstrum* ou coeficientes cepstrais do sinal de voz. Sabe-se que a parcela do sinal de excitação varia mais rapidamente que a resposta impulsiva do trato vocal, então os dois sinais poderiam ser separados no domínio cepstral. Na prática, são utilizados apenas os primeiros coeficientes componentes do cepstrum. Tais coeficientes contém a informação relativa ao trato vocal, que está intimamente relacionada tanto com o comando falado, quanto com o locutor.

Atualmente, muitos sistemas utilizam os coeficientes mel-cepstrais (*mel frequency cepstral coefficients* – MFCC) para o reconhecimento de voz [1][6]. A análise mel-cepstral vem progressivamente substituindo a forma tradicional de utilização de parâmetros cepstais previamente descritos. A diferença entre o cálculo dos coeficientes cepstrais e os coeficientes mel-cepstrais está na aplicação de um banco de filtros digitais ao espectro real do sinal, antes da aplicação da função logarítmica. Tais filtros não estão linearmente espaçados no domínio freqüência. O objetivo de tais filtros é uma tentativa de aproximar a resposta humana a sinais sonoros. Mel é a unidade de medida de freqüências ou picos percebidos de um tom. Essa unidade não corresponde linearmente à freqüência física bem como, aparentemente, o ouvido humano também não o faz. Experiências conduzidas por Stevens e Volkman serviram para traçar uma comparação entre a freqüência real (medida em Hz) e a freqüência percebida (medida em mels). Logo, o espaçamento dos filtros digitais deve respeitar a escala de freqüências percebidas (escala *Mel*).

Podemos definir uma função para mapeamento da freqüência acústica *f* (em Hz) para uma escala de freqüências percebidas *Mel* (em mels) como mostra a equação 7.

$$Mel = 2595.\log_{10}\left(1 + \frac{f}{700}\right) \tag{7}$$

Uma forma de aplicar os filtros digitais de forma espaçada segundo a escala *Mel* seria primeiramente mapear as freqüências acústicas (em Hz) para a escala de freqüências percebidas (em mels), e após aplicar um banco de filtros espaçados linearmente nesse domínio (domínio mel). Isso corresponderia à aplicação de filtros digitais espaçados segundo a escala mel, no domínio das freqüências reais.

O processo de obtenção dos MFCC, *frame* a *frame*, é ilustrado na figura 2. Usualmente é utilizada a transformada inversa do coseno (ICT) para obtenção dos MFCC ao invés da transformada inversa discreta de Fourier. A utilização dessa transformada inversa reforça a informação útil nos MFCC iniciais.

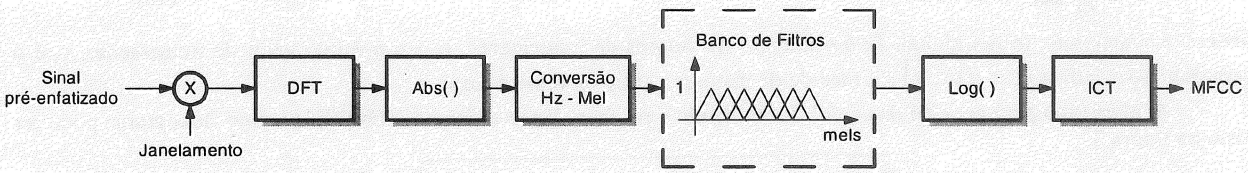

FIGURA 2: Processo de obtenção dos MFCC

O processo de obtenção dos MFCC é matematicamente descrito na equação 8.

$$c(n) = \sum_{k=1}^{K} \log|S(k)| . \cos\left[n\left(k - \frac{1}{2} \right)\frac{\pi}{K} \right] \qquad \text{para } 0 \le n < P \tag{8}$$

onde *c(n)* é o n-ésimo coeficiente mel-cepstral, *P* é o número de coeficientes mel-cepstrais extraídos, *K* é o número de filtros digitais utilizados e *S(k)* é o sinal de saída do banco de filtros digitais.

A aplicação de uma janela de ponderação aos coeficientes mel-cepstrais obtidos, chamada janela de janela de *liftro*, é amplamente empregada em reconhecimento de voz, apresentando bons resultados. Seu objetivo é enfatizar componentes com maior informação espectral útil. Tal ponderação *l(n)* é dada pela equação 9.

$$l(n) = 1 + \frac{Q}{2}\sin\left(\frac{n\pi}{Q} \right) \qquad \text{para } 0 \le n < P \tag{9}$$

onde Q é uma constante chamada de coeficiente de liftro (usualmente igual a 22) e P é o número de coeficientes previamente extraídos.

3.3 Classificação

A partir dos coeficientes extraídos, procedemos ao reconhecimento de padrões. Um padrão é o conjunto de vetores oriundos de uma locução. Denomina-se vetor o conjunto de coeficientes extraídos de um *frame* de voz. Existem três etapas distintas no processo de classificação de padrões através da quantização vetorial[7][8][9][10][11], que são a *geração de codebook*, a *quantização* de um padrão desconhecido, e a *comparação* ou medida de distorção. Adicionalmente, é mostrado como a técnica de quantização vetorial pode ser melhorada utilizando-se um artifício chamado *multisecção*.

Tendo-se um universo de vetores p-dimensionais, são estabelecidos vetores (também p-dimensionais) representantes desse universo. Esses vetores representantes são chamados centróides. O conjunto de todos os centróides é chamado de *codebook*. Os centróides são em número muito menor que o número de vetores que compõem o universo. Assim, podemos discretizar qualquer vetor p-dimensional em um entre os centróides previamente treinados. Quantização é a conversão de um vetor de entrada em um código relacionado a vetores de mesma dimensão previamente treinados (centróides).

Geração de *Codebook*

O objetivo da etapa de geração de *codebook* é estabelecer referências para posterior classificação. Nessa etapa, é criado um *codebook* por comando candidato para representação da palavra desconhecida, no caso de reconhecimento de comandos. Já para a verificação de locutor, é criado um *codebook* por locutor cadastrado no sistema. Nesse caso, é estabelecido também um limiar associado a cada *codebook*.

A etapa de geração de *codebook* consiste na geração dos níveis discretos que cada vetor poderá assumir – os centróides. Esses níveis são armazenados em um *codebook*. Para a geração de tal *codebook*, é utilizado um grupo de vetores de treinamento. Os centróides encontrados nessa etapa devem ser os melhores representantes dos vetores de treinamento.

Em outras palavras, os centróides devem ser tais que minimizem o somatório da distorção de cada vetor de treinamento relacionado com seu respectivo centróide. Um determinado *codebook* é chamado ótimo (ou globalmente ótimo) se, para um número T de centróides, a distorção média de todos os vetores de treinamento (distorção média global), quando comparados com o centróide associado, é menor que a distorção média produzida por qualquer outro *codebook* de T centróides. A associação entre um vetor qualquer e o centróide que melhor o representa é feita utilizando a regra de seleção pela mínima distorção (*nearest neighbor*). O cálculo da distorção média global (DMG) é realizado de acordo com a equação 10.

$$D = \frac{1}{Tr}\sum_{n=1}^{Tr} d[x_n, y_n] \qquad (10)$$

onde D é a distorção média global, Tr é o número de vetores de treinamento, x_n é o n-ésimo vetor de treinamento, y_n é o centróide associado a x_n, e $d[x, y]$ é a medida de distorção entre os vetores x e y.

O algoritmo para geração de centróides utilizado é o Linde, Buzo e Gray (LBG) [7][11], cujo fluxograma pode ser visto na figura 3.

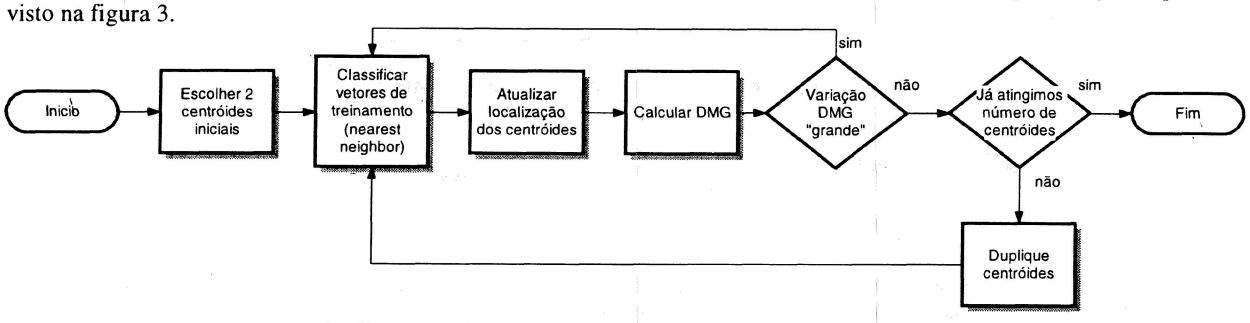

FIGURA 3: Fluxograma do algoritmo LBG de treinamento.

A escolha dos 2 centróides iniciais pode ser realizada de forma aleatória, sem que haja prejuízo à performance do sistema. A classificação dos vetores de treinamento se dá através da técnica de *nearest neighbor*. A atualização dos centróides é feita simplesmente calculando-se o centro geométrico dos vetores que possuem o mesmo centróide associado. Esse centróide é atualizado assumindo as coordenadas desse centro geométrico. O cálculo da distorção média global é realizado de acordo com a equação 10, sabendo-se previamente a qual centróide está associado cada vetor de treinamento. Para identificarmos o quão "grande" foi a variação da DMG, calculamos o valor da distorção diferencial, dado pela equação 11. Se o valor da distorção diferencial for superior a um limiar (normalmente 10^{-2}), consideramos uma variação "grande". Caso contrário, consideramos que os centróides estão "estáveis" o suficiente para serem duplicados ou, caso já forem em número suficiente, para serem armazenados.

$$Dif = \left| \frac{DMG_{atual} - DMG_{anterior}}{DMG_{anterior}} \right| \qquad (11)$$

onde Dif é a distorção diferencial.

A duplicação dos centróides consiste em adicionar um fator de perturbação (normalmente 10^{-2}) positivo e um negativo em cada um dos centróides. Os centróides duplicados iniciais são exatamente iguais aos centróides que os deu origem, a não ser pelo fator de perturbação.

Quantização

A quantização de um padrão é a escolha do centróide que melhor representa cada vetor. Isso é feito levando-se em conta a distância entre o vetor em questão e todos os centróides existentes no *codebook* que está sendo utilizado.

O método mais simples para realizar uma quantização de uma seqüência de vetores, chamado *full search*, seria o de comparar cada vetor da seqüência com todos os centróides armazenados no *codebook*. O centróide mais "semelhante" é assumido como representante do vetor em questão. O processamento necessário para executarmos uma quantização através desse método, porém, é enorme. Um outro método, chamado *tree search*, armazena todos os centróides que foram sendo duplicados e considerados "estáveis" ou prontos para próxima duplicação, e a quantização é realizada através da comparação entre o vetor em questão e os dois centróides que as comparações anteriores indicaram. A figura 4 ilustra o método *tree search*, onde o vetor a ser quantizado é primeiramente comparado com os dois centróides iniciais. Decidido qual dos dois centróides é mais "semelhante", o vetor é comparado então com os próximos dois centróides que foram

derivados do primeiro centróide "vencedor", e assim por diante. O centróide associado ao vetor em questão será escolhido na última comparação.

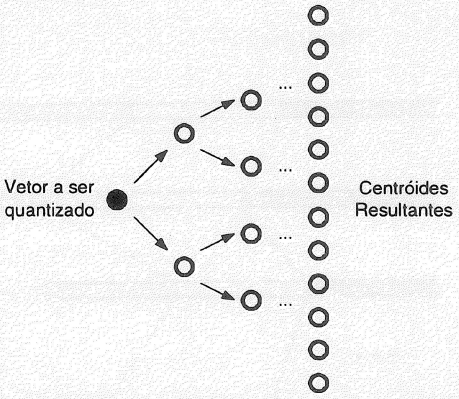

FIGURA 4: Método *tree search* para quantização de um vetor.

Comparação

Como última etapa do processo de classificação de padrões temos a comparação entre dois vetores. A comparação é realizada através do cálculo da distorção entre eles. Há vários tipos de medidas de distorção entre vetores que podem ser utilizadas em reconhecimento de voz. A medida de distorção mínima ou euclidiana é a medida mais conhecida. Outro tipo de medida de distorção que aproxima com maior fidelidade a medida de distorção de máxima verossimilhança (que é a medida que espelha a distorção entre variáveis aleatórias com distribuição gaussiana), é chamada de distância de Mahalanobis, descrita pela equação 12.

$$d[x,y] = (x-y)^T \sum^{-1} (x-y) \tag{12}$$

onde $d[x,y]$ é a medida de distorção entre o vetor x e o vetor y, e $\sum$ é a matriz diagonal de covariância do vetor y.

Podemos reescrever a equação 12, como mostra a equação 13.

$$d[x,y] = \frac{(x_1-y_1)^2}{\sigma_1} + \frac{(x_2-y_2)^2}{\sigma_2} + \ldots + \frac{(x_p-y_p)^2}{\sigma_p} \tag{13}$$

onde $d[x,y]$ é a medida de distorção entre o vetor x e o vetor y, p é a dimensão do vetor y e do vetor x, x_i é a i-ésima componente dimensional do vetor x, y_i é a i-ésima componente dimensional do vetor y, σ_i é a variância da i-ésima componente dimensional do vetor y.

Multisecção

A técnica de quantização vetorial descrita utiliza um *codebook* para cada comando a ser treinado (reconhecimento de comandos) ou um codebook por locutor (verificação de locutor). Essa técnica não preserva a característica temporal de tais amostras. Essa falta de caracterização explícita de aspectos seqüenciais das amostras pode ser remediada. Utilizamos para isso a técnica chamada Quantização Vetorial Multiseção (*Multi-Section Vector Quantization* – MSVQ).

O processo de MSVQ é idêntico ao processo usual de quantização vetorial, com a diferença que as amostras de treinamento são divididas em pedaços de mesmo tamanho. Uma palavra é falada com determinada duração, e se for repetida pelo mesmo locutor terá sua duração diferente da anterior. Mas se repartíssemos a palavra em partes de tamanhos idênticos, teríamos cada pedaço com, praticamente, a mesma informação. Geramos então um *codebook* para cada pedaço da palavra. Quando desejássemos avaliar uma palavra a ser classificada, repartiríamos também tal palavra e cada pedaço seria avaliado com o *codebook* correspondente. Isso evitará que vetores localizados, por exemplo, no final da amostra a ser classificada sejam associados a centróides gerados com vetores do início das amostras de treinamento. Dessa forma, a MSVQ faz com que cada vetor da amostra a ser classificada só possa ser associado a um centróide com alguma correspondência temporal com tal vetor. A figura 5 ilustra o processo de treinamento ou geração dos *codebooks* utilizando, por exemplo, N amostras de treinamento, subdivididas em três partes iguais. Quando uma amostra for classificada, ela seria também repartida em três partes iguais e cada parte só poderia ter seus vetores associados aos centróides pertencentes ao *codebook* correspondente àquela parte.

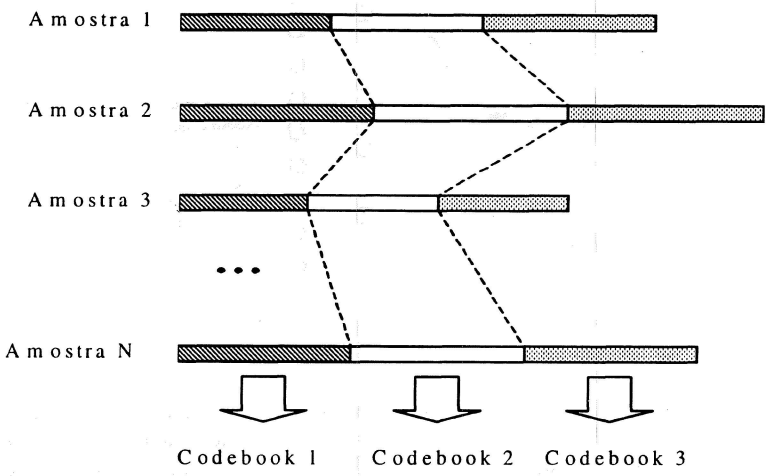

FIGURA 5: Treinamento da quantização multiseção

4 Metodologia Aplicada

Toda a análise teórica mostrada anteriormente não evidencia claramente como é utilizada a técnica de quantização vetorial para fazermos a classificação de um padrão desconhecido, representante de um comando vocal. As técnicas descritas são utilizadas para a realização do *treinamento de comandos* e *treinamento de locutores*, realizados a *priori*. Após o sistema ter sido treinado, é feito o *reconhecimento de comandos*, que na aplicação proposta identifica o andar que o usuário deseja ir. Feita a identificação do andar, é efetuada a *verificação de locutor*, que definirá se o usuário foi previamente cadastrado naquele andar. Só então é enviada a informação ao elevador de como proceder.

Treinamento de Comandos

Para o treinamento de comandos, deve-se primeiramente obter, para cada comando vocal a ser treinado, um conjunto de amostras de vozes digitalizadas que claramente reproduzem o comando em questão. Deve-se obter amostras do comando falado por diversos locutores diferentes, o que reforçará a desejada característica de independência ao locutor. A seguir deve-se extrair as características, *frame* a *frame*, de cada uma das amostras de vozes. O conjunto de vetores p-dimensionais resultante de tal extração de características estará relativamente agrupado em uma região do espaço p-dimensional. Esse conjunto de vetores será o conjunto de vetores de treinamento. Deve-se então utilizar o algoritmo de treinamento previamente descrito para geração dos centróides, agrupados em *codebooks*, representantes desses vetores de treinamento. Assim, haverá um *codebook* para cada comando a ser treinado. Isso finaliza a etapa de treinamento do classificador. Salienta-se que a etapa de treinamento do classificador é realizada a *priori*.

Reconhecimento de Comandos

Quando desejar-se reconhecer um comando desconhecido, representado por um conjunto de vetores p-dimensionais, deve-se quantizar cada um desses vetores utilizando, primeiramente, o primeiro *codebook* gerado no treinamento do classificador. Isso resulta em um conjunto de vetores quantizados diferentes do conjunto original. Mede-se a distorção entre esse conjunto quantizado e o conjunto original vetor a vetor, acumulando-se as distorções individuais. É feito o mesmo para cada um dos *codebooks* gerados na etapa de treinamento. O *codebook* que melhor quantizar o conjunto de vetores original será o *codebook* que estará mais próximo desses vetores. Para identificar qual *codebook* melhor quantizou nosso conjunto de vetores original, basta identificarmos qual o *codebook* que apresentou a menor distorção acumulada. O processo de reconhecimento de comandos, num universo de F grupos possíveis, está ilustrado na figura 6.

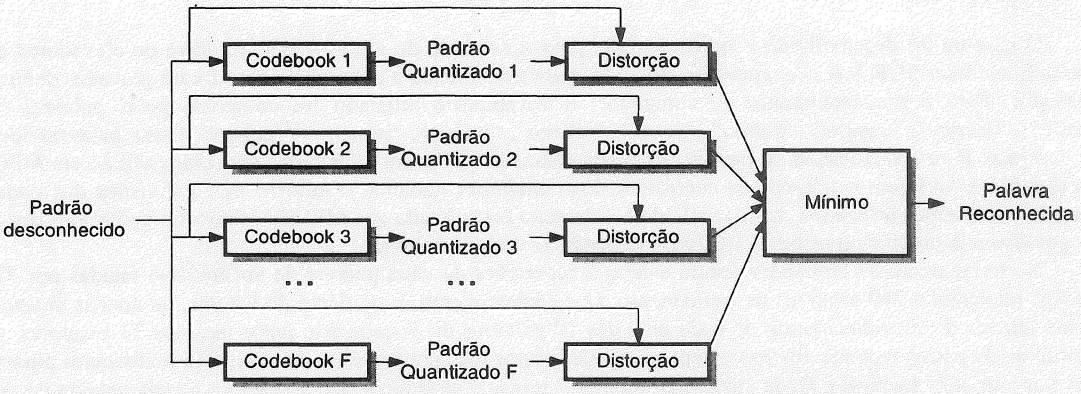

FIGURA 6: Processo de reconhecimento de comandos através da quantização vetorial.

Treinamento de Locutores

O processo de treinamento de locutores é bastante semelhante ao de treinamento de comandos descrito anteriormente. A diferença básica aqui é que teremos um *codebook* para cada um dos locutores cadastrados no sistema. O *codebook* de um determinado locutor é obtido utilizando amostras de voz daquele locutor específico É também estabelecido um limiar associado a cada *codebook*. O cálculo do limiar associado a cada *codebook* assume distribuição gaussiana [9]. Ele é estabelecido levando-se em conta as distorções geradas por dois tipos de amostras de voz: as que geraram o *codebook* as que sabidamente não pertencem àquele locutor. O limiar é calculado de acordo com a equação 14.

$$L_i = \frac{\mu_i^{in}\sigma_i^{out} + \mu_i^{out}\sigma_i^{in}}{\sigma_i^{out} + \sigma_i^{in}} \qquad (14)$$

onde *Li* é o limiar associado ao i-ésimo *codebook*, μ_i^{in} é a média entre os valores de distorção obtidos com as amostras de voz que pertencem ao i-ésimo locutor, μ_i^{out} é a média entre os valores de distorção obtidos com as amostras de voz que não pertencem ao i-ésimo locutor, σ_i^{in} é o desvio padrão correspondente às distorções obtidas com as amostras de voz que pertencem ao i-ésimo locutor, e σ_i^{out} é o desvio padrão correspondente às distorções obtidas com as amostras de voz que não pertencem ao i-ésimo locutor.

Verificação de Locutor

Para verificar a amostra de voz como pertencente a um determinado locutor primeiramente calcula-se a distorção acumulada obtida quantizando o padrão desconhecido com o *codebook* daquele locutor e comparando os vetores resultantes com os vetores do padrão inicial, vetor a vetor. Após, é feita uma comparação entre tal distorção e o limiar associado. Se a distorção acumulada for superior ao limiar, a amostra de voz que está sendo avaliada não é reconhecida como pertencente ao locutor em questão. Caso contrário, aceita-se a amostra de voz como pertencente àquele locutor. Faz-se realmente apenas uma verificação de locutor, ou seja, a amostra avaliada pertence ou não ao locutor.

5 Comunicação entre PC e Controlador

Foi estabelecido um protocolo de comunicação entre o computador pessoal e o microcontrolador 8051, que comanda o elevador. Foram desenvolvidas funções em linguagem C que manipulam a porta serial do computador pessoal. Um cabo conecta a porta serial do computador ao *hardware* do elevador. As funções desenvolvidas informam ao elevador que andar ele deverá ir.

Se o sistema determinar que o locutor não tem acesso ao andar requerido, o comando para o elevador mover-se a tal andar não é enviado. Uma mensagem pré-gravada é reproduzida ao usuário informando que seu acesso foi negado e o sistema volta a procurar a primeira palavra. Já se a identidade do usuário for aceita como pertencente ao andar requerido, as funções desenvolvidas são utilizadas e o elevador efetivamente desloca-se àquele andar.

6 Testes Realizados

O sistema foi desenvolvido e implementado para o controle de um elevador, na torre de elevadores da empresa brasileira Elevadores SÛR S.A.. As amostras utilizadas para treinamento e reconhecimento foram gravadas dentro da cabina do elevador. Para o reconhecimento de comandos, o vocabulário utilizado foi composto pelas palavras "primeiro", "segundo", "terceiro", "quarto", "quinto", "sexto", "sétimo", "oitavo", "nono" e "décimo". Essas palavras identificam o andar desejado. Para a verificação de locutor, a palavra utilizada foi "andar". Os testes realizados utilizaram MFCC com 26 filtros digitais, e variaram o número de centróides dos *codebooks* gerados, o número de subdivisões dos *codebooks* e o número de coeficientes utilizados. Uma janela de ponderação foi aplicada aos MFCC obtidos. Os centróides dos codebooks foram gerados utilizando o algoritmo LBG, usando distância de Mahalanobis.

No treinamento de comandos foram usadas 2 repetições de cada palavra de vocabulário faladas por 37 locutores diferentes, totalizando 740 arquivos de treinamento. O reconhecimento dependente de locutor, ou apenas *dependentes*, foi realizado através do reconhecimento de cada uma das 10 palavras do vocabulário pelos mesmos 37 locutores, totalizando 370 arquivos. Já para o reconhecimento independente de locutor, ou apenas *independentes*, foram utilizadas amostras de voz faladas por outros 24 locutores. Cada um dos 24 locutores testou o sistema de 2 a 3 vezes para cada palavra do vocabulário, totalizando 691 reconhecimentos.

Pode-se notar uma diminuição clara na taxa de reconhecimento quando aumentamos muito o número de coeficientes mel-cepstrais utilizados, como mostrado na figura 7. Pode-se assumir que o aumento excessivo do número de coeficientes não traz informação adicional relevante. Assim, o acréscimo de componentes sem muita informação é análogo a adição de ruído aos coeficientes, causando uma degradação nas taxas de reconhecimento.

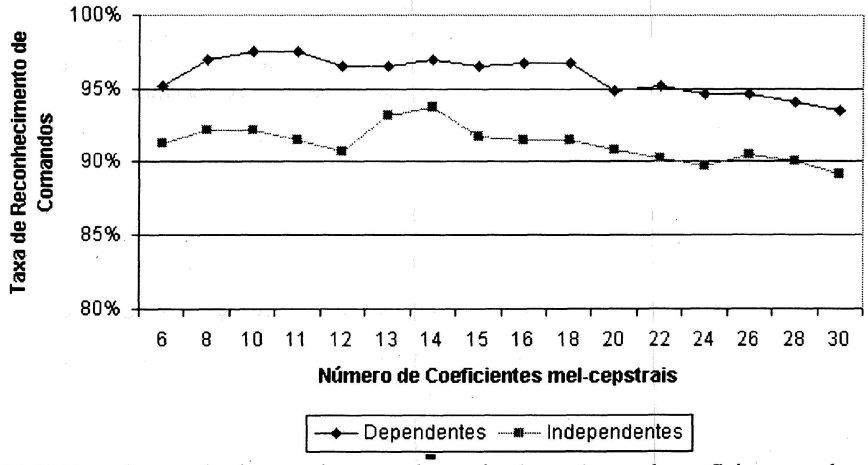

FIGURA 7: Taxa de reconhecimento de comandos variando o número de coeficientes mel-cepstrais

A análise da figura 8 indica um aumento nas taxas de reconhecimento de comandos, quando o número de subdivisões (ou subsecções) aumenta. Isso evidencia o melhoramento obtido através do uso da técnica de multisecção descrita anteriormente. Com o acréscimo das subdivisões, a relação temporal entre os vetores que deram origem aos *codebooks* e os vetores componentes do padrão desconhecido melhora.

Também pode-se notar que um aumento no número de centróides que cada *codebook* possui causará uma quantização mais exata, elevando a taxa de reconhecimento de comandos. A partir de um certo número de centróides, esse aumento passa a não ser mais muito significativo, não justificando o acréscimo de processamento requerido. Este é o caso quando aumenta-se o número de centróides de 4 para 8 na figura 8. Mas a elevação desse número de 2 para 4 centróides acarretou uma elevação significativa na taxa de reconhecimento.

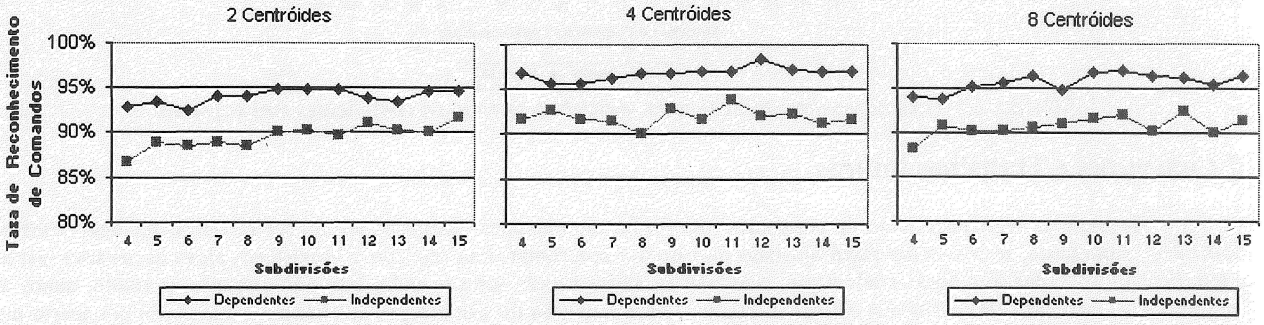

FIGURA 8: Taxa de reconhecimento de comandos variando o número de centróides por *codebook* e o número de subdivisões utilizando 14 MFCC

O treinamento de locutores foi realizado utilizando 50 locutores. Foram usadas 10 repetições da palavra "andar" para cada locutor, totalizando 500 arquivos. O cálculo do limiar necessita amostras de voz faladas por locutores não cadastrados. Assim, para cada um dos 50 locutores também são utilizados no treinamento 1 repetição da palavra "andar" falada por 37 locutores diferentes do 50 cadastrados.

No reconhecimento dos locutores cada um dos 50 locutores tentou acessar 10 vezes o andar que foi cadastrado. Essas verificações compõem a taxa de aceitação de locutores com direito de acesso, ou apenas *aceitação*. Cada um dos 50 locutores também tentou por 10 vezes acessar os outros 9 andares para o qual não tinha sido cadastrado. Isso totaliza 4500 tentativas de acesso sem permissão. Essas tentativas de acesso compõem a taxa de rejeição de impostores, ou apenas *rejeição*.

Na figura 9 nota-se que a rejeição a impostores aumenta, quando aumenta-se o número de centróides em cada *codebook*. Entretanto, a taxa de aceitação de locutores cadastrados cai de forma mais acentuada. Como nosso sistema deve otimizar ambas taxas, *codebooks* com 2 centróides são preferidos, apresentando inclusive menor necessidade de processamento.

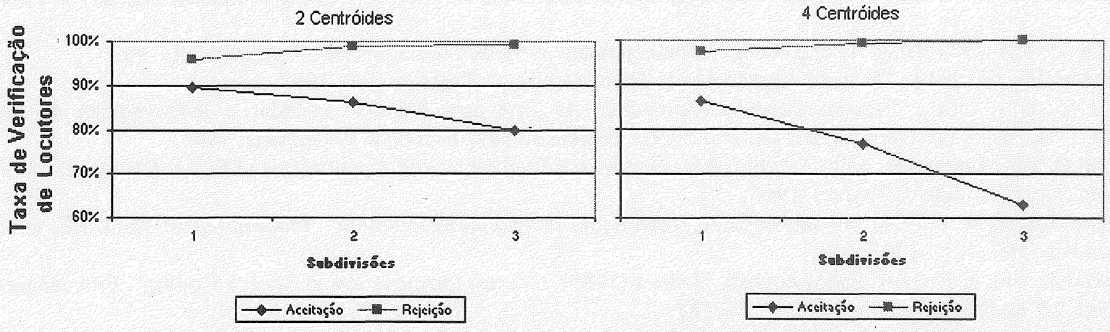

FIGURA 9: Taxa de verificação de locutor variando o número de centróides por *codebook* e o número de subdivisões utilizando 14 MFCC

Podemos notar na figura 10 uma queda gradual na taxa de rejeição de impostores, com o aumento do número de coeficientes mel-cepstrais utilizados. A taxa de aceitação de locutores com acesso não varia muito no intervalo para o número de coeficientes proposto. Fica claro, então, que a utilização de coeficientes excessivos prejudica a verificação de locutores, da mesma forma que o faz com as taxas de reconhecimento de comandos.

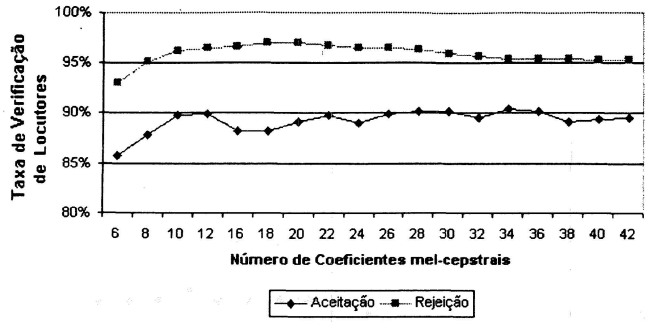

FIGURA 10: Taxa de verificação de locutor variando o número de coeficientes mel-cepstrais

7 Conclusões e Trabalhos futuros

As taxas de reconhecimento de comandos não foram tão elevadas quanto poderiam ser se outras técnicas fossem utilizadas. Entretanto, se o sistema fosse ajustado de forma a apresentar uma rejeição a impostores elevada, mesmo que a identificação do andar seja feita erradamente, o acesso ao andar errado seria negado. Isso não causaria, portanto, danos a segurança do edifício onde o sistema estiver instalado. Outra tentativa do usuário provavelmente ocasionaria um acerto na identificação do andar desejado. Na aplicação proposta, a questão da segurança deve ter prioridade, quando comparada ao acerto do andar que se deseja.

O reconhecimento de locutor é uma área específica que ainda tem muito a desenvolver. A extração de novos coeficientes, que representem de forma mais fiel as características singulares do locutor poderá nos levar à criação de sistemas mais confiáveis e robustos. Seria ainda importante que tais coeficientes fossem cada vez mais independentes do estado emocional do locutor, de anomalias momentâneas no aparelho respiratório (como resfriados ou inflamações) ou do ambiente onde é realizada a aquisição da voz. Isso facilitaria a implementação de sistemas onde a segurança e confiabilidade é imperativo, como o sistema descrito anteriormente.

8 Agradecimentos

Os autores gostariam de agradecer a Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) e o Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) por seu apoio financeiro.

9 Referências Bibliográficas

[1] John R. Deller, Jr. e John G. Proakis e John H. L. Hansen , "Discrete-time Processing of Speech Signals". Prentice Hall, 1987.

[2] Lawrence Rabiner e Biing-Hwang Juang , "Fundamentals of Speech Recognition". Prentice Hall, 1993.

[3] Panos E. Papamichalis , "Pratical Approaches to Speech Coding". Prentice Hall, 1987.

[4] Joel Augusto Luft , "Reconhecimento Automático de Voz para Palavras Isoladas e Independente de Locutor". Dissertação de mestrado, Universidade Federal do Rio Grande do Sul - Escola de Engenharia, 1991.

[5] David H. Kil e Frances B. Shin , "Pattern Recognition and Prediction with Applications to Signal Characterization". AIP Press, American Institute of Physics, 1996.

[6] Picone, Joseph W. (1993). "Signal Modeling Techniques in Speech Recognition". Proceedings of The IEEE, vol. 81, no. 9, september 1993, 1215-1247.

[7] Makhoul, John & Roucos, Salim & Gish, Herbert (1985). "Vector Quantization in Speech Coding". Proceedings of the IEEE, vol. 73, no. 11, novembre 1985, 1551-1587.

[8] Gray, Robert M. (1984). "Vector Quantization". Readings in Speech Recognition, 75-100.

[9] Burton, David K. (1987). "Text-Dependent Speaker Verification Using Vector Quantization Source Coding". IEEE Transactions on Acoustics, Speech, and Signal Processing, , vol. ASSP-35, 133-143.

[10] Rosenberg, A. E. & Soong, F. K. (1987). "Evaluation of a Vector Quantization Talker Recognition System in Text Independent and Text Dependent Modes". Computer Speech and Language, 22, 143-157.

[11] Soong, Frank K. & Rosenberg, Aaron E. & Juang, Biing-Hwang & Rabiner, Lawrence R. Rabiner (1987). "A Vector Quantization Approach to Speaker Recognition". AT&T Technical Journal, vol. 66, Issue 2, march/april 1987, 14-26.