UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA

SÍNTESE DE VOZ A PARTIR DE TEXTO PARA A LÍNGUA PORTUGUESA

FRANCISCO EGASHIRA

ORIENTADOR: PROF. DR. FÁBIO VIOLARO †

Dissertação apresentada à
Faculdade de Engenharia Elétrica da UNICAMP
como requisito parcial para a obtenção do título de
Mestre em Engenharia Elétrica

Este exemplar corresponde à redação final da tese
defendida por Francisco Egashira
e aprovada pela Comissão
Julgadora em 24/04/92
Fábio Violaro
Orientador

CAMPINAS
Julho de 1992

RESUMO

Este trabalho descreve os procedimentos e critérios utilizados na implementação de um sistema de conversão texto-voz para a Língua Portuguesa. O objetivo deste sistema é realizar uma síntese de voz de vocabulário irrestrito a partir de um texto de entrada genérico.

A geração de uma voz de boa qualidade, segundo critérios de inteligibilidade e naturalidade, exige que diversas etapas de processamento sejam realizadas, visando reproduzir os mesmos processos existentes na produção de fala natural. Neste trabalho, apenas alguns módulos foram implementados, por se tratar de um estudo inicial visando estabelecer um primeiro contato com a área.

O método de síntese utilizado é o de concatenação, tendo como unidade básica o difone. O difone é o elemento resultante da combinação de dois fones, limitado pela região estável dos fones e contendo a transição completa entre eles. Um dicionário de difones de cerca de 1000 elementos permite que a síntese de um texto de vocabulário irrestrito possa ser realizada.

A geração de voz a partir de texto, exige que os sons associados aos símbolos e letras existentes no texto sejam determinados. Isto é realizado pelo módulo de conversão ortográfica-fonética.

A fim de prover maior naturalidade à voz sintetizada, uma variação correta dos parâmetros prosódicos- duração, freqüência fundamental e amplitude- deve ser realizada. Neste trabalho, apenas algumas regras simples de controle da duração são consideradas.

Uma placa baseada no processador de sinais TMS320C30 é utilizada para realizar a síntese do sinal de voz. Isto permite que a síntese possa ser realizada em tempo real.

ÍNDICE

CAPÍTULO 1

INTRODUÇÃO

Neste capítulo procuraremos apresentar a estrutura de um sistema de síntese de voz a partir de texto, algumas de suas aplicações e um breve histórico de seu desenvolvimento.

1.1. CONSIDERAÇÕES INICIAIS

A Linguagem na sua forma falada é, para nós, um meio de comunicação bastante utilizado no dia a dia. Informações são obtidas e fornecidas de maneira rápida e informal através dela. Porém é incontestável que a maneira mais eficiente, econômica e confiável de se transmitir e armazenar informações é através da linguagem escrita. A escrita tem desempenhado papel fundamental ao longo da história como forma de comunicação. Isso não significa que a mensagem escrita seja sempre a forma mais desejada de se obter acesso a informações. Existem situações nas quais a mensagem falada é vantajosa por permitir que as mãos e os olhos estejam livres para realizar outras tarefas. Podemos por exemplo escutar as notícias no rádio enquanto dirigimos ou realizamos uma outra atividade.

Na interação com as máquinas, uma resposta ou mensagem de alerta falada pode ser mais eficiente que uma resposta visual. Uma mensagem de advertência, por exemplo, pode atingir as pessoas sem exigir que sua atenção esteja voltada para um ponto específico. Além disso, um sistema de resposta falada pode dar à máquina um aspecto mais humano, facilitando a interação do homem com a mesma.

Outro fato que contribui de maneira decisiva em favor da mensagem falada, é que ela permite o acesso através da rede de telefonia pública a sistemas de informação automatizados, sem a necessidade de equipamentos especiais. Pode-se desenvolver uma série de aplicações interativas onde o único meio de entrada é o terminal telefônico.

Apesar das vantagens que a mensagem falada pode oferecer em muitas situações, o seu uso ainda é bastante restrito. Isso se deve ao fato de as técnicas digitais de armazenamento e reprodução de voz só terem se desenvolvido recentemente. Os sistemas de gravação e reprodução analógicos, por outro lado, não são portáteis, são incômodos de serem operados, e pouco flexíveis em relação ao acesso às informações armazenadas.

O desenvolvimento de um sistema que aceite como entrada um texto escrito e produza na saída uma mensagem falada, busca exatamente aproveitar a facilidade de geração e armazenamento da mensagem na sua forma escrita para posterior geração de um sinal de voz.

Os sistemas de síntese podem se dividir em sistemas de *resposta de voz* e sistemas de *voz a partir de texto* . Os sistemas de resposta de voz operam com um vocabulário limitado. Sua operação consiste basicamente em uma gravação e armazenamento de todas as possibilidades de mensagens para uma posterior reprodução. Neste caso, os requisitos exigidos para tal sistema são: uma forma econômica de armazenamento, provido por uma técnica de codificação adequada, e uma forma eficiente de acesso. A codificação por predição linear (LPC) é com certeza a técnica mais eficiente para tal aplicação [1]. Os sistemas LPC permitem uma representação precisa e econômica, em termos de taxa de bits do sinal codificado, sendo possível encontrar comercialmente circuitos integrados dedicados que realizam a síntese LPC. Os sistemas de resposta de voz exigem pouco processamento de voz ou lingüístico. Isto por sua vez implica em uma quantidade de armazenamento que pode se tornar excessiva para um vocabulário grande.

Os sistemas de voz a partir de texto aceitam como entrada um texto de vocabulário ilimitado e, por meio de extensivo processamento lingüístico, produzem uma fala sintética do texto correspondente. Devido ao fato de estarmos operando neste caso com um vocabulário ilimitado, torna-se inviável armazenar todas as combinações de mensagens possíveis, e precisamos portanto partir para a escolha de unidades básicas menores, a partir das quais possamos construir as mensagens desejadas. Este fato leva à necessidade da realização do processamento lingüístico, capaz de extrair do texto informações relevantes que possam prever variações de parâmetros acústicos de modo a reproduzir o processo de produção natural da fala.

A figura 1.1 mostra basicamente os elementos que compõem o sistema de síntese de voz a partir de texto.

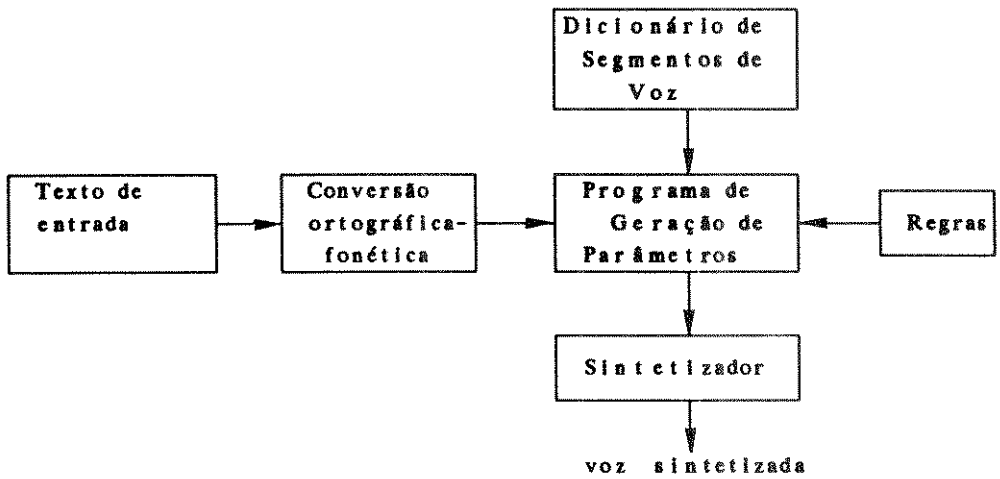


Fig. 1.1. Diagrama básico do sistema de conversão texto-voz

Inicialmente temos como entrada um texto na forma convencional, contendo símbolos, abreviações, algarismos e sinais de pontuação. Este texto é submetido a uma análise a fim de se determinar a sua composição fonética, ou seja, obter a pronúncia da palavra a partir de sua forma escrita. A palavra *cidade* por exemplo, teria uma possível pronúncia [sidad3i]. Em seguida, de posse do texto fonético e tendo acesso a um conjunto de regras fonéticas e fonológicas e um dicionário de unidades básicas, gera-se um conjunto de parâmetros que serão utilizados para controlar um sintetizador de voz. Este sintetizador procura modelar o processo de produção de voz do ponto de vista acústico. Através do conjunto de parâmetros de entrada e da variação destes no tempo, o sintetizador é capaz de produzir uma fala contínua.

O dicionário de unidades básicas armazena, sob uma forma paramétrica, sons isolados da fala correspondentes a uma produção estática. O conjunto de regras procura simular o aspecto dinâmico da fala, indicando como os parâmetros referentes às unidades básicas devem variar à medida que estas unidades se sucedem no tempo, e como eles se modificam face aos diferentes contextos em que se encontram. Para exemplificar tomemos como exemplo o parâmetro *freqüência fundamental*. Apesar da possibilidade dos segmentos sonoros serem armazenados com um freqüência fundamental própria, estes valores terão que ser modificados quando os diferentes segmentos forem concatenados para formar sentenças, a fim de se poder controlar a prosódia do enunciado. Assim, frases tais como *João gosta de Maria* e *João gosta de Maria?*, uma assertiva e a outra interrogativa, serão distingüidas pela diferença de entonação, e portanto os valores de freqüência fundamental serão impostos pelo enunciado. Podemos ter também modificações a nível segmen-

tal, nos efeitos de coarticulação por exemplo, e dependendo do tipo de unidades básicas com as quais estamos trabalhando, deveremos prever a existência de regras para realizar tais modificações.

Tanto o dicionário de unidades básicas como o sintetizador de voz, podem ser implementados de diferentes maneiras, dando origem a diferentes tipos de esquema de síntese. No capítulo 2 discutiremos algumas das possibilidades.

1.2. APLICAÇÕES DE SISTEMAS DE SÍNTESE DE VOZ

Tendo em vista os aspectos relativos à importância da fala apresentados, podemos perceber que existe um grande número de aplicações em potencial.

Conforme comentado anteriormente, as aplicações utilizando a rede de telefonia pública são muito atrativas. Porém, a fim de que todo potencial pudesse ser aproveitado, seria necessário que um sistema de reconhecimento de voz pudesse ser acoplado e utilizado como meio de entrada de informações. Isso possibilitaria oferecer serviços como reserva de passagens de avião automatizada, tal como no sistema experimental descrito em [2]. A tecnologia na área de reconhecimento de voz, entretanto, ainda apresenta resultados bastante limitados em relação à área de síntese devido à maior complexidade da natureza dos problemas envolvidos. Devido a esta limitação, os sistemas ficam restritos apenas a aplicações que exigem uma quantidade limitada de informação de entrada, como números por exemplo, que podem ser fornecidos pelo teclado ou disco do terminal telefônico. À medida que os sistemas de reconhecimento de voz forem evoluindo, a quantidade e abrangência das aplicações poderão ser ampliadas.

A seguir apresentaremos alguns exemplos de aplicações para os sistemas de síntese de voz.

- **Saldo bancário por telefone**

Neste caso temos um sistema de síntese de vocabulário limitado. Poderíamos utilizar um sistema de voz a partir de texto. Porém, para este tipo de aplicação, um sistema de resposta de voz pode ser mais vantajoso. A seguir faremos algumas considerações que devem ser levadas em conta no desenvolvimento de tal tipo de sistema.

O método de síntese utilizado é a gravação de palavras pronunciadas isoladamente e uma posterior concatenação destas para produção de sentenças. Apesar de não requerer tanto processamento quanto os sistemas de voz a partir de texto, é necessário realizar variações contextuais a fim de produzir uma fala natural e contínua. Isto acontece porque as palavras apresentam coarticulação e portanto sofrem variações conforme o contexto em que se encontram, fazendo que uma sentença completa seja diferente de uma seqüência de palavras pronunciadas isoladamente. Cada sentença possui um padrão específico de acentuação, ritmo e entonação, que depende de fatores sintáticos e semânticos. Caso esta informação não possa ser recuperada pelo sistema de síntese, a qualidade da fala ficará comprometida.

A falta de variação adequada dos parâmetros mencionados ocasiona descontinuidade e efeitos desagradáveis. Um estudo feito por J.P. Olive [3] analisa os aspectos relacionados com estes problemas. No caso, o estudo abordou a síntese de voz por meio de concatenação de palavras. A fim de poder prover variações contextuais, as palavras foram representadas de forma paramétrica. Portanto, após gravar as palavras, foi feita uma análise a fim de efetuar a extração dos parâmetros de pitch, freqüências formantes, e amplitude. Em seguida foi realizado um experimento para determinar quais parâmetros deveriam ser variados a fim de atingir uma prosódia próxima da natural. Os resultados deste experimento foram utilizados na elaboração de um conjunto de regras de variação dos parâmetros, em um sistema de síntese para geração de números telefônicos a partir da concatenação de dígitos gravados. O fato das palavras estarem representadas na forma paramétrica tornou possível, utilizando as regras, efetuar as variações necessárias para produzir uma fala mais natural. Estes parâmetros foram fornecidos a um sintetizador por formantes que convertia os parâmetros em sinal sonoro novamente.

Para um sistema de acesso a saldo bancário por telefone, deve-se realizar um estudo semelhante, visando um sistema de qualidade aceitável. O vocabulário exigido consiste aproximadamente dos seguintes elementos:

Algumas frases introdutórias tais como:

Bom dia.
Boa tarde.
Boa noite.
Entre com o número de sua senha.
Entre com o número de sua conta.

Alguns palavras básicas a partir das quais formamos os números tais como:

Um, dois, três, ..., nove.
Dez, onze, doze, ..., dezenove.
Vinte, trinta, quarenta, ..., noventa.
Cem, cento, duzentos, ..., novecentos.
Mil, milhão, milhões, bilhão, bilhões.

Este vocabulário deve ser guardado em uma forma paramétrica, utilizando por exemplo um dos modelos de produção de voz a serem descritos, a fim de podermos realizar posteriormente uma variação adequada dos parâmetros, visando uma continuidade da fala e entonação correta.

- **Acesso a informações pelo telefone tais como previsão meteorológica, eventos esportivos ou culturais, feiras e exposições, programação de teatro e cinema.**

Estes tipos de informação mudam constantemente e portanto precisam ser constantemente atualizadas por meio de gravações. Com um sistema de síntese de voz a partir de texto bastaria dispor das informações geradas e armazenadas na forma de texto, utilizando alguma forma de editoração por computador.

- **Máquina de leitura para cegos**

Uma importante aplicação dos sistemas de síntese de voz está relacionada ao auxílio para deficientes visuais. Existem esforços sendo feitos para desenvolver sistemas que possam reproduzir de maneira falada um determinado texto disponível na forma impressa. A primeira etapa consiste em transformar o texto impresso em uma forma adequada de entrada para o sistema de síntese. Esta entrada pode ser, por exemplo, o formato ASCII. Isto pode ser feito utilizando um scanner e em seguida aplicando um algoritmo de reconhecimento de caracteres.

Um exemplo de sistema desenvolvido, capaz de produzir voz a partir de material impresso é a máquina de Kurzweil [4]. Este sistema é capaz de ler em voz alta um livro comum. O custo desta máquina é alto devido em parte à sofisticação do algoritmo de reconhecimento de caracteres. Este algoritmo deve processar a imagem digitalizada produzida por uma câmera a fim de eliminar possíveis degradações. Em seguida cada letra deve ser isolada, e devem ser encontradas características geométricas que possam auxiliar na identificação do caracter em questão. O alto custo desta máquina dificulta o objetivo de torná-la um sistema pes-

soal, o que seria a situação ideal no auxílio a deficientes visuais. Neste sentido é interessante o desenvolvimento de um sistema que possa ser acoplado a um computador pessoal. Atualmente é muito comum que a maioria dos textos tenham sido produzidos utilizando algum editor ou processador de texto no computador. Desta forma, é possível ter acesso ao texto numa forma que pode ser utilizado diretamente no sistema de síntese.

Outra aplicação possível é o auxílio a deficientes visuais no desenvolvimento de atividades no trabalho. O trabalho em um escritório, por exemplo, pode ser facilitado caso um conversor texto-voz esteja acoplado a uma máquina de escrever ou terminal de computador. Neste caso, é possível ao operador verificar o texto digitado por meio de um retorno auditivo. Isso torna o trabalho dos deficientes visuais mais eficiente, pois evita a necessidade que seu trabalho seja conferido por uma pessoa de visão normal, tornando-os mais independentes.

- **Auxílio de fala a deficientes vocais**

Um sistema de síntese pode servir de auxílio para pessoas impossibilitadas de falar. Neste tipo de sistema um aspecto que deve ser considerado com cuidado é com relação à entrada do sistema. É preciso prover uma forma interativa de formar mensagens de entrada de modo a não torná-la um processo cansativo. A idéia no caso é poder fornecer apenas algumas palavras chaves a um sistema que gere, a partir destas, sentenças gramaticalmente corretas. Isso exige que o usuário passe por um período de treinamento. Uma vez construída a sentença, na forma de texto por exemplo, um sintetizador é utilizado para reproduzir a mensagem. Neste tipo de aplicação a prosódia desempenha um papel fundamental, sendo que através do seu controle é possível dar sentimento ao que está sendo falado, o que é importante num processo de comunicação.

Para que um sistema de síntese possa realmente auxiliar os deficientes da fala é preciso que ele seja portátil, de modo que possa servir como um aparelho de uso pessoal. Talvez esta seja uma das maiores limitações que devam ser superadas a fim de tornar esta aplicação difundida, pois a maioria dos sistemas desenvolvidos até agora funcionam acoplados a um computador de uso geral.

1.3. HISTÓRICO

A idéia de se gerar voz artificialmente já existe há algum tempo. Já em 1939 um sistema de síntese foi apresentado por Homer Dudley chamado de "Voder" [5]. O Voder consistia de chaves para seleção de uma fonte de ruído ou uma fonte sonora, com um pedal no pé para controle da freqüência fundamental. O sinal da fonte era transmitido através de uma seqüência de dez filtros passa-banda cujas amplitudes eram controladas manualmente por um operador. Na fig. 1.2 temos um diagrama mostrando a estrutura do Voder e sua analogia com o aparelho fonador humano.

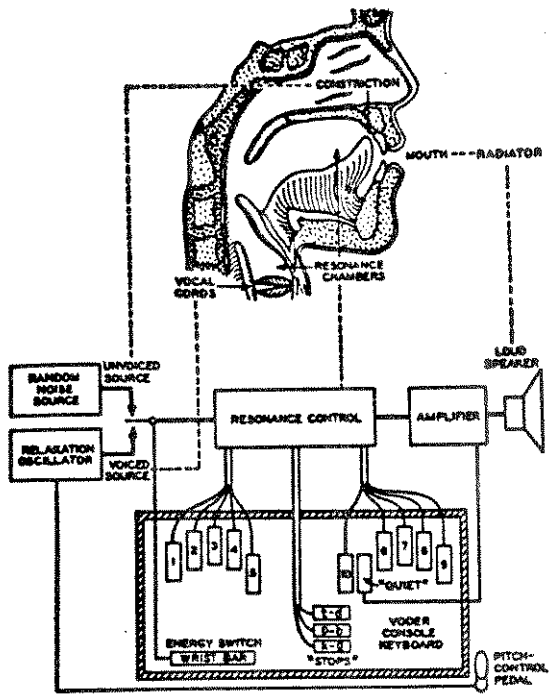


Fig. 1.2. O Voder (Dudley et al., 1989. [5])

Em 1950 um sintetizador chamado *Pattern Playback* [6] foi desenvolvido nos Laboratórios Haskins. Esta máquina realizava a função inversa de um espectrógrafo, ou seja, a partir de um espectrograma, era possível ouvir o som correspondente ao padrão fornecido. Seu funcionamento se baseia no rastreamento de um espectrograma, pintado sobre um filme transparente, por um feixe de luz modulado por uma roda tonal. As porções da luz modulada que são selecionadas pelo espectrograma, são coletadas por um sistema óptico e fornecidas a um elemento fotos-

sensível. A fotocorrente gerada é amplificada, e enviada a um alto-falante. Na fig. 1.3 temos um diagrama esquemático do *Pattern Playback*. Este dispositivo possibilitava a conversão de espectrogramas em som, tanto na forma original como na forma de padrões simplificados e estilizados, desenhados manualmente. Isto permitiu que estudos perceptuais fossem realizados acerca das pistas acústicas suficientes para a percepção de diferentes sons da fala [7]. Uma das principais contribuições destes estudos foi a constatação da importância das transições entre fonemas na percepção destes.

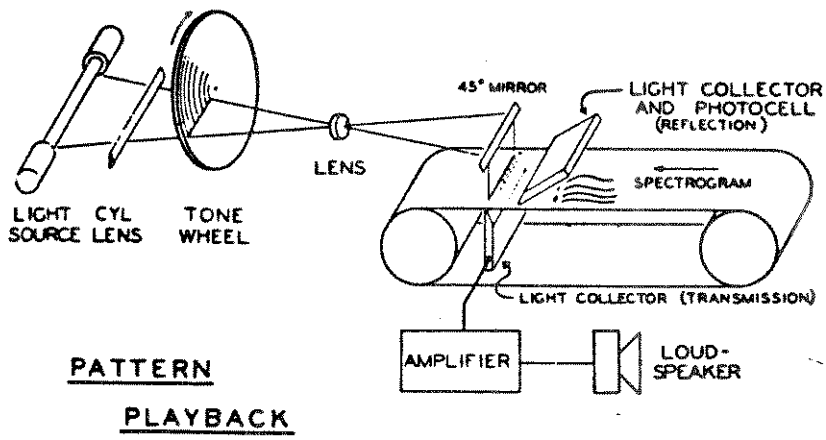


Fig. 1.3. O Pattern Playback (Cooper et al., 1951. [6])

Mais tarde, com o desenvolvimento de uma teoria acústica da produção de voz, surgiram modelos baseados na teoria fonte-filtro, que representam o sinal de voz como sendo a resposta de um filtro linear a uma excitação independente. Alguns desses modelos serão mais detalhados no próximo capítulo.

Os sistemas de síntese de voz a partir de texto para a língua inglesa começaram a ser desenvolvidos desde a década de 60. Atualmente é possível encontrar um grande número de sistemas experimentais e dispositivos comerciais. Um resumo bastante completo da conversão de texto para fala para a língua Inglesa se encontra em [8].

1.4 OBJETIVO DO TRABALHO

O objetivo deste trabalho foi iniciar o desenvolvimento de um sistema de síntese de voz a partir de texto para a Língua Portuguesa.

A tarefa de conversão de texto em voz é bastante complicada por estarem envolvidos na produção da fala tanto processos físicos como processos lingüísticos. A caracterização acústica dos sons de uma língua exige que estudos extensivos em laboratório sejam realizados. O estudo dos fenômenos prosódicos requer a montagem e análise de um corpus extenso. Estes estudos, bem como propostas de sistemas de conversão de texto em voz, são praticamente inexistentes para o Português. Isto nos levou à necessidade de iniciar um trabalho de base, procurando realizar um levantamento das publicações de pesquisas realizadas para outras línguas. Após esta etapa, escolhemos uma configuração e iniciamos o desenvolvimento de ferramentas computacionais para auxiliar na implementação do sistema.

A implementação do sistema visou sobretudo adquirir uma experiência na área de síntese de voz e estabelecer uma plataforma inicial, sobre a qual estudos posteriores mais avançados possam ser realizados.

1.5 ESTRUTURA DA TESE

No capítulo 2 procuraremos descrever alguns aspectos teóricos da produção acústica de voz a fim de poder compreender um dos componentes principais de um conversor texto/voz: o sintetizador baseado no modelo de produção de voz.

O capítulo 3 apresenta alguns conceitos básicos de lingüística, a fim de definir melhor alguns termos que serão utilizados ao longo da descrição do trabalho, e outros termos que freqüentemente aparecem em textos relacionados com processamento de voz.

O capítulo 4 descreve o sistema implementado, apresentando a configuração escolhida, os procedimentos e critérios utilizados na implementação do sistema, e as ferramentas desenvolvidas para auxiliar as várias etapas do processo.

O capítulo 5 trata dos processamentos realizados sobre o texto de entrada,

cujo objetivo é extrair as informações necessárias para a geração de parâmetros que serão utilizados pelo sintetizador na produção da voz.

O capítulo 6 descreve os procedimentos para geração dos parâmetros que serão utilizados pelo sintetizador na geração do sinal de voz.

O capítulo 7 descreve o hardware e o procedimento utilizado para implementar o sistema de síntese em tempo real, utilizando um processador de sinais digitais.

No capítulo 8 apresentamos os resultados do trabalho, as dificuldades encontradas no desenvolvimento do sistema e as propostas para continuação do trabalho e futuras melhorias

CAPÍTULO 2

MODELOS DE PRODUÇÃO DE VOZ

Neste capítulo apresentaremos a teoria que serviu de base para a criação de modelos de produção de voz e que deram origem aos sintetizadores atualmente utilizados.

2.1 TEORIA ACÚSTICA DE PRODUÇÃO DE VOZ

O desenvolvimento de sintetizadores de voz teve um grande impulso após a formulação de uma teoria acústica de produção de voz. Com base nesta teoria, vários modelos capazes de produzir voz artificialmente foram propostos. Esta teoria está muito bem apresentada em Fant [9] e Flanagan [10]. Alguns aspectos serão considerados a seguir, a fim de podermos compreender melhor os modelos de produção de voz que serão apresentados.

As ondas sonoras são criadas por meio de vibrações e se propagam em um meio material. Portanto, a geração e propagação de sons no sistema vocal, é regida pelas leis fundamentais da física como conservação de massa, momento e energia, juntamente com as leis da termodinâmica e mecânica dos fluídos. Podemos considerar o trato vocal como um tubo acústico de seção transversal não-uniforme, variante no tempo. Baseado nestas afirmações e desprezando as perdas de energia, podemos considerar que as ondas sonoras satisfazem as seguintes equações:

$$\frac{-\partial p}{\partial x} = \rho \frac{\partial (u/A)}{\partial t}$$

$$\frac{-\partial u}{\partial x} = \frac{1}{\rho c^2} \frac{\partial (pA)}{\partial t} + \frac{\partial A}{\partial t} \qquad (2.1)$$

onde

$p = p(x,t)$ é a pressão sonora no tubo, na posição x e no instante t.

$u = u(x,t)$ é a velocidade volumétrica, na posição x e no instante t.

ρ é a densidade do ar no tubo.

c é a velocidade do som.

A = $A(x,t)$ é a área da seção transversal do tubo, na posição x e no instante t.

A fim de resolver estas equações é necessário conhecermos as condições de contorno nos dois extremos do tubo, ou seja, nos lábios e na glote, bem como a função área $A(x,t)$. Os diferentes sons provenientes da fala são gerados à medida que $A(x,t)$ varia de acordo com a alteração dos articuladores (lábios, língua, mandíbula, etc.). Devido à dificuldade de se obter uma descrição detalhada de $A(x,t)$ para sons complexos, utilizam-se modelos simplificados a fim de facilitar a análise. Para uma primeira análise consideremos um tubo de seção uniforme e constante no tempo. Esta configuração corresponderia aproximadamente à produção de uma vogal neutra (*schwa*). Considerando ainda como condições de contorno uma pressão nula nos lábios e uma excitação glotal constituída por uma exponencial complexa, teremos:

$$A(x,t) = A$$

$$u(0,t) = u_G(t) = e^{j\Omega t}$$

$$p(l,t) = 0$$

onde

Ω é a freqüência angular

l é o comprimento do tubo.

Com as condições acima obtemos a seguinte solução no domínio da freqüência, relacionando as transformadas de Fourier da velocidade volumétrica nos lábios e da velocidade volumétrica de excitação:

$$V(\Omega) = \frac{U(l,\Omega)}{U_G(\Omega)} = \frac{1}{\cos(\Omega l/c)} \qquad (2.2)$$

Esta função representa a função de transferência relacionando a velocidade volumétrica de saída e entrada. Podemos verificar que a mesma apresenta ressonância nas freqüências onde o denominador se torna nulo, ou seja:

$$\Omega_n l/c = (2n - 1)(\pi/2) \qquad n = 1, 2, 3, \ldots$$

$$F_n = (2n - 1)c/(4l)$$

onde

$$F_n = \Omega_n/(2\pi)$$

Estas freqüências de ressonância são denominadas freqüências formantes nos estudos de produção acústica da fala. Considerando um comprimento típico de 17cm para o trato vocal e 340 m/s para a velocidade do som, chegaremos aos valores de 500, 1500 e 2500 Hz para os três primeiros formantes da vogal neutra, modelada pelo tubo uniforme.

O próximo passo seria considerar os efeitos de perda no trato vocal e radiação nos lábios. Utilizando métodos numéricos é possível encontrar soluções para diferentes configurações do trato vocal, considerando a função área correspondente a diferentes sons. Embora este procedimento forneça importante informação para a compreensão do processo de produção da fala, ele não provê um modelo adequado para a geração automática de voz. Os modelos práticos para geração de voz, fáceis de serem implementados na forma digital, podem ser esquematizados conforme a figura abaixo.

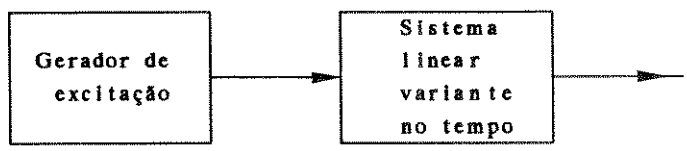


Fig. 2.1. Modelo fonte-filtro de produção de voz

Neste esquema os efeitos da fonte de excitação e do trato vocal são considerados independentemente. Os efeitos do trato vocal e da impedância de radiação são modelados pelo sistema linear variante no tempo. O gerador de excitação, por sua vez, pode fornecer dois tipos de entrada: pulsos periódicos para sinais sonoros e ruído aleatório para sinais não-sonoros.

2.2. MODELO EQUIVALENTE TERMINAL

O modelo equivalente terminal tenta representar o processo de produção de voz em seus terminais, de modo que o sinal de voz possa ser reproduzido na saída mesmo sem dominarmos o conjunto de eventos físicos que levaram à produção do respectivo som. Desse modo, teremos um modelo controlado por um conjunto de parâmetros que refletem o processo físico de produção de voz do ponto de vista da saída. Estes parâmetros variam no tempo de acordo com a dinâmica dos articulado-

res da fala. A atualização dos parâmetros é feita considerando que para pequenos intervalos de tempo, da ordem de 10 a 20 ms, o sistema fonador permanece fixo, e portanto o modelo deve ser atualizado aproximadamente a essa taxa.

Neste tipo de modelamento, é possível considerar o processo de produção de voz como um simples processo de filtragem, e o modelo mostra-se bastante apropriado para ser implementado na forma digital. Portanto, uma vez considerado o modelamento sob este ponto de vista, o problema consiste em se obter uma função de transferência para este filtro, que leve em conta os efeitos do pulso glotal, o trato vocal e a impedância de radiação nos lábios. Esta função de transferência, quando excitada por um trem de impulsos para sinais sonoros ou ruído branco para sinais não-sonoros, produzirá na saída o sinal de voz desejado.

A fim de obter uma função de transferência para o trato vocal, consideremos inicialmente o modelamento por um conjunto de N tubos sem perdas, de comprimentos l_i e áreas de seção transversal A_i conectados em série, onde i varia de 1 na glote a N nos lábios. Os valores de A_i são escolhidos de modo a aproximar a função área A(x) do trato vocal. Quanto maior o número de tubos de pequeno comprimento, melhor a aproximação.

Um modelo considerando a concatenação de quatro tubos é mostrado na fig. 2.2.

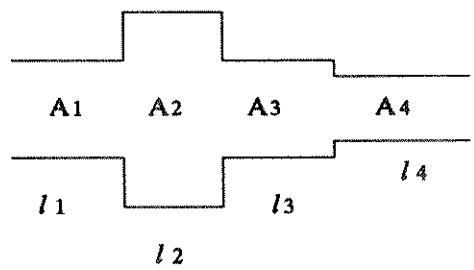


Fig. 2.2 Modelo de tubos uniformes conectados em série

Neste caso podemos considerar que cada tubo satisfaz a equação 2.1. Efetuando a análise deste sistema obtém-se a seguinte função de transferência:

$$V(z) = \frac{G}{1 - \sum_{i=1}^{N} a_i z^{-i}} = \frac{G}{\prod_{i=1}^{N} (1 - p_i z^{-1})} \qquad (2.3)$$

onde

G e a_i dependem das áreas e comprimentos dos tubos

N é o número de tubos

p_i é um pólo de V(z)

V(z) é a transformada Z do filtro equivalente

Os pólos de V(z) correspondem às freqüências formantes do sinal de voz. Para simular um determinado som, o seu espectro deve ser analisado a fim de se determinar quais posições os pólos devem ocupar para uma boa reprodução.

Este modelo apresenta somente pólos, e portanto não modela bem os sons fricativos e nasais. No caso dos sons nasais, a cavidade nasal é acoplada ao trato vocal gerando anti-ressonâncias [11]. De acordo com a função V(z) apresentada, o trato vocal pode ser considerado como um conjunto de ressonadores em cascata. A fim de contornar o problema da existência de zeros na função de transferência, alguns modelos utilizam uma estrutura em paralelo, a qual oferece um controle individual sobre as amplitudes dos formantes [12].

O efeito de radiação nos lábios é modelado por uma equação de diferenças de primeira ordem cuja transformada Z é dada por:

$$R(z) = 1 - z^{-1} \tag{2.4}$$

e apresenta uma inclinação da ordem de 6 dB/oitava, correspondendo a uma diferenciação do sinal no tempo.

A excitação deverá prover dois tipos de sinal: um para sons sonoros e outro para sons não-sonoros. No caso de fala sonora, a fonte de excitação deverá ser um trem de pulsos periódicos. O formato dos pulsos deve tentar reproduzir da melhor maneira possível os pulsos observados na fala natural. A formação dos pulsos é geralmente obtida a partir de um gerador de impulsos seguido por um filtro de função de transferência G(z), o qual transforma os impulsos em pulsos de formato adequado.

O modelo completo pode ser apresentado conforme a figura 2.3.

O modelo da fig. 2.3 serviu de base para o desenvolvimento de vários sintetizadores e foi implementado de diversas maneiras. Discutiremos mais detalhadamente dois desses sintetizadores: o modelo de síntese por formantes e o modelo de análise/ressíntese utilizando a teoria de codificação por predição linear (LPC).

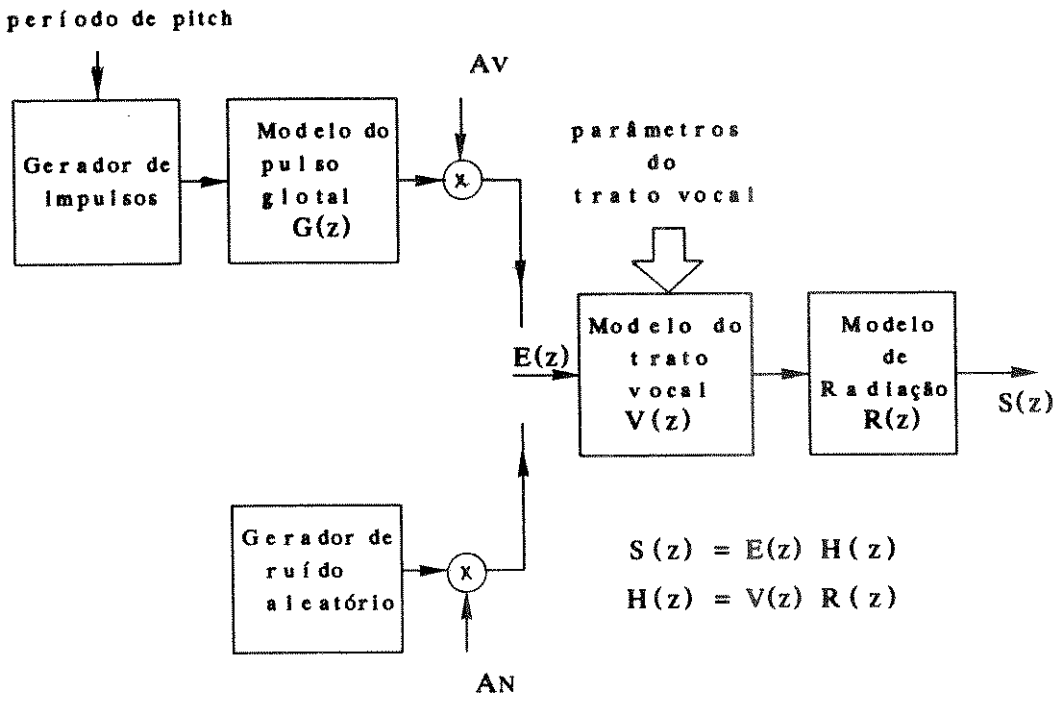


Fig.2.3. Modelo equivalente terminal completo.
(Rabiner & Schaffer, 1979. [13])

2.3 SÍNTESE POR FORMANTES

No modelo de síntese por formantes, um determinado espectro de voz pode ser obtido fornecendo-se as freqüências formantes e as respectivas larguras de banda correspondentes a cada par de pólos.Estes parâmetros são fornecidos a um conjunto de filtros, geralmente implementados como estruturas de segunda ordem com um par de pólos complexos conjugados. Estes filtros podem estar associados em cascata ou em paralelo. A associação em cascata não consegue levar em consideração a existência de zeros no espectro, a menos que sua ordem seja bastante elevada [13]. Na associação em paralelo é possível ter controle individual sobre a amplitude dos picos de cada formante e portanto ajustar o espectro adequadamente, levando em conta a influência dos zeros sobre os formantes na composição do espectro total.

As associações em paralelo e em cascata são mostradas na fig. 2.4.

A seguir apresentaremos alguns aspectos de um modelo cascata/paralelo desenvolvido por Klatt [14], e que pode ser implementado em software.

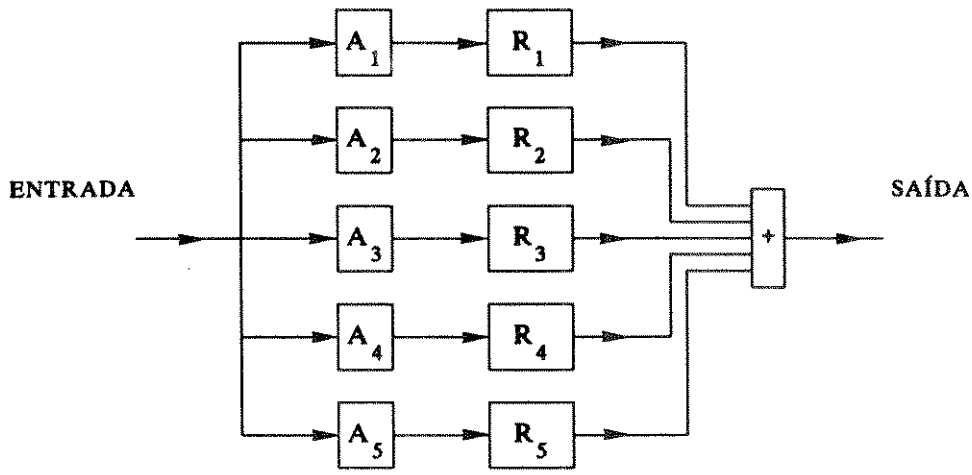


Fig. 2.4. (a) Associação de ressonadores em paralelo

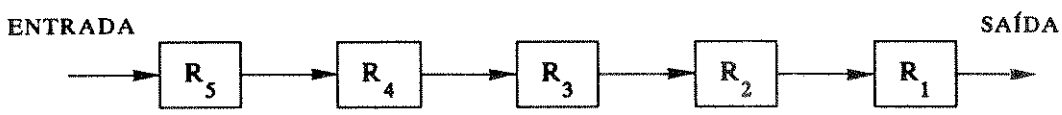


Fig. 2.4. (b) Associação de ressonadores em cascata

Na fig. 2.5 temos um diagrama mostrando as principais partes do sintetizador.

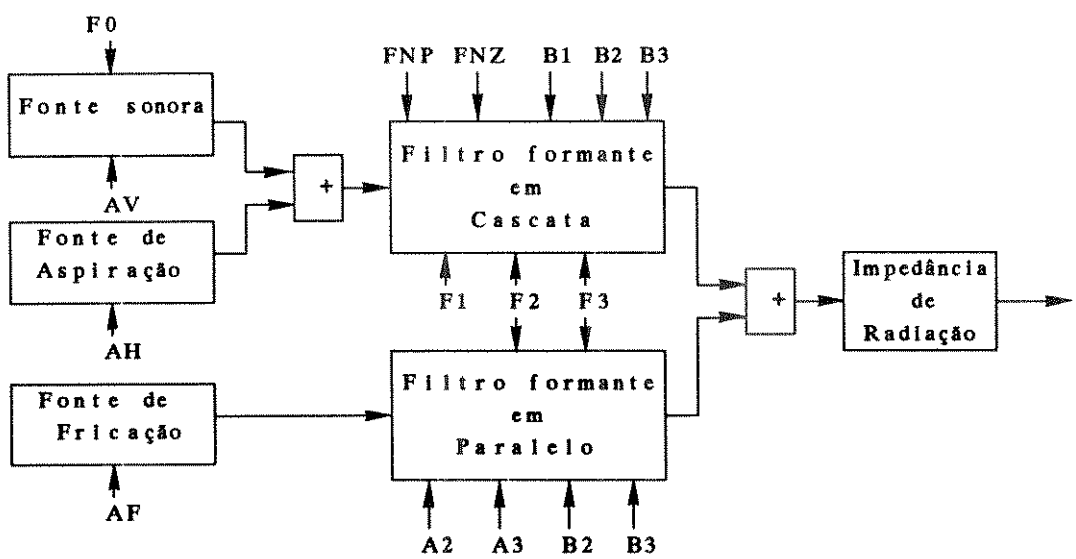


Fig. 2.5. Sintetizador cascata/paralelo de Klatt. (Klatt, 1980. [14])

Este sintetizador possui 39 parâmetros de controle, que são atualizados a cada 5 ms, operando com uma freqüência de amostragem de 10 kHz em uma configuração típica, podendo estes valores serem alterados. No diagrama estão representados apenas alguns dos parâmetros mais importantes que devem ser variados a fim de se sintetizar um determinado som. Estes parâmetros são:

AV	amplitude da fonte sonora
AF	amplitude da fonte de fricação
AH	amplitude da fonte de aspiração
F0	freqüência fundamental de excitação sonora
F1-F3	freqüência dos três primeiros formantes
FNP	freqüência do pólo nasal
FNZ	freqüência do zero nasal
B1-B3	largura de banda dos três primeiros formantes
A2-A3	amplitude dos formantes de ordem 2 e 3

Apesar de não estarem representados na figura, o modelo possui um total de 6 ressonadores.

A estrutura básica do sintetizador é formada pelo conjunto de ressonadores, os quais são conectados em cascata ou paralelo para simular a função de transferência do trato vocal. O conjunto de ressonadores conectados em série é utilizado para simular as características do trato vocal para fontes sonoras localizadas na laringe. Neste caso a função de transferência apresenta somente pólos, desde que não esteja envolvida a produção de sons nasais. Para os sons fricativos, que apresentam uma fonte localizada acima da laringe, a função de transferência do trato vocal apresenta a existência de pólos e zeros [15]. Os ressonadores conectados em paralelo simulam a presença dos zeros, cujo efeito é provocar alteração das amplitudes dos formantes adjacentes. Este efeito é simulado através do ajuste adequado dos controles de amplitude dos formantes.

A estrutura do ressonador digital, um filtro IIR de segunda ordem, é apresentado na fig. 2.6.

A equação de diferença que relaciona a entrada x(n) à saída y(n) é dada por:

$$y(n) = a_1 x(n) + a_2 y(n-1) + a_3 y(n-2) \qquad (2.5)$$

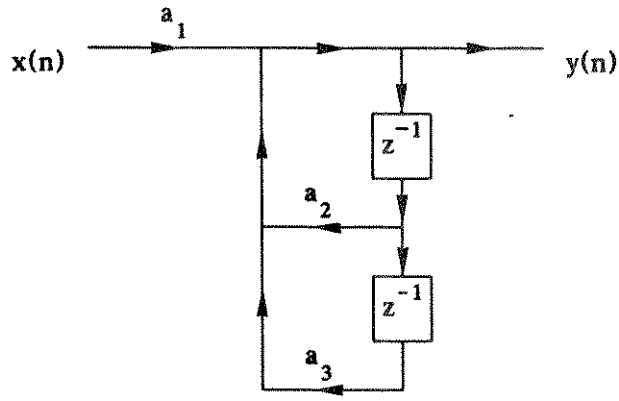


Fig. 2.6. Ressonador digital de segunda ordem

Os coeficientes a_1, a_2, a_3 são função da freqüência de ressonância e largura de banda do ressonador, relacionados pela transformação de invariância da resposta impulsiva [16].

$$a_3 = -e^{(-2\pi BT)}$$

$$a_2 = 2e^{(-\pi BT)}\cos(2\pi FT)$$

$$a_1 = 1 - a_3 - a_2$$

onde

F é a freqüência de ressonância do ressonador em Hz

B é a largura de banda do ressonador em Hz

T é o período de amostragem em segundos

A função de transferência do ressonador é dada por:

$$R(z) = \frac{a_1}{1 - a_2 z^{-1} - a_3 z^{-1}} \qquad (2.6)$$

As fontes de excitação podem produzir dois tipos de excitação: uma fonte de excitação sonora e uma fonte de ruído. A fonte de excitação sonora consiste de um trem de impulsos cuja saída é submetida a uma filtragem passa baixas a fim de produzir pulsos que se assemelhem aos pulsos glotais típicos. A fonte de ruído simula o ruído de turbulência produzido pela passagem do ar por uma constricção. Se esta constricção está localizada no nível das cordas vocais como na produção de [r] por exemplo, o ruído é chamado de aspiração. Se a constricção se localiza

acima da laringe como no caso de [s], o ruído é chamado de ruído de fricação. Para simular a produção de ruído é utilizado um gerador de números aleatórios que produz uma distribuição pseudo-gaussiana.

A aproximação de sons nasais é conseguida pela inserção de um ressonador e um anti-ressonador no modelo do trato vocal em cascata. A função destes circuitos é simular o aparecimento de pólos e zeros adicionais devido à presença de um ressonador (acústico) lateral presente nos sons nasalizados. O efeito deste ressonador e anti-ressonador é eliminado nos sons não nasalizados igualando a freqüência do pólo à freqüência do zero, provocando deste modo o cancelamento de ambos.

A característica de radiação modela o efeito dos padrões de radiação do som em função da freqüência. Esta característica é simulada pela seguinte equação de diferenças, relacionando a pressão sonora p(n) e a velocidade volumétrica u(n):

$$p(n) = u(n) - u(n - 1) \tag{2.7}$$

sendo a relação no domínio da freqüência dada por:

$$\frac{P(z)}{U(z)} = 1 - z^{-1} \tag{2.8}$$

A dificuldade da utilização do modelo apresentado é a obtenção de parâmetros para operá-lo. A metodologia utilizada é de obter parâmetros a partir de uma produção natural da fala e tentar ajustar o modelo até conseguir um bom casamento espectral entre o sinal obtido artificialmente e a amostra de fala natural. O acompanhamento de formantes é um processo bastante difícil e que por enquanto não está totalmente automatizado. Portanto, muitas vezes é preciso usar um processo de tentativa e erro para se chegar a valores de freqüências formantes e larguras de banda adequadas. Além disso, como veremos mais adiante, é preciso prover a evolução adequada dos parâmetros no tempo a fim garantir que a fala obtida seja de boa qualidade. Apesar dessas dificuldades, este é um modelo que consegue reproduzir com boa qualidade os sons da fala quando operado corretamente.

2.4. ANÁLISE/RESSÍNTESE UTILIZANDO TÉCNICA LPC

A análise LPC é uma importante técnica de codificação que tem sido utilizada em diversas aplicações. A sua popularidade reside no fato de permitir uma representação precisa e compacta de parâmetros espectrais de voz, obtida com um baixo custo computacional.

Os sistemas que utilizam análise LPC modelam o processo de produção da fala conforme o esquema mostrado na Fig. 2.7.

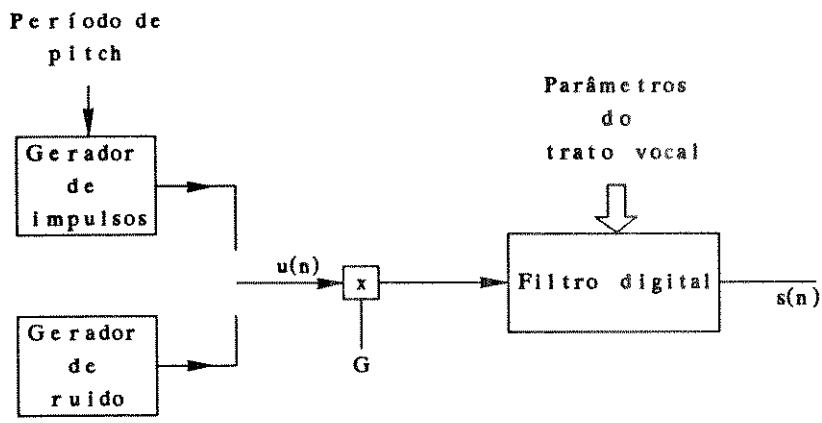


Fig. 2.7. Esquema do vocoder LPC

Neste modelo os efeitos espectrais compostos da impedância de radiação, o trato vocal e a excitação glotal são representados por um filtro digital variante no tempo cuja função de transferência é da seguinte forma:

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \qquad (2.9)$$

A excitação de entrada pode ser implementada de diferentes maneiras. No caso do vocoder (voice coder), onde um tipo de excitação simples é utilizado, o sistema é excitado por um trem de impulsos para fala sonora e uma sequência aleatória para fala não-sonora. Portanto, os parâmetros deste modelo são: classifição sonora/não-sonora, período de pitch para fala sonora, o ganho G, e os coeficientes $\{a_k\}$ do filtro digital.

Estes parâmetros são atualizados a cada 20 ms aproximadamente para levar em

conta a natureza variante do sinal de voz ao longo do tempo.

A determinação dos coeficientes do filtro $\{a_k\}$ e do ganho G é feita aplicando a este modelo a teoria de predição linear. Esta teoria estabelece que uma amostra do sinal de voz pode ser estimada a partir das p amostras anteriores do sinal, ponderadas pelos coeficientes do filtro. Baseado na minimização de uma função erro quadrático, e definindo convenientemente o intervalo de avaliação do erro, podemos chegar a um conjunto de equações lineares que, por apresentarem simetria, podem ser resolvidas de maneira eficiente pelo método de Levinson-Durbin [17].

A decisão sonoro/não-sonoro e o período de pitch no caso de sinal sonoro, devem ser determinados por meio de um algoritmo específico. A obtenção de um algoritmo que seja eficiente constitui a etapa de maior dificuldade na implementação do vocoder LPC.

A principal utilização da técnica LPC tem sido em aplicações de análise/ ressíntese para apresentação de mensagens pré-gravadas. O sistema LPC se apresenta muito interessante nestas aplicações pois sua representação espectral na forma paramétrica permite que um armazenamento econômico seja efetuado, diminuindo portanto a quantidade de memória exigida.

Um dos principais problemas apresentados pelo vocoder LPC reside no tipo de excitação que é utilizado. Neste sintetizador temos apenas dois tipos de excitação, a escolha devendo ser feita entre som sonoro/não-sonoro. Alguns sons tais como os fricativos sonoros, apresentam dois tipos de fonte de excitação, sendo modelados de maneira inadequada.

A fim de minimizar estes problemas, um novo modelo de exictação foi proposto por Atal & Remde [18]. Neste novo tipo de modelo chamado de LPC multi-pulso, a excitação é constituída por um conjunto de pulsos cujas amplitudes e posições são obtidas por um processo de análise por síntese, conforme o esquema mostrado na fig. 2.8.

A determinação dos impulsos constituintes da excitação é efetuada por um procedimento recursivo. Inicialmente uma excitação nula é aplicada ao filtro LPC. A saída do filtro é calculada e subtraída do sinal de voz original para descontar a memória do quadro anterior. Em seguida um impulso é adicionado à excitação; sua posição e amplitude são escolhidas de modo a minimizar o sinal de erro. Este erro é processado por um filtro perceptual, cuja função é enfatizar o erro fora da região dos formantes, onde ele é mais perceptível. O efeito deste

pulso é descontado do sinal de voz e o ciclo se repete até atingirmos um número de impulsos determinado. Neste modelo não é necessário efetuar a decisão se um determinado segmento de fala é sonoro ou não-sonoro.

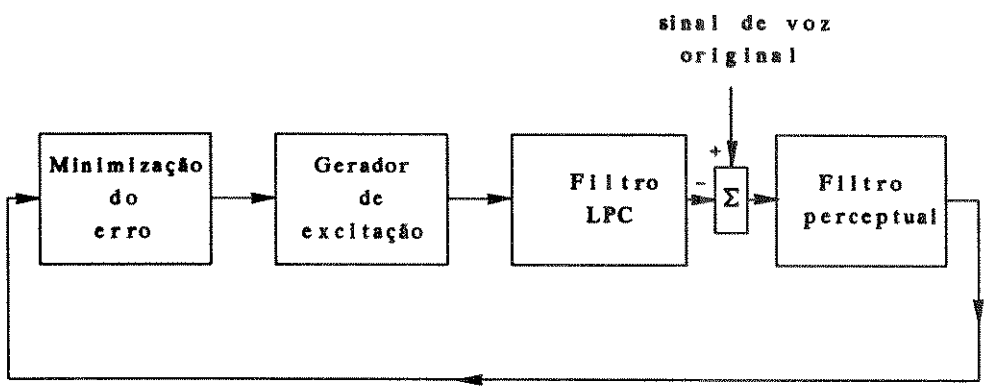


Fig. 2.8 Sistema multi-pulso

Nas aplicações de síntese a partir de texto utilizando técnica LPC, é empregado um procedimento de análise/ressíntese a partir de elementos pré-gravados, obtidos a partir de uma produção natural da fala. Inicialmente gera-se um conjunto de elementos básicos, e a partir da concatenação destes, pode-se produzir uma mensagem genérica desejada.

2.5. COMPARAÇÃO ENTRE OS MODELOS DE SÍNTESE POR FORMANTES E ANÁLISE/ RESSÍNTESE EM APLICAÇÕES DE SÍNTESE DE VOZ A PARTIR DE TEXTO.

O vocoder LPC, conforme apresentado, mostra-se bastante interessante para ser utilizado em sistemas de síntese, pois possui um processo bastante imediato para obtenção de parâmetros. Os algoritmos de síntese são relativamente simples de serem implementados em circuitos integrados. Possui a desvantagem porém, de possuir uma qualidade um pouco artificial, um certo aspecto metálico. Outro problema que apresenta é que o modelamento do trato vocal é feito levando em consideração apenas a existência de pólos. Com isso, certos tipos de sons não são modelados adequadamente. Além disso, a excitação se mostra muitas vezes inadequada para modelar certos tipos de sons que apresentam uma excitação mista. A utilização de uma excitação mais sofisticada como a excitação multi-pulso, por

exemplo, consegue compensar em boa parte as limitações apresentadas acima. Porém, a dificuldade que surge é que nos sistemas de síntese de voz a partir de texto, é preciso prover uma variação da freqüência fundamental de acordo com a prosódia do enunciado, determinada pelo texto. Isto implica dizer que os elementos pré-gravados que constituem o vocabulário básico, e a partir dos quais a síntese será efetuada, deverão ser sintetizados com uma freqüência fundamental diferente da original. Como no modelo multi-pulso LPC a excitação é determinada de maneira dependente do filtro de síntese, não é possível variar livremente a excitação como no caso do vocoder LPC, onde a excitação é determinada independentemente do filtro. Portanto, a utilização do sintetizador multi-pulso exige uma técnica especial para realizar a variação da freqüência fundamental, resultando em um aumento relativo de complexidade [19].

O modelo de síntese por formantes possui parâmetros que estão muito proximamente relacionados com o processo de produção e transmissão do som no trato vocal. Assim, podemos obter uma boa qualidade de voz se pudermos fornecer os parâmetros adequados ao modelo, e efetuarmos suas transições ao longo do tempo corretamente. Podemos ter controle sobre a forma de onda da excitação e obter diferentes qualidades de voz. O modelo permite que sejam considerados pólos e zeros na função de transferência. Desse modo, este modelo possui maior liberdade na escolha de parâmetros para o modelamento da fala.

Na síntese utilizando análise LPC, a qualidade de voz obtida já foi estabelecida no momento da análise, não sendo possível alterá-la no momento da síntese, o que pode ser importante para implementação de algumas regras que alteram as características dos sons baseados no contexto em que se encontram (coarticulação).

A principal desvantagem dossintetizadores por formantes é a dificuldade de se obter parâmetros para operá-los. O acompanhamento de formantes é um processo difícil de ser realizado precisamente de maneira automática. Estes parâmetros devem ser obtidos a partir de uma análise da fala natural. Os parâmetros devem ser atualizados a cerca de cada 10 ms e deve-se prover que as transições entre diferentes sons sejam realizadas adequadamente. Isto é necessário para evitar descontinuidades espectrais, o que pode causar aparecimento de ruído na saída ou perda de clareza no enunciado.

Em resumo, a síntese por formantes pode produzir voz sintetizada de excelente qualidade, mas o processo de obtenção de parâmetros pode ser penoso e muitas vezes envolve um processo de tentativa e erro.

2.6. MODELO ARTICULATÓRIO

Nos modelos discutidos até agora, tentou-se representar o trato vocal sob o aspecto terminal, ou seja, simular o processo de produção de voz em termos de saída. Este procedimento foi adotado visando obter um modelo simples de ser implementado na prática. Os modelos articulatórios procuram, por sua vez, modelar os movimentos do trato vocal, a fim de tentar reproduzir de maneira mais próxima do natural a produção da fala [20]. Nestes modelos, um conjunto de parâmetros correspondente aos articuladores é controlado por um conjunto de regras, que prevêem um seqüência de movimentos correspondentes a sucessivos fonemas.

Valores de referência destes parâmetros são armazenados para cada fonema e a variação destes parâmetros é obtida fazendo-se uma interpolação entre estes valores de referência.

As regras para controle dos parâmetros articulatórios geralmente são obtidos a partir de dados de observações do trato vocal durante simples enunciados.

Os modelos articulatórios, apesar de poderem apresentar uma boa qualidade para determinados sons, não tiveram uma boa aceitação para serem implementados em sistemas práticos. As principais razões disto foram as dificuldades em se obter dados para determinação dos parâmetros articulatórios e o alto custo computacional que estes modelos apresentam.

CAPÍTULO 3

CONSIDERAÇÕES LINGÜÍSTICAS

Neste capítulo apresentaremos alguns conceitos lingüísticos básicos.

3.1. ATRIBUTOS BÁSICOS DA FALA

A fala pode ser analisada sob dois aspectos: o aspecto físico e o sistema que a organiza.

O aspecto físico da fala

Os estudos iniciais sobre a fala procuram analisá-la sob o aspecto físico, considerando o sinal de voz como uma onda sonora produzida e propagagada no sistema vocálico humano. Pode-se estudar o processo de produção da fala e, conforme visto no capítulo 2, derivar modelos capazes de reproduzir artificialmente um sinal de voz.

Porém, a fala não se resume a um processo físico de produção de uma onda sonora, pois possui uma função comunicativa, e no sinal de voz está codificada uma mensagem.

Os sintetizadores modelam a fala apenas no seu aspecto físico, no sentido de simplesmente serem capazes de produzir em suas saídas os mesmos tipos de sinais acústicos que o sistema vocal humano produz. A geração de mensagens completas, portadoras de significado, dependerá da capacidade de se fornecer os parâmetros adequados para realização da síntese. Este processo de escolha de parâmetros será mais efetivo na medida em que pudermos segmentar o sinal de voz em suas partes componentes elementares, e recombinar estes elementos para criar novas mensagens. Esta segmentação não é tão imediata de ser realizada, pois estamos tratando com um processo, a produção acústica de voz, que por natureza é um processo contínuo e não discreto.

A partir deste ponto começa a importância de entendermos o processo de produção da fala não no seu aspecto físico, mas em um nível mais abstrato, procurando entender a existência de um sistema que organiza a fala e como este sistema utiliza o sinal de voz para codificar uma mensagem.

Sistemas que organizam o processo da fala

A fim de se avaliar a importância da existência de um sistema que controle e organize o processo da formação de mensagens faladas como forma de linguagem, observemos a flexibilidade do sistema vocal humano. Na figura 3.1 temos uma vista da seção transversal do trato vocal indicando os articuladores da fala. Estes articuladores podem alterar a área da seção sagital do trato vocal, alterando suas características de ressonância. Se nós considerarmos os graus variáveis de estreitamento de cada um destes pontos de articulação e as possibilidades de suas combinações simultâneas, chegaremos à conclusão que o número de graus de liberdade do trato vocal e, conseqüentemente, o número de sons acusticamente diferentes que ele pode produzir é imenso.

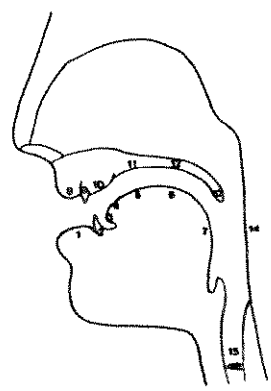


Fig. 3.1. Vista da seção sagital do trato vocal. (1) lábio inferior, (2) incisivo inferior, (3) ponta da língua, (4) dorso, (5) frente, (6) costas, (7) raiz, (8) lábio superior, (9) incisivo superior, (10) alveolo, (11) palato duro, (12) velum, (13) uvula, (14) faringe, (15) laringe, (16) cordas vocais e glote.

Para comunicar informação a um ouvinte, o falante produz um sinal de voz por meio de comandos neuromotores enviados aos músculos do trato vocal, fazendo os articuladores se movimentarem, alterando a forma do trato vocal. Para que esta comunicação seja efetiva, é necessário a existência de um sistema que exer-

ça domínio e opere sobre o número virtualmente infinito de efeitos acústicos possíveis de serem produzidos. Este sistema seleciona um subconjunto de sons e então organiza estes em um pequeno número de classes de sons básicas para comunicação. Este é o sistema fonológico da língua.

O sistema fonológico permitirá que um conjunto de diferentes sons - diferentes no sentido de uma articulação precisa, com seus correspondentes efeitos acústicos - sejam mapeados em uma mesma classe básica ou *fonema*, desempenhando a mesma função comunicativa. O ramo da Lingüística que estuda os sons da fala sob este aspecto funcional é chamado de *Fonologia*. Enquanto a *Fonética* procura obter informações acerca das propriedades acústicas e articulatórias do sinal sonoro, a fonologia atua utilizando critérios puramente lingüísticos para o agrupamento e classificação do material obtido pela fonética. Desse modo, para a fonologia, dois sons foneticamente distintos poderão ser considerados o mesmo, caso apresentem as mesmas propriedades funcionais, entendo por funcional aquilo que é relevante para fins de comunicação. O fato por exemplo do *r* ser pronunciado de maneira diferente no dialeto *caipira* não impede que as palavras possam ser interpretadas e compreendidas corretamente.

Tanto a Fonética quanto a Fonologia propõem a possibilidade de se segmentar a fala em unidades seqüenciais mínimas. Para a Fonética estas unidades são chamadas de *segmentos fonéticos* ou *fones*, enquanto na Fonologia as unidades seqüenciais mínimas são os *traços distintivos*, que quando combinados darão origem aos *fonemas*.

3.2 FONOLOGIA

Fonemas

Conforme a discussão anterior, vimos que a partir do conjunto de todos os sons que o trato vocal humano é capaz de produzir, cada língua escolhe apenas um número bastante limitado para produzir contrastes semânticos. Por exemplo, no Português temos /m/ e /s/ como unidades sonoras distintivas, tais que permitem que o ouvinte possa perceber a diferença entre palavras tais como *medo* e *cedo*. Visto que uma diferenciação pode ser percebida, estas palavras podem representar diferentes conceitos. Os lingüistas têm tentado explicar o mecanismo pelo qual

os seres humanos codificam uma mensagem em sons e decodificam estes em uma mensagem, propondo a existência de unidades intermediárias chamadas *fonemas*. Os fonemas são unidades abstratas com papel distintivo na língua e que atuam supostamente em um domínio psicológico e não físico e portanto não podem ser pronunciados. O papel distintivo do fonema, atuando como a unidade que o sistema fonológico usa para distinguir palavras, é a sua principal função a ser desempenhada.

Apesar de serem consideradas unidades abstratas, os fonemas são definidos em termos de propriedades ou traços, que fazem a mediação entre a descrição lingüística abstrata e a descrição fonética, a qual pode ser caracterizada acusticamente. De certo modo, os fonemas são sons para os quais apenas as características mais importantes foram especificadas, deixando o restante para ser preenchido posteriormente. A fim de distingüi-los dos sons realmente produzidos, os fonemas são usualmente representados entre barras (/ /) enquanto os sons são representados entre colchetes ([ ]). A palavra *dia* por exemplo, é representada pelos fonemas /dia/ e pode ser pronunciada [dʒiə].

Alofones

A produção de voz involve uma seqüência de movimentos de articuladores de maneira tal que uma sucessão de configurações do trato vocal, correspondente à seqüência de fonemas desejados, ocorra no tempo. Os movimentos para sucessivos fonemas sobrepõem-se no tempo de modo que as configurações do trato vocal durante a produção de um fonema são fortemente dependentes das variações devidas a fonemas adjacentes. Este fenômeno de mudanças na articulação e acústica de um fonema devido ao seu contexto fonético é chamado de coarticulação [21].

Uma regra fonológica é então o mecanismo pelo qual as características fonéticas precisas de um fonema são preenchidas de acordo com contexto fonético em que este fonema se encontra. O som realmente produzido (ou fone), gerado pela operação de uma regra é chamado de *alofone* do fonema em questão. Podemos dizer que o alofone é a realização acústica do fonema. Podemos ver o problema de maneira diferente e definir o fonema como uma coleção de alofones. Sob esta interpretação, a função do conjunto de regras fonológicas é selecionar o alofone que é apropriado a um determinado contexto fonético.

Uma propriedade que os alofones correspondentes a um determinado fonema possuem é de apresentar uma distribuição complementar, ou seja, onde um ocorre, o outro não pode ocorrer. No Português por exemplo, nos dialetos em que o fonema

/t/ é realizado como [tʃ] e [t], o segmento [tʃ] ocorre somente antes da vogal [i] como por exemplo na palavra *tia*, enquanto o segmento [t] ocorre somente diante das outras vogais. Esta propriedade implica que os alofones não podem desempenhar função distintiva na língua. Não existe a possibilidade por exemplo de se distinguir duas palavras [t]*ia* e [tʃ]*ia*, porque o sistema fonológico considerado não permite que o segmento [t] ocorra naquele ambiente.

O alofone pode também ocorrer como variação livre de um determinado fonema. Neste caso, uma realização do fonema é escolhida livremente, não ocorrendo apenas em algum contexto específico. Geralmente esta variação é regional como por exemplo o fonema /R/, mais retroflexo no interior de São Paulo.

3.3 FONÉTICA

Fonética articulatória

A fonética articulatória procura descrever os sons da língua relacionando-os com as posições e movimentos dos órgãos do aparelho fonador.

As palavras são tradicionalmente divididas em partes chamadas sílabas. Cada sílaba contém uma vogal, que corresponde ao som mais intenso e para o qual o trato vocal apresenta o maior grau de abertura. As sílabas podem conter consoantes, para as quais o trato vocal encontra-se parcial ou completamente obstruído. Os fonemas vocálicos e consonantais podem ser classificados com relação ao modo e ponto de articulação. O *modo de articulação* refere-se a como o trato vocal restringe o fluxo de ar. O *ponto de articulação* refere-se à localização no trato vocal onde ocorre a constricção mais estreita.

Modo de articulação

O modo de articulação refere-se ao caminho do fluxo de ar e a que grau é obstruído pelas constricções do trato vocal.

Nas *vogais* e *ditongos*, o fluxo de ar pelo trato vocal é direto, não encontrando uma constrição estreita suficiente para causar turbulência.

As consoantes *líquidas* são similares às vogais, mas usam a língua como uma obstrução no centro da passagem, que pode ser rápida ou intermitente, ou com

escapamento lateral. É o caso de /l/e/r/.

Nos sons *nasais* a corrente de ar é obstruída oralmente, e escapa pelas fossas nasais, com o abaixamento do velum. Como exemplos temos /m/ e /n/.

Nas consoantes *oclusivas* (*plosivas*) ocorre uma obstrução total e em seguida a liberação da passagem de ar no trato vocal. Após o fechamento, a pressão atrás da oclusão cresce e é repentinamente liberada, causando uma breve *explosão*. Como oclusivas podemos citar /p/ e /t/.

As consoantes *fricativas* tais como o /f/ ou /s/, são produzidas excitando o trato com um fluxo de ar contínuo, que se torna turbulento na região de constricção do trato vocal.

Tanto as consoantes fricativas como as oclusivas podem ser produzidas com uma emissão simultânea de sonoridade ou não. Neste caso elas serão classificadas como *sonoras* ou *surdas* respectivamente. As fricativas sonoras apresentam duas fontes de excitação: os pulsos glotais periódicos, e ruído na região de constricção. É o caso de /v/ e /z/. Nas oclusivas sonoras, durante o período em que ocorre a oclusão total do trato, há vibração das cordas vocais e uma pequena quantidade de energia é irradiada através das paredes do trato vocal. Isso ocorre na produção de /b/ e /g/, por exemplo.

Ponto de Articulação

Chama-se de ponto de articulação o local onde a obstrução mais estreita ocorre. Os principais pontos de articulação são:os lábios, os dentes, os alvéolos, o palato duro, o palato mole, a úvula, a faringe e a glote. As consoantes para o Português podem ser classificadas em cinco principais zonas de articulação.

As consoantes *bilabiais* são articuladas com a constricção de ambos os lábios como em /p/ e /b/.

Se o lábio inferior se aproxima dos dentes, como em /f/ ou /v/, teremos uma *labiodental*.

Nas consoantes *dentais* a língua toca os dentes incisivos superiores como em /d/ e /t/.

As consoantes *alveolares* são articuladas com a língua tocando os alvéolos

tais como em /l/ e /n/.

Nas consoantes *palatais* o dorso da língua articula-se contra o palato duro como em /nh/.

As consoantes *velares* são articuladas com o dorso da língua aproximando-se do palato mole. Exs.: /k/, /g/.

Vogais

As vogais são geralmente classificadas segundo três principais aspectos articulatórios: o grau de abertura da boca (fechadas ou abertas), a posição dos lábios (arredondados ou não arredondados) e a posição do ponto de constricção máxima (anterior, central ou posterior).

Transcrição fonética

O objetivo da transcrição fonética é representar graficamente os sons existentes em uma língua. Esta representação pode variar com relação ao número de detalhes que se deseja representar. Desse modo podemos ter uma transcrição larga, sem considerar muitos detalhes, ou uma transcrição estreita, procurando levar em conta o maior número possível de detalhes acusticamente perceptíveis.

Uma transcrição fonética é baseada no princípio de que um fone é sempre representado por um símbolo e que este símbolo sempre representa um fone. Dado um conjunto de símbolos para os fones, um foneticista deve ser capaz de transcrever os fones de qualquer língua e comunicar sua pronúncia de maneira inambígua para outros foneticistas familiares com os símbolos.

O Alfabeto Fonético Internacional (AFI) é um conjunto de símbolos criado pela Associação Fonética Internacional que teve maior aceitação. Ele utiliza principalmente letras do alfabeto romano, juntamente com símbolos adicionais criados ou tomados de outras fontes.

Na fig. 3.2 temos um exemplo do Alfabeto Fonético Internacional [22].

		Bilabiais	Labio-dentais	Dentais e alveolares	Retroflexas	Palato-alveolares
CONSOANTES	Plosivas	p b		t d	ʈ ɖ	
	Nasais	m	ɱ	n	ɳ	
	Fricativas laterais			ɬ ɮ		
	Laterais não-fricativas			l	ɭ	
	Vibrantes			r		
	Flapes			ɾ	ɽ	
	Fricativas	ɸ β	f v	θ ð | s z | ɹ	ʂ ʐ	ʃ ʒ
	Contínuas sem fricção e semivogais	w ɥ	ʋ	ɻ		
VOGAIS	Fechadas	(y ʉ u)				
	Semifechadas	(ø o)				
	Semi-abertas	(œ ɔ)				
	Abertas	(ɒ)				

		Alvéolo-palatais	Palatais	Velares	Uvulares	Faríngais	Glotais
CONSOANTES	Plosivas		c ɟ	k g	q ɢ		ʔ
	Nasais		ɲ	ŋ	ɴ		
	Fricativas laterais						
	Laterais não-fricativas		ʎ				
	Vibrantes				ʀ		
	Flapes				ʀ		
	Fricativas	ɕ ʑ	ç j	x ɣ	χ ʁ	ħ ʕ	h ɦ
	Contínuas sem fricção e semivogais		j (ɥ)	(w)	ʁ		
VOGAIS			anteriores	centrais	posteriores		
	Fechadas		i y	ɨ ʉ	ɯ u		
	Semifechadas		e ø		ɤ o		
	Semi-abertas		ɛ œ	ɜ	ʌ ɔ		
	Abertas		æ	ɐ			
			a		ɑ ɒ		

Fig. 3.2 Alfabeto Fonético Internacional (Maia, 1986. [22])

3.4 PROSÓDIA E TRAÇOS PROSÓDICOS

A transcrição fonética procura representar os sons da fala a fim de indicar a pronúncia das palavras em uma determinada língua ou dialeto. Esta transcrição pode ser realizada considerando diferentes níveis de análise. O primeiro nível de análise é o nível segmental. Segundo considerações feitas anteriormente, considerou-se a possibilidade de segmentar a fala em unidades mínimas, os sons distintivos da língua, formando o sistema sonoro primário da língua. Estes sons são chamados de segmentos e este tipo de transcrição que os representa é denominado de transcrição segmental. Porém, é claro que além dos traços acústicos selecionados pelo falante com finalidade distintiva, existem outros traços utilizados para efeitos comunicativos, não indicados em uma transcrição segmental. Estes processos considerados secundários devem ser tratados em um nível acima dos segmentos, chamado de *suprasegmental* [23]. Estes processos incluiriam o acento, ritmo e entonação. Uma palavra pode ser dita mais forte ou mais fraca; ela pode ser dita variando a duração das vogais; ela pode ser dita com um padrão de pitch que começa alto e termina baixo ou com uma variação que começa baixo e termina alto. Estas variações são realizadas sem que a identidade lexical das palavras seja alterada. Tais traços geralmente se estendem por extensões de fala maiores que apenas um som, e por isso são chamados de suprasegmentais. O termo prosódia é usado alternativamente a suprasegmental.

De acordo com as considerações feitas, a prosódia pode ser definida como um conjunto de propriedades cuja relação com as palavras selecionadas é essencialmente variável [24]. Ou seja, enquanto uma palavra ou frase pode ser identificada em termos de sua composição segmental, ela não é individualizada por sua intensidade, duração ou pitch. Uma certa palavra não terá seu sentido denotativo modificado se articulada mais forte, mais longa ou mais alta que em outro contexto. Esta definição considera uma dada língua, pois o mesmo conjunto de propriedades pode ter função distintiva em outras línguas. No Chinês, por exemplo, o tom é distinto, enquanto no Português não.

De acordo com esta definição, o acento nem sempre se inclui inteiramente no domínio da prosódia. Em línguas como o Português, o acento é utilizado para distinguir uma palavra de outra. É o caso por exemplo das palavras sábia, sabia e sabiá. Estas palavras se distinguem primariamente pelo padrão de acento lexical. Neste caso a função do acento não pertence ao domínio da prosódia. Já em línguas como o Francês, o acento marca apenas os finais das palavras, e pode-se dizer

que ele é uma característica prosódica.

Os traços prosódicos podem se estender sobre domínios variáveis: às vezes sobre trechos de fala relativamente curtos tais como uma sílaba ou uma palavra; às vezes sobre trechos relativamente mais longos tais como uma frase, ou uma sentença.

Os aspectos prosódicos são bastante importantes na síntese de voz, pois uma fala sintetizada com uma prosódia inadequada perderá inteligibilidade, além de se tornar desagradável de ser escutada por longos períodos.

3.5 PITCH, DURAÇÃO E INTENSIDADE

A prosódia de fala contínua pode ser analisada e descrita em termos da variação de três principais traços ou parâmetros prosódicos. Estes traços são o pitch, a duração e a intensidade [25,26]. Os termos pitch, duração e intensidade se referem aos traços sob o aspecto perceptual. Estes traços possuem seus correlatos acústicos e fisiológicos.

Pitch

O pitch é o traço prosódico mais centralmente envolvido na entonação. Fisiologicamente, o pitch é primariamente dependente da taxa de vibração das cordas vocais na laringe.

A taxa de vibração das cordas vocais é refletida na medida acústica da freqüência fundamental. Este termo se refere ao número de repetições da forma de onda regular em um segundo, tal forma de onda regular sendo tipicamente produzida quando as cordas vocais vibram para produzir voz. Assim o número de vezes que as cordas vocais fecham e abrem completamente é diretamente relacionado à freqüência de repetição da forma de onda.

Enquanto a freqüência fundamental envolve medidas acústicas em Hz, pitch é usado como um termo perceptual, relacionado aos julgamentos do ouvinte se um som é alto ou baixo, se um som é mais alto ou mais baixo que outro som ou se a voz está subindo ou descendo. Tais julgamentos não são linearmente relacionados à freqüência fundamental. Para um ouvinte perceber um tom como o dobro de outro, a diferença entre os dois tons é muito maior em freqüência absolutas maiores. Por

exemplo 1000 Hz é percebido como o dobro de 400 Hz, mas 4000 Hz é percebido como o dobro de 1000 Hz. Os valores de freqüência fundamental para a fala são relativamente baixos e para muitos propósitos práticos o termo pitch pode ser usado em equivalência a freqüência fundamental.

Duração

A duração é o parâmetro prosódico relacionado ao tempo. A produção de uma determinada unidade lingüística envolve uma sucessão de movimentos articulatórios que exigem que uma determinada quantidade de tempo seja gasta para realizá-los. Isto faz que os segmentos tenham durações intrínsecas mais longas ou menos longas, dependendo dos articuladores envolvidos na produção daquela unidade.

A medida acústica da duração é uma tarefa complicada, pois a natureza contínua da fala não permite que limites bem claros entre os segmentos fonéticos possam ser estabelecidos. Desse modo, é difícil determinar se um dado evento acústico, tal como a transição de um certo parâmetro acústico, será importante para a percepção de uma unidade lingüística, e portanto se deve ou não ser considerado ao efetuar a medida de sua duração.

Intensidade

A intensidade é o parâmetro prosódico relacionado com a energia presente no sinal acústico. A energia do som por sua vez está relacionada com a amplitude do sinal de voz, sendo as variações de amplitude produzidas pelas variações de pressão do ar vinda dos pulmões. Um som produzido com mais força será percebido como de intensidade maior. Porém, a intensidade não está linearmente relacionada com seu correlato acústico. Um som tem que ter uma energia muito maior que o dobro antes que seja percebida como tendo o dobro de intensidade.

A intensidade se relaciona com o julgamento do ouvinte se um som é forte ou fraco, ou se é mais forte ou fraco que um som. Este julgamento não está unicamente relacionado com a amplitude do sinal, sendo afetado também pela duração e freqüência fundamental.

A relevância da intensidade como um traço prosódico, assim como a duração, é difícil de ser avaliada por causa das diferentes influências na intensidade absoluta de uma sílaba ou seqüência de sílabas. Por exemplo, as vogais abertas são acusticamente de maior intensidade que as vogais fechadas. Além disso, a relação entre a intensidade absoluta e a intensidade percebida não é linear.

CAPÍTULO 4

DESENVOLVIMENTO DO DICIONÁRIO DE UNIDADES BÁSICAS

Neste capítulo descreveremos o método de síntese escolhido e os procedimentos utilizados na obtenção das unidades básicas de síntese.

4.1 TAMANHO DA UNIDADE DE CONCATENAÇÃO

Lembrando o esquema do sistema de síntese de voz apresentado na figura 1.1, nós vemos que um dos blocos existentes é o dicionário de segmentos de voz. Uma das principais questões no desenvolvimento de um sistema de conversão texto-voz, é a escolha da unidade básica de voz que será utilizada na composição do dicionário. O método mais simples consistiria em se armazenar palavras e concatená-las para produzir sentenças. Porém, estamos trabalhando com uma aplicação de vocabulário irrestrito e isto inviabiliza a utilização da palavra como unidade básica, devido à capacidade de armazenamento necessária, uma vez que seria preciso gravar um léxico de cerca de 200.000 palavras. Além disso, palavras pronunciadas isoladamente diferem bastante das mesmas quando pronunciadas em diferentes contextos, levando a uma perda de naturalidade e inteligibilidade. Para evitarmos este problema teríamos que gravar uma mesma palavra considerando suas variações em diferentes contextos, agravando ainda mais o problema da capacidade de armazenamento. Portanto, para aplicações de vocabulário irrestrito, esta primeira opção torna-se inviável. Isto leva à necessidade de procurar outras possíveis unidades básicas, que sejam capazes de gerar qualquer tipo de enunciado e ao mesmo tempo sejam em número limitado. As possibilidades de escolha dividem os sistemas em dois grupos principais: os de síntese por regras e síntese por concatenação.

4.1.1 Síntese por Regras

Para os sistemas de síntese de voz a partir de texto, torna-se necessário partir para a escolha de unidades sonoras mais básicas, como por exemplo os fonemas. O número de fonemas é restrito, situando-se em torno de 30 para o idioma Português. Portanto, pode-se pensar em uma possível técnica de síntese, onde uma referência para cada fonema é armazenada, e uma mera concatenação com uma possível suavização nas junções realizada para produzir a saída de voz desejada. Esta suavização serviria para produzir um espectro suave e evitar descontinuidades espectrais. Contudo, estudos perceptuais [27] mostram a importância das transições entre fonemas na percepção dos mesmos, levando a uma forte dependência do contexto fonético onde eles ocorrem, devido aos efeitos de coarticulação. Portanto, para garantir a inteligibilidade dos sons produzidos, é necessário garantir que as transições sejam realizadas o mais precisamente possível, a fim de assegurar uma continuidade espectral ao longo de todo o enunciado.

Para realizar as transições deve ser criado um conjunto de regras obtidas a partir da análise de realizações da fala natural. A função destas regras é indicar a evolução que um conjunto de parâmetros de controle de um sintetizador deve seguir a fim de produzirmos um determinado som. Estas regras são complexas e para obtê-las é necessário realizar estudos extensivos das propriedades espectrais do processo de produção natural da fala. A complexidade está em se determinar as mudanças que são perceptualmente relevantes no domínio acústico.

Este tipo de síntese, em que as transições são controladas por um conjunto de regras, é chamado de síntese por regras.

O sintetizador mais adequado para ser utilizado na síntese por regras é o sintetizador por formantes, pois possui um conjunto de parâmetros de controle diretamente relacionado com os resultados obtidos de análises acústicas. As regras serão elaboradas em função dos parâmetros de síntese disponíveis. Quanto mais completo for o sintetizador, mais complexas e elaboradas deverão as regras, porém maior será o número de detalhes acústicos possíveis de serem considerados pelas regras.

4.1.2 Síntese por Concatenação

A fim de evitar o problema das transições, outras opções foram propostas, tais como a semi-sílaba e o difone [28]. A sílaba consiste de um núcleo (tanto uma vogal ou um ditongo) e uma consoante vizinha. As semi-sílabas são unidades de voz obtidas dividindo-se as sílabas ao meio com o corte efetuado durante a vogal, onde os efeitos de coarticulação são mínimos. O difone é formado por dois fones adjacentes, limitado pela região estável dos fones e compreendendo a transição completa entre eles. Quando semi-sílabas ou difones são concatenados na seqüência adequada, uma fala contínua é geralmente obtida, pois os sons integrados nas junções são espectralmente similares.

A suavização de parâmetros espectrais nas junções entre unidades se torna mais crítica à medida que o tamanho da unidade decresce, pois o número de junções que ocorrem no tempo é maior. Portanto, as regras de suavização para difones e semi-sílabas são relativamente simples, pois as regras de transições já estão implicitamente contidas nos mesmos. Já para unidades tais como fonemas, torna-se necessário obter um conjunto de regras que represente a coarticulação no trato vocal e a partir das quais é possível sintetizar o sinal acústico desejado.

Na síntese por concatenação, os sintetizadores que utilizam o método de análise/ressíntese apresentam-se como a opção mais indicada. Isto porque uma vez que estaremos utilizando unidades que incorporam as transições, uma boa estratégia a ser adotada é obtê-las diretamente a partir da análise e armazenamento de porções de fala natural.

4.2 ESCOLHA DA CONFIGURAÇÃO

A primeira decisão a ser tomada na fase inicial do desenvolvimento do sistema de síntese estava relacionada com a escolha do método de síntese a ser utilizado. A escolha residia basicamente entre a síntese por regras e a síntese por concatenação. Para realizar tal escolha é necessário levar em consideração critérios tais como memória disponível, complexidade, qualidade exigida e flexibilidade do sistema.

A síntese por regras utiliza pouca memória para armazenamento dos parâme-

tros pois tem uma representação econômica dos segmentos mínimos. Basta armazenar um conjunto de parâmetros de referência para cada segmento considerado, juntamente com as regras para efetuar as transições entre os diferentes segmentos. Isto por sua vez implica na complexidade do sistema para a obtenção destas regras, que devem ser extraídas a partir de uma análise da fala natural. Porém este controle sobre a realização das transições permite que uma boa qualidade, próxima da natural, seja obtida. Além disso, a síntese por regras permite que o sistema seja mais flexível no caso em que se deseje obter diferentes tipos e qualidades de voz. Isto pode ser feito alterando as regras e os valores dos parâmetros de referência a fim de adaptá-los para o tipo de voz desejado. Esta flexibilidade de variação pode ser importante para aplicações onde um tipo de voz específico, uma voz feminina por exemplo, é mais adequada.

A síntese por concatenação tem como principal atrativo a simplicidade para obtenção de parâmetros dos segmentos mínimos, os quais são obtidos diretamente a partir da gravação de porções de fala natural. Isto faz com que a quantidade de memória exigida seja grande, uma vez que o número de elementos no sistema de síntese por concatenação é elevado. A qualidade tende a ser inferior em relação à síntese por regras, pois por mais que se tente realizar uma concatenação suave nas junções, ocorrem descontinuidades em algumas partes. Estas descontinuidades ocorrem devido a variações encontradas nas diferentes realizações dos fonemas quando da gravação de fala natural. Este tipo de descontinuidade pode ser evitado na síntese por regras pois as transições podem ser controladas de tal modo que possam ser realizadas de maneira suave.

Apesar das vantagens oferecidas pelo sistema de síntese por regras, resolvemos optar pela escolha da síntese por concatenação, pela sua menor complexidade. Por se tratar de um estudo inicial, tentamos evitar a concentração em apenas uma parte do sistema, visando a implementação de um sistema completo, para num trabalho posterior tentar melhorar a qualidade de cada um de seus blocos constituintes. Como unidade de concatenação utilizamos os difones. Os difones foram extraídos de palavras gravadas isoladamente, digitalizadas, e analisadas pelo algoritmo LPC. O método LPC foi escolhido por poder tornar o processo de criação do inventário de difones rápido e eficiente.

A seguir na tabela 4.1 temos os valores dos parâmetros utilizados na digitalização da voz e análise LPC.

A filtragem na entrada do conversor A/D é feita em 3.4 kHz visando aplicações em telefonia. Isto implica porém em uma degradação de determinados sons,

tais como os fricativos, para os quais boa parte da energia espectral se encontra além da faixa de 4 kHz. Esta degradação deve ser levada em consideração na avaliação da qualidade da voz sintetizada.

Tabela 4.1

Parâmetro	Valor
Freqüência de corte do filtro de entrada do A/D	3.4 kHz
Número de bits do A/D	12
Freqüência de amostragem	8.0 kHz
Ordem do preditor	8
Quadro de análise	10 ms
Janela de Hamming	15 ms
Método de análise	Autocorrelação

Existem várias aplicações que não utilizam a rede telefônica, e para as quais é possível conseguir uma melhora de qualidade adicional. Neste caso o sistema deve ser desenvolvido permitindo uma faixa de passagem mais larga e aumentando-se a freqüência de amostragem, possibilitando deste modo uma melhor reprodução de alguns sons.

O tamanho do quadro escolhido foi de 10 ms. Apesar do valor utilizado em aplicações de codificação de voz ser usualmente de 20 ms, adotamos o valor de 10 ms pelo fato de estarmos trabalhando com segmentos curtos de voz. Verificamos que para difones compostos por segmentos e transições rápidas tais como o flap [r], por exemplo, um quadro de duração menor permitia uma melhor escolha do ponto de segmentação do difone.

O algoritmo utilizado para o cálculo dos coeficientes LPC é o algoritmo de Levinson-Durbin. A ordem do preditor foi fixada em 8 pois se mostrou adequada em testes anteriores feitos no desenvolvimento de vocoders.

A detecção do período de pitch é feita utilizando o algoritmo de Kurt Schäffer-Vincent [29]. No nosso caso a detecção do período de pitch é feita apenas para nos indicar se o quadro é sonoro ou não sonoro. Na síntese deve ser

utilizado um outro valor determinado a partir de regras para o controle da entonação. Na versão atual do conversor é utilizado um valor fixo, pelo fato de ainda não dispormos destas regras.

4.3 CONJUNTO DE ALOFONES CONSIDERADOS

O sistema de síntese a partir de texto deve ser capaz de produzir mensagens com um vocabulário irrestrito. Para que isso seja possível, é necessário que tenhamos armazenado um conjunto de segmentos mínimos a partir dos quais qualquer enunciado possa ser produzido. Portanto, a primeira etapa na construção do vocabulário de difones é a escolha de um conjunto representativo básico de segmentos fonéticos. Isto naturalmente deve ser feito tendo em vista um dialeto específico.

Inicialmente temos que considerar quais os fonemas e alofones existentes na Língua Portuguesa. Se fizermos uma transcrição fonética estreita do Português, o número de alofones pode se tornar grande. Como a proposta do trabalho é realizar um estudo inicial, resolvemos adotar um conjunto mais simplificado. O fato de considerarmos um número maior de alofones adicionaria pouco à inteligibilidade do sistema nesta fase, servindo apenas para reproduzir alguns aspectos mais sutis da língua. Portanto trabalharemos com o seguinte conjunto de fones:

Consoantes					
plosiva	fricativa	nasal	lateral	flap	vibrante
p	f	m	l	ɾ	R
t	s	n	ʎ		
k	ʃ	ɲ			
b	v				
d	z				
g	ʒ				

Vogais	
orais	nasais
a	ã
e	ẽ
ɛ	
i	ĩ
ɪ	
o	õ
ɔ	
u	ũ
ʊ	

Fig. 4.1. Conjunto de alofones considerados

Neste caso foram consideradas as vogais com suas variações átonas, tônicas e nasalizadas, as consoantes e os fonemas /ʎ/ e /ɲ/. Os difones considerados foram obtidos a partir das combinações CV, VC, VV e CC, onde C representa qualquer consoante e V qualquer vogal. Os fonemas /ʎ/ e /ɲ/ foram incluídos entre as consoantes. Destas combinações apenas CC não foi considerado na sua totalidade, eliminando-se as combinações que não ocorrem para o Português. Algumas combinações que não existem dentro de palavras mas que ocorrem entre palavras tiveram que ser consideradas. O difone / ẽĩ / por exemplo, não existe nas palavras mas ocorre por exemplo na combinação *maçã inteira*.

A seguir descreveremos a metodologia utilizada na obtenção dos difones.

4.4. DESENVOLVIMENTO DO VOCABULÁRIO DE DIFONES

Uma das etapas principais no desenvolvimento do sistema de síntese é a construção do dicionário de unidades básicas a partir das quais será efetuada a síntese. A obtenção de um sistema de boa qualidade dependerá da capacidade de se realizar esta etapa com êxito.

No sistema desenvolvido a unidade básica utilizada foi o difone. Como foi definido, o difone é o segmento de voz que inicia no centro da região estável de um fone e termina no centro da região estável do próximo fone, contendo a transição completa entre os dois fones.

Uma das principais vantagens da utilização do difone como elemento básico, é o fato dele poder ser extraído e armazenado diretamente a partir do sinal de voz natural. Porém, a exigência de vocabulário irrestrito faz com que durante o processo de concatenação os segmentos de voz armazenados sejam recombinados em uma grande variedade de seqüências diferentes, de acordo com o texto de entrada. A junção dos segmentos em uma nova seqüência diferente da original, pode dar origem a descontinuidades espectrais. Isto faz com que o processo de obtenção de difones tenha que ser realizado com cuidado a fim de tentar minimizar este problema.

Para obtenção dos difones utilizamos palavras gravadas isoladamente. Para cada difone é necessário gravar uma palavra contendo o difone desejado. Devido à dificuldade de encontrar palavras que atendessem certos requisitos, criamos pa-

lavras sem sentido (logátomos), de estrutura regular. As palavras devem ser escolhidas de modo a evitar coarticulação sobre o difone em questão. Também devemos estar atentos para o fato de que o difone deve compreender em seus limites porções espectralmente estáveis. Para isso, procuramos manter uma certa consistência, criando palavras que mantivessem uma estrutura regular de modo a atender certos requisitos [30].

Um dos requisitos foi procurar palavras de pelo menos três sílabas de modo que o difone se localizasse na parte central da palavra. Isto é necessário pois quando a palavra é enunciada, possui maiores variações em sua extremidades, atingindo uma estabilidade em sua porção central.

Para facilitar o processo de segmentação e evitar problemas de coarticulação, a plosiva [p] e a vogal [a] foram utilizadas na formação da estrutura das palavras. A plosiva [p] foi escolhida por possuir um tempo de silêncio, o que facilita a localização das vogais e consoantes. Além disso, é uma consoante que apresenta pouco movimento dos articuladores, o que é desejável a fim de evitar coarticulação sobre o difone desejado.

Para obtenção de difones do tipo VC e VV, as palavras são iniciadas com vogal. Para difones do tipo CV e CC as palavras são iniciadas em consoante. Como exemplos destes casos temos os difones **[ea]** e **[ep]** que são isolados a partir de uma gravação das palavras *apeapa* e *apepa* respectivamente, e os difones **[ta]** e **[tr]** isolados a partir das palavras *patapa* e *patrapa*.

As palavras foram digitalizadas diretamente utilizando o SAPDV-A [31]. No momento da gravação dois cuidados foram observados a fim de não alterar a qualidade entre as gravações. Para isso, tentou-se manter a velocidade e a amplitude constantes. Isso assegura que as realizações de um mesmo segmento sejam consistentes e garantam uma uniformidade no momento da concatenação.

Após efetuadas as gravações, realizamos a análise LPC de todas as palavras, guardando os oito coeficientes, o ganho, e o período de pitch. Estes parâmetros servirão de auxílio na segmentação dos difones.

4.4.1 Obtenção dos Difones

Para isolarmos corretamente os difones a partir das palavras, é necessário que tenhamos parâmetros que nos auxiliem em dois aspectos. Primeiramente devemos

ter condições de determinar os limites entre os quais se situa o difone desejado. Em segundo lugar devemos escolher o melhor ponto onde efetuar a segmentação, levando em conta que este ponto deve se mostrar espectralmente estável.

Isto nos levou a utilizar os parâmetros LPC para auxiliar na segmentação dos difones. O ganho e o período de pitch ajudam a identificar as partes sonoras e não sonoras na palavra, e portanto determinar a região onde o difone desejado se encontra. Por meio do primeiro coeficiente LPC, podemos determinar onde o espectro se mantém estável. Isto ocorre porque sabemos que os coeficientes de predição especificam a função de transferência do filtro de síntese e, indiretamente, representam as propriedades espectrais do sinal. Em uma região onde os coeficientes se mantêm constantes podemos afirmar que o espectro não varia.

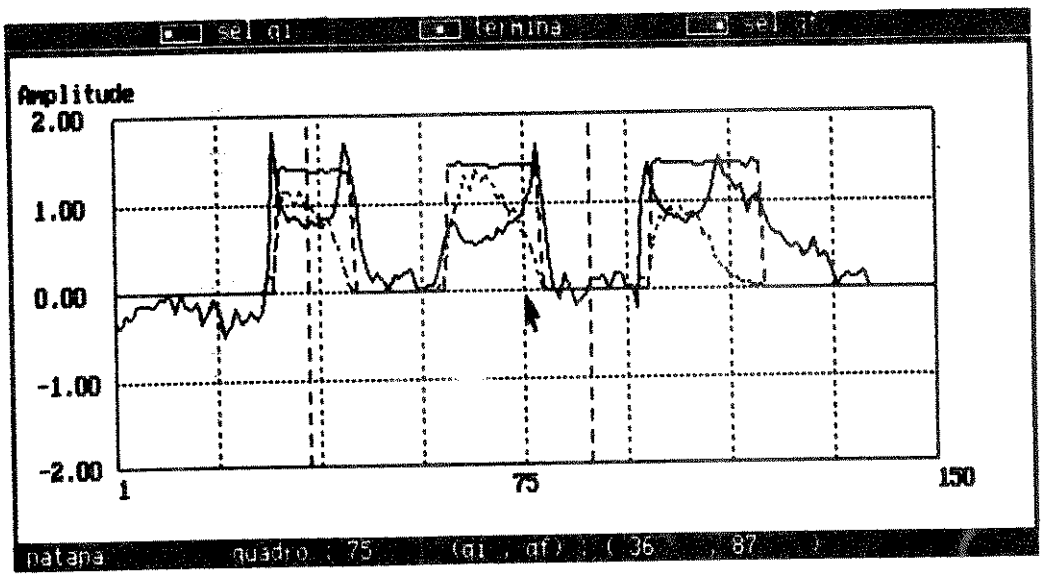
A decisão do ponto de segmentação é feita de posse da visualização gráfica dos parâmetros acima mencionados. No caso, utilizamos apenas o primeiro coeficiente para facilitar a visualização. Para cada difone desejado, escolhe-se a palavra correspondente e mostra-se a evolução dos parâmetros na forma gráfica na tela do computador. Primeiramente fazemos uma ampliação da região onde se localiza o difone. Em seguida escolhe-se um ponto pertencente à região limite do difone, determinada com auxílio da visualização do ganho e do período de pitch. Este ponto não deve apresentar variação espectral e portanto deve pertencer a uma região onde o primeiro coeficiente não varia. Deve-se procurar manter a duração dos difones o mais constante possível, a fim de manter uma uniformidade. O controle individual da duração segmental deve ser realizado posteriormente no momento da síntese.

Uma vez realizada a separação do difone, faz-se um teste, ressintetizando o difone em combinação com outros difones já testados e separados corretamente, e aceitando ou eventualmente refazendo a segmentação. O processo é repetido para cada difone do vocabulário e armazenado na forma de arquivo.

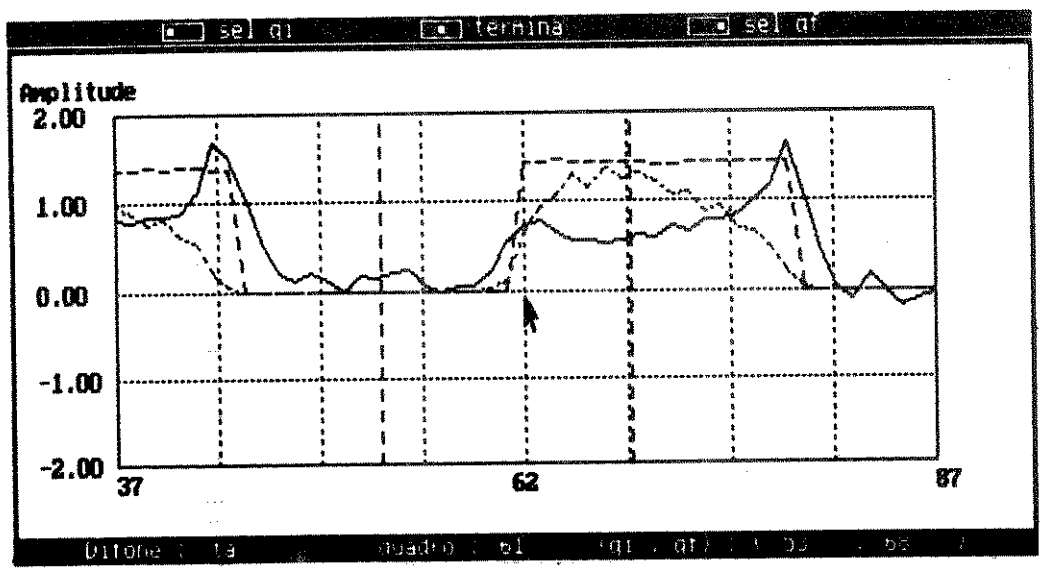
A representação de cada difone será composto pelo tamanho do difone em número de quadros de 10 ms, os coeficientes LPC de cada quadro, o ganho e o período de pitch. O ganho e os oito coeficientes são guardados no formato de ponto flutuante de quatro bytes. O período de pitch é guardado como inteiro de dois bytes. Portanto cada quadro ocupa um espaço de 38 bytes. Considerando uma duração média de 15 quadros, teremos para cada difone um espaço médio ocupado de 570 bytes.

Para ilustrar o processo descrito, daremos um exemplo da obtenção do difone

[ta]. Para isolar este difone temos a palavra *patapa*. Os parâmetros relativos a esta palavra são mostrados na figura 4.2 (a). Na figura 4.2 (b) temos ampliada a região relativa ao difone [ta].



(a)



(b)

Fig. 4.2. (a) Gráfico dos parâmetros correspondentes à palavra patapa
(b) Região ampliada da fig. (a) relativa ao difone ta.

A fig. 4.2 mostra a apresentação da tela do computador na utilização do programa desenvolvido para extração dos difones. O botão esquerdo do mouse permite selecionar o quadro inicial, o botão direito o quadro final e o botão central é utilizado para terminar. Na parte de baixo da tela estão indicados a posição do mouse e os quadros inicial e final selecionados. A escala vertical indicada é válida apenas para o coeficiente LPC. A escala horizontal corresponde aos quadros de análise LPC, sendo que segundo a configuração adotada, cada quadro equivale a um intervalo de 10 ms.

4.4.2 Utilização de Trifones

O uso de trifones é uma maneira de tentar superar a deficiência da suposição inicial dos sistemas de difones, que consideram que a coarticulação só ocorre entre fones adjacentes. A única maneira de levar em conta a coarticulação nos sistemas de difones é a utilização dos trifones ou até mesmo seqüências maiores. Estas exigências crescem à medida que se tenta reproduzir mais detalhes fonéticos da fala natural. Deve-se estar atento para o fato de que o aumento no número de elementos representa um aumento no requisito de memória. Caso o volume de memória requerido seja muito grande, a escolha tende a ser favorável a um sistema que implemente os efeitos de coarticulação por regras, ou seja um sistema de síntese por regras.

No nosso conjunto de unidades básicas, fizemos uso de alguns trifones a fim de superar dificuldades que tivemos na produção de alguns sons. Devido a limitações do próprio método de análise LPC, alguns sons são modelados de maneira pouco eficiente. Isto ocorre por exemplo com difones formados por uma obstruinte (oclusiva ou fricativa) seguida do fone [r]. A dificuldade é que temos aqui uma transição muito rápida, sendo que uma região de grande variação pode ficar compreendida dentro de um quadro de análise, pois a duração da transição é menor que a duração do quadro. Neste caso, a junção se torna bastante problemática, pois a degradação devido ao método de concatenação torna o problema mais evidente. Tendo em vista este problema, resolvemos adotar neste caso o uso de trifones, ou seja, considerar a seqüência obstruinte-[r]-vogal como sendo uma unidade só. Isto evita a necessidade da concatenação de seqüências do tipo [obstruinte][r]-[r][vogal]. A palavra *Pedro* por exemplo, ao invés de ter a seguinte seqüência de difones /pe-ed-dr-rω/, será substituída por /pe-ed-drω/. Os trifones representam um acréscimo no armazenamento, porém o ganho na qualidade obtida justifica este acréscimo.

Considerando os difones e trifones necessários, chegamos a um total de cerca de 1000 unidades para o nosso dicionário.

4.5 SOFTWARE DESENVOLVIDO PARA A GERAÇÃO DO DICIONÁRIO DE DIFONES

A grande quantidade de difones a serem isolados, levou à necessidade de se criar um programa para um microcomputador do tipo PC, a fim de tornar o processo para a geração do dicionário de difones o mais automático possível, deixando apenas a decisão do ponto de segmentação por conta do operador. A idéia foi criar um programa interativo de modo a facilitar o processo de segmentação, fornecendo facilidades para possíveis correções. Basicamente o programa fornece facilidades para as seguintes tarefas dentro de um ambiente integrado:

- Gravação de um determinado número de arquivos de voz digitalizados. Pode-se fornecer uma seqüência de nomes de arquivo e em seguida gravá-los seqüencialmente.
- Análise LPC de um determinado número de arquivos de voz. Os arquivos de saída recebem automaticamente a extensão [lpc].
- Apresentação gráfica na tela dos parâmetros LPC relativos a uma certa palavra gravada e analisada. Simplesmente especificamos a palavra a partir da qual será retirado o difone e os parâmetros relativos a esta palavra são apresentados de forma gráfica na tela. Inicialmente os parâmetros dos quadros LPC relativos a toda a palavra são apresentados na tela e espera-se o próximo comando. Através do mouse marca-se a região onde se encontra o difone para ser ampliada, a fim de se aumentar a precisão na próxima etapa que será a segmentação para isolamento do difone.
- Segmentação do difone a partir dos gráficos dos parâmetros da palavra correspondente na tela, assinalando-se na tela com o mouse os pontos onde deve ser efetuada a segmentação. Os parâmetros LPC compreendidos entre os quadros escolhidos são guardados na forma de arquivo, juntamente com o comprimento em quadros do difone.
- Reprodução e concatenação a partir de parâmetros LPC. Por meio desta opção podemos verificar se o ponto de segmentação foi satisfatório.

Além destas tarefas, dentro do programa é possível dispor das seguintes opções de análise de um arquivo de voz:

- Detecção do período de pitch do arquivo de voz.
- Extração da curva de energia do arquivo de voz.
- Cálculo do espectro de potência e da envoltória de uma parte selecionada do sinal de voz.
- Seleção por meio de dois cursores de uma região do sinal de voz. Uma vez feita a seleção, esta região pode ser ampliada ou reproduzida sob forma sonora, sendo também possível a apresentação na tela dos gráficos de período de pitch ou energia, relativos à região selecionada

Uma outra opção que o programa oferece, permite que uma curva de variação desenhada com um mouse na tela seja lida e transformada em arquivo. Esta opção foi implementada visando o estudo de variação da freqüência fundamental e seus efeitos perceptuais, pois permite o desenho de um certo contorno de variação de pitch na tela e sua utilização posterior, na síntese de uma frase com aquele contorno.

CAPÍTULO 5

ANÁLISE DO TEXTO

Neste capítulo descreveremos o processamento realizado sobre o texto de entrada a fim de extrair informações que serão utilizadas nas etapas posteriores de geração de voz.

5.1 Introdução

A primeira etapa para a conversão do texto em voz é a realização da conversão ortográfica-fonética. Para auxiliar nesta etapa, utilizamos o programa LEX [32]. Este programa gera um analisador lexical em linguagem C. Este programa é compilado e então usado para realizar a conversão. As regras são escritas em uma notação bastante simples. O analisador percorre um texto e, ao encontrar um determinado padrão, realiza uma determinada tarefa, composta por comandos escritos em linguagem C. No nosso caso estes comandos são utilizados para realizar a conversão ortográfica-fonética, escolhendo o segmento fonético adequado, e para determinar a sílaba tônica. O conversor é utilizado também para processar caracteres e símbolos especiais tais como algarismos, pontuação, abreviaturas, etc. Uma vez obtida a transcrição fonética, é realizada a separação das sílabas de cada palavra, utilizando um algoritmo separado.

O programa LEX se mostrou adequado para as necessidades de nossa aplicação. A principal vantagem oferecida foi a simplicidade de sua notação, facilitando o processo de escrita e alteração das regras. Uma listagem das regras escritas para o programa LEX é fornecida no Apêndice.

A seguir descreveremos quais os critérios e metodologia utilizados para efetuar cada uma das etapas da análise do texto.

5.2 PRÉ-PROCESSAMENTO DO TEXTO

Um texto genérico pode conter os mais diversos símbolos e caracteres não-alfabéticos. Antes de realizar qualquer análise ou transformação do texto de entrada, é preciso realizar um pré-processamento a fim de converter estes símbolos e caracteres em uma forma possível de ser processada em etapas posteriores. Naturalmente, nem sempre é possível realizar a transformação correta com relação a alguns símbolos, pois sua utilização pode ter diferentes interpretações, dependendo do meio em que o texto se encontra (jornal, livro, etc) e a área de conhecimento a que o texto se refere. Portanto, os símbolos considerados e as transformações efetuadas seguem a convenção usualmente utilizada em textos comuns, podendo ser adaptados e ampliados para casos específicos.

Os símbolos considerados no processo de conversão são os seguintes:

a) Algarismos:

Os números compostos pelos algarismos de 0 a 9 são convertidos na sua forma extensa. A faixa de valores prevista vai de 0 a 999.999.999, sendo facilmente estendida para valores maiores. Isto não constitui uma limitação uma vez que, para valores muito grandes, nos textos em geral, é usada a forma escrita por extenso. O número 300.000.000, por exemplo, é escrito 300 milhões. No caso de uma seqüência de algarismos seguida de vírgula e uma nova seqüência de algarismos, as duas seqüências são convertidas nos números correspondentes e separadas pela palavra *vírgula.* A seqüência 20,52, por exemplo, será transformada em *vinte vírgula cinqüenta e dois.*

Uma limitação apresentada na conversão é a determinação nos números terminados pelos algarismos *1* ou *2*, se a forma *um* ou *uma,* e *dois* ou *duas* será utilizada. Neste caso, o gênero do objeto ao qual o número se refere, necessário para realizar a escolha correta, não é facilmente determinado. As formas *um* e *dois* serão sempre utilizadas.

b) Abreviaturas

As abreviaturas que podemos encontrar nos textos dependem da aplicação a que se refere este texto. Procuramos considerar apenas aquelas que ocorrem com mais freqüência nos textos de uma maneira geral.

As abreviaturas inicialmente consideradas são:

av.	avenida
cm	centímetro(s)
cr$	cruzeiros
Dr(a).	doutor(a).
Ex.	exemplo
etc.	etcétera
h	hora(s)
Jr.	júnior
kg	quilo(s)
km	quilômetro(s)
m	metro(s)
mm	milímetro(s)
prof.	professor
r.	rua
s	segundo(s)
Sr(a).	senhor(a)

No caso da abreviatura de horas, esta será considerada quando ocorrer uma seqüência de algarismos seguido da letra *h*. Caso ocorra uma seqüência posterior de algarismos indicando os minutos, apenas os valores das horas e minutos separados pela letra *e* serão pronunciados. Se tivermos 21h, por exemplo, esta seqüência será transformada em *vinte e uma horas*. Porém, no caso de 21h30, teremos como resultado *vinte e uma e trinta*.

c) Siglas

São consideradas siglas as seqüências de letras maiúsculas delimitadas por espaço. Uma vez detectada uma sigla, as letras que a compõem são soletradas uma a uma. Ex: PCM terá como saída "pê cê eme".

d) Sinais de pontuação

Os sinais de pontuação considerados são: a vírgula (,), ponto e vírgula (;), ponto (.), ponto de exclamação (!), ponto de interrogação(?), parênteses (()), aspas ("), travessão (-).

Estes sinais são importantes marcas no texto, que servirão mais tarde para auxiliar na incorporação de regras de variação prosódica. Eles servem para indicar mudanças no contorno de entonação e inserção de pausas.

e) Símbolos especiais

O sinal de porcentagem (%) é substituído pela palavra *porcento*. O símbolo de adição (+) é substituído pela palavra *mais*. O sinal de menos (-), quando precede uma seqüência de algarismos, é substituído pela palavra *menos*.

5.3. CONVERSÃO ORTOGRÁFICA-FONÉTICA

Para realizar a conversão ortográfica-fonética, devemos considerar inicialmente qual o conjunto de segmentos fonéticos que será utilizado na geração da voz. Como para a Língua Portuguesa o número de alofones para se obter uma voz de qualidade aceitável é pequeno, a tarefa de transcrição fonética fica razoavelmente simplificada, sendo realizada apenas uma transcrição larga.

Os segmentos considerados no sistema de síntese já foram apresentados na fig. 4.1 e são mostrados na fig. 5.1 novamente, junto com os símbolos convencionados para representá-los.

Esta representação dos segmentos por caracteres foi adotada visando facilitar a manipulação da representação fonética do texto ao longo do processamento no computador, em tarefas tais como comparação de strings, nomeação de arquivo, etc.

A convenção da representação do acento ortográfico para o texto de entrada foi adotada da seguinte maneira:

O **acento agudo** é representado pela letra seguida de aspas simples.
Ex. café = cafe'

O **cê cedilha** é obtido digitando-se 'c' seguido de vírgula.
Ex.: caçador = cac,ador

O **trema** é representado pela letra seguida de aspas duplas.
Ex.: lingüiça = lingu"ic,a

O **acento circunflexo** é obtido digitando a letra seguida do acento circunflexo.
Ex.: ciência = cie^ncia

Repres. Fonét.	Representação (2 caracteres)
a	A_
ɐ	A
ɐ̃	AN
e	E
ɛ	EH
ɐ	EN
i	I
ɪ	Y
ĩ	IN
o	O
ɔ	OH
õ	ON
u	U
ω	W
ũ	UN

Repres. Fonét.	Representação (2 caracteres)
p	P
t	T
k	K
b	B
d	D
g	G
ɾ	R
ʀ	RR
l	L
ʎ	LH
m	M
n	N
ɲ	NH
f	F
v	V
ʒ	J
ʃ	X
s	S
z	Z

Fig. 5.1 Conjunto de alofones com a representação utilizando dois caracteres

Esta convenção de acentuação foi utilizada para estabelecer um padrão ao qual os diferentes modos de representação de caracteres acentuados devem ser convertidos a fim de poderem ser processados pelo módulo de conversão ortográfica-fonética. Desse modo, caso o texto de entrada seja um arquivo gerado utilizando algum editor ou processador de texto específico, deve-se efetuar uma conversão inicial do texto a fim de adequá-lo à convenção de representação adotada.

As regras de conversão são basedas apenas em considerações lexicais, examinando o contexto próximo ao conjunto de caracteres a ser analisado.

Uma vez detectado um certo padrão, o segmento correspondente é escolhido e armazenado em uma variável, juntamente com a informação se é vogal e, caso seja vogal, se é tônica ou não.

As vogais possuem variações átonas e tônicas e portanto parte do problema da transcrição consiste em se determinar se a vogal pertence à sílaba tônica ou não. A princípio todas as vogais são consideradas em suas versões átonas. Após a determinação da sílaba tônica a vogal correta será escolhida. Alguns dos critérios para determinar a sílaba tônica serão fornecidos no próximo ítem.

As maiores dificuldades de conversão se apresentaram nos seguintes casos:

a) **Determinar se as letras *e* e *o* não marcadas por diacrítico (acento ortográfico) corresponderão a vogais abertas ou fechadas.** Esta dificuldade ocorre porque para estes casos, o contexto a nível lexical não é suficiente para prever a ocorrência de um caso ou de outro.

Consideremos por exemplo a palavra bolo, cuja vogal o neste caso é fechada, e a palavra bola, onde a vogal o é aberta. Para este tipo de situação, onde a pronúncia não é determinada pelo contexto, não conseguimos determinar uma regra para prever a ocorrência de uma ou de outra realização. Para resolver esta indeterminação seria necessário a criação de um dicionário de exceções, o qual conteria a transcrição fonética de um conjunto de palavras para as quais o conjunto de regras não atua corretamente.

Existe um outro caso em que palavras com ortografias idênticas se distinguem foneticamente pela vogal aberta ou fechada dependendo de suas funções gramaticais. Temos por exemplo a palavra piloto (substantivo) onde a vogal o é fechada, e a palavra piloto (verbo) onde a vogal o é aberta. Neste caso, a fim de determinar a forma correta a ser utilizada, seria necessário conhecer em qual função gramatical a palavra está sendo utilizada, o que só seria possível pela realização de uma análise sintática.

Uma outra situação que exige um nível de análise mais profundo, refere-se ás palavras derivadas por sufixação, cujos sufixos apresentam uma pronúncia regular. Nestes casos, a transcrição fonética de palavras tais como *cuidadoso* (/kuɪ dadozω/) e *cuidadosa* (/kuɪ dadɔza/) pode ser feita corretamente, caso seja possível identificar as terminações *oso* e *osa* como sendo sufixos formadores de palavras e que portanto posuem uma pronúncia determinada. Porém, esta identificação exige a utilização de um analizador morfológico. Este analizador seria capaz de decompor a palavra *cuidadoso* em um radical *cuidado* e um sufixo *oso,* possibilitando uma escolha da pronúncia correta.

b) **Determinar o fone associado à letra x.** A dificuldade neste caso é que a

associação fonética a esta letra ocorre de maneira não-sistemática. Temos por exemplo na palavra *próximo* o segmento [s] associado à letra x, na palavra *lixo* o segmento [x] e na palavra *fixo* os segmentos [k] e [s].

Diante destas dificuldades podemos tentar formular regras que, mesmo não atuando corretamente em todas as situações, procurem abranger o maior número de casos possível. Neste sentido o processo de elaboração das regras será mais eficiente na medida em que pudermos ter acesso a medidas estatísticas de associação ortográfica-fonética. Criamos algumas regras para a letra x, mas não pudemos utilizar dados estatísticos mais completos por não termos um léxico acessível em computador, de onde seria possível tentar extrair alguma regularidade. Já para as letras *e* e *o* procuramos deixar a tentativa de obtenção de regras para uma etapa mais posterior. Com isso, a utilização da vogal aberta ou fechada deve ser indicada no texto de entrada, utilizando a letra maiúscula quando a vogal é aberta, ou *eh* ou *oh* em começo da palavra, pois a letra maiúscula no começo da palavra pode indicar começo de frase ou nome próprio. Como exemplo temos: *piloto* (substantivo) e *pilOto* (verbo), *este* ou *ehsta*.

5.4 DETERMINAÇÃO DA SÍLABA TÔNICA

A sílaba desempenha um papel importante no estudo da prosódia. Portanto, a fim de decompor as palavras em um conjunto de sílabas, para auxiliar posteriormente na implementação de regras nos módulos de processamento prosódico, um procedimento para a separação de sílabas e determinação da sílaba tônica é utilizado.

A posição da sílaba tônica é uma informação importante a ser considerada quando da formulação de modelos que controlem a variação dos parâmetros prosódicos a fim de imprimir à fala sintetizada uma maior naturalidade. A variação dos parâmetros prosódicos da fala tais como duração segmental, freqüência fundamental e amplitude são fortemente dependentes da posição da sílaba tônica. Portanto, a sua determinação fornecerá importante informação a ser utilizada em etapas posteriores, na incorporação de prosódia, principalmente nos aspectos referentes a ritmo e entonação.

A determinação da sílaba tônica torna-se mais simples caso a separação das

sílabas já tenha sido efetuada. Porém, para efetuar a separação, precisamos utilizar informação relativa à posição da sílaba tônica. Para resolver este impasse, tivemos que fazer algumas considerações a fim de criar regras de acentuação que, embora não fossem eficientes para todos os casos, pudessem ser o mais abrangente possível. No nosso caso, a sílaba tônica é determinada inicialmente, e em seguida esta informação é utilizada para auxiliar a separação das sílabas.

Sabemos inicialmente que toda sílaba possui um núcleo vocálico, ou seja, toda sílaba deve conter pelo menos uma vogal. Portanto, a análise pode ser feita considerando que a posição da sílaba corresponde à posição da vogal. Uma sílaba porém, pode conter mais de uma vogal, e o problema ocorre com os encontros vocálicos em se determinar se duas vogais adjacentes pertencem ou não à mesma sílaba.

O procedimento adotado para tentar contornar este problema é descrito a seguir. Para o Português, estatisticamente o acento lexical tende a recair sobre a penúltima sílaba, ou seja, a maior parte das palavras são paroxítonas. Adotamos então o procedimento de considerar a princípio que todas as palavras são paroxítonas e efetuar a correção quando essa suposição inicial não for verdadeira. Desenvolvemos algumas regras que efetuam esta correção. Estas regras, mesmo não sendo verdadeiras em todos casos procuram abranger o maior número de casos possível. Portanto, as palavras são consideradas paroxítonas a não ser que se enquadrem em alguma das regras apresentadas a seguir.

Regra 1: As palavras marcadas pelos diacríticos (acento ortográfico) já possuem informação sobre a posição da sílaba tônica. Esta regra prevalece sobre todas as outras.

Regra 2: As palavras terminadas em *ar*, *er*, *ir* e *or* levam o acento nesta última vogal. A principal motivação para a formulação desta regra é o fato das formas infinitivas dos verbos apresentarem este padrão de acentuação.

Regra 3: As vogais *e* e *o* abertas são tônicas.

Regra 4: As palavras terminadas em *im* e *um* são oxítonas.

Regra 5: As palavras terminadas em vogal seguidas de *z* levam acento nesta última vogal. Exs. *matriz* (/matr'is/), *xadrez* (/xadr'es/).

Regra 6: Quando a penúltima e antepenúltima vogal são adjacentes, a antepenúltima vogal é tônica caso ocorram as seguintes condições:

a) A penúltima vogal é *y* ou *w*.

b) A antepenúltima vogal é *a*, *e*, *o* ou *w*.

Exs. *peito* (/p'eytw/), *saldo* (/s'awdw/).

Caso estas condições não sejam satisfeitas, a penúltima vogal será acentuada. Exs. *piada* (/py'ada/), *saiu* (/sa'iw/).

5.5 SEPARAÇÃO DE SÍLABAS

A separação de sílabas é feita utilizando um algoritmo desenvolvido por Leda Lúcia Spelta [33] e é descrito a seguir. A separação é feita sobre os fones. Conforme mencionado anteriormente, para realizar a separação de sílabas é necessário a determinação da sílaba tônica. Isto pode ser justificado observando que o algoritmo utiliza como informação de entrada se as letras i e u correspondem à vogais átonas ou tônicas.

Símbolos utilizados:

Contexto

- **V** - qualquer fone vocálico;
- **yw** - semivogais y e w;
- **lrs** - fones 'l', 'r' e 's';
- **C1** - fones 'b', 'd', 'f', 'g', 'k', 'p', 't' e 'v';
- **C2** - demais fones consonantais;
- **$** - final da palavra (símbolo final);

Ação

- **a** - separar a sílaba antes da letra;
- **d** - separar a sílaba depois da letra;
- $\mathbf{d_2}$ - separar a sílaba depois de duas letras;

Estado

- **1** - estado inicial;
- **6** - estado final;

A fig. 5.2 apresenta o algoritmo na forma de tabela de transição.

Est	*V*	*yw*	*l r s*	*C1*	*C2*	$
1	*2*	*5*	*1*	*1*	*1*	*d6*
2	*d2*	*d3*	*4*	*a1*	*a1*	*d6*
3	$d_2 2$	*3*	*4*	*a1*	*a1*	*d6*
4	*d2*	*d5*	*d1*	*a1*	*d1*	*d6*
5	*2*	*5*	*4*	*a1*	*a1*	*d6*
6	—	—	—	—	—	—

Fig. 5.2. Tabela de transição para separação de sílabas

O estado inicial é sempre o estado *1*. A análise é feita partindo dos fones à direita em direção aos fones à esquerda. Cada coluna da tabela corresponde a um contexto ao qual o fone em análise pertence. Cada linha da tabela corresponde a um estado. As entradas na tabela indicam qual ação deve ser tomada e para qual estado deve ocorrer a transição. À medida que ocorrem as transições entre os estados, os fones que compõem a palavra são analisados sucessivamente. O estado 6 indica simplesmente o final da palavra.

Para exemplificar tomemos a palavra /psykolojia/. Partindo inicialmente do estado *1* teremos uma seqüência de análise conforme mostrado na fig. 5.3.

Passo	estado	fone	classe	próximo estado	resultado
1	*1*	**a**	*V*	*2*	*psykolojI***a**
2	*2*	**i**	*V*	*2*	*psykoloj***i-a**
3	*2*	**j**	C_2	*1*	*psykolo-***ji-a**
4	*1*	**o**	*V*	*2*	*psykol***o-ji-a**
5	*2*	**l**	*l r s*	*4*	*psyko***lo-ji-a**
6	*4*	**o**	*V*	*2*	*psyk***o-lo-ji-a**
7	*2*	**k**	C_1	*1*	*psy-***ko-lo-ji-a**
8	*1*	**y**	*yw*	*5*	*psy-***ko-lo-ji-a**
9	*5*	**s**	*l r s*	*4*	*p***sy-ko-lo-ji-a**
10	*4*	**p**	C_1	*1*	**psy-ko-lo-ji-a**

Fig. 5.3. Seqüência de passos na separação das sílabas da palavra psicologia

CAPÍTULO 6

SÍNTESE DE VOZ

Neste capítulo trataremos da produção do sinal de voz no sistema, através da geração dos parâmetros de síntese e o seu fornecimento ao sintetizador.

6.1 INTRODUÇÃO

Uma vez obtida a transcrição fonética do texto de entrada, determinada a sílaba tônica e realizada a separação das sílabas de cada palavra, podemos efetuar a síntese do sinal de voz propriamente dita.

A produção do sinal de voz é realizada fornecendo uma seqüência de parâmetros ao filtro de síntese LPC, sendo estes parâmetros correspondentes aos difones escolhidos após a análise do texto. Os parâmetros devem ser modificados adequadamente a fim de prover a adaptação de seus valores obtidos isoladamente ao novo contexto.

Com relação aos coeficientes, é necessário que se processe algum tipo de suavização nas junções entre difones, efetuando uma interpolação de parâmetros na região de transição. Para o período de pitch e a duração, deve-se adaptá-los para fornecer um ritmo e uma entonação corretos.

6.2 INTERPOLAÇÃO DOS COEFICIENTES

Para realizar a interpolação dos coeficientes LPC, adaptamos uma equação a princípio utilizada para interpolação de freqüências formantes [34]. Este algoritmo usa como parâmetro uma taxa de variação espectral obtida a partir das freqüências formantes e que para nosso caso será obtido a partir dos coeficientes

LPC. A variação dos coeficientes nos dá uma medida indireta de variação espectral. Por isso chamaremos à taxa de variação dos coeficientes de derivada espectral. A derivada espectral DEi é calculada da seguinte maneira:

$$DE_i = \sum_{j=1}^{8} | C_j(i) - C_j(i-1) |$$

onde

i é o i-ésimo quadro e $C_j(i)$ é o valor do j-ésimo coeficiente no quadro i.

A transição dos coeficientes nas junções entre dois difones será feita durante um intervalo de superposição, levando em conta a derivada espectral média dos difones nos dois lados da junção. A derivada espectral média no intervalo de superposição t_c é dada por:

$$\overline{DE_1} = \frac{1}{t_c} \sum_{i=1}^{t_c} DE_1(i)$$

$$\overline{DE_2} = \frac{1}{t_c} \sum_{i=1}^{t_c} DE_2(i)$$

onde DE1 e DE2 são as derivadas espectrais dos dois difones concatenados durante os t_c quadros de superposição.

A função de interpolação dos coeficientes é dada por:

$$C_j(i) = \frac{C_j^1(i)\,(t_c - i + 1)\,\overline{DE_1} + C_j^2(i)\,i\,\overline{DE_2}}{(t_c - i + 1)\,\overline{DE_1} + i\,\overline{DE_2}} \qquad \begin{matrix} i = 1, 2, \ldots, t_c \\ j = 1, 2, \ldots, 8 \end{matrix}$$

onde

$C_j(i)$ é o valor do j-ésimo coeficiente para o quadro i durante a região de superposição.

$C_j^k(i)$ é o valor do j-ésimo coeficiente para o quadro i durante a região de superposição para o difone k.

Esta equação fornece uma transição de modo a acompanhar mais a curva cuja derivada espectral forneça um valor maior. Na região de superposição, os valores dos coeficientes dos dois difones concatenados são substituídos pelos valores interpolados. Isto provoca uma diminuição na duração total dos dois difones, o

que permite que os difones sejam isolados com uma duração mais longa, tornando menos restritivo o processo de segmentação. De acordo com testes feitos tentando variar o intervalo de superposição, chegamos a um bom valor de tc igual a 4 quadros, o que equivale a um intervalo de 40 ms.

Apesar da possibilidade da interpolação direta dos coeficientes de predição poder levar a uma instabilidade do filtro de síntese, realizamos testes para vários casos e não tivemos problemas. Uma maneira de garantir a estabilidade, é trabalhar com coeficientes de reflexão ou coeficientes log-área, que podem ser calculados a partir dos coeficientes de predição [13]. Porém os resultados de teste subjetivos realizados no laboratório demonstraram uma inferioridade na qualidade obtida com este procedimento.

6.3 INCORPORAÇÃO DE PROSÓDIA

A fim de que um sistema de síntese possa produzir uma voz de boa qualidade, ou seja, possua tanto uma boa inteligibilidade como naturalidade, é necessário incorporar à fala sintetizada uma prosódia correta. A prosódia, ou características suprasegmentais, carrega informação lingüística sobre um domínio além dos limites do fone. Através da variação no tempo dos parâmetros prosódicos tais como intensidade, duração e freqüência fundamental (F0), é possível imprimar à fala a prosódia. Portanto são estes parâmetros que devem ser controlados no sistema de síntese a fim de que possamos obter uma fala de boa qualidade. Uma das funções primárias da prosódia é fornecer indicações da localização do acento, ressaltando as sílabas acentuadas contra um conjunto de sílabas não acentuadas. Isto vai criar a sensação de ritmo da língua. A prosódia pode fornecer pistas acerca das estruturas sintáticas, resolvendo possíveis ambigüidades. Permite ainda que através da segmentação de enunciados longos em unidades menores, um dado texto seja compreendido e que uma relação entre estas unidades possa ser estabelecida. Estas funções podem ser consideradas de caráter lingüístico. A prosódia também pode ter funções expressivas, que refletem o estado emocional do locutor, tais como o medo, raiva, tristeza ou alegria.

A seguir, faremos uma discussão de como os parâmetros prosódicos podem ser controlados em um sistema de síntese, a fim de reproduzir na fala sintetizada alguns aspectos da fala natural.

6.3.1 Duração

As durações dos fones variam de acordo com vários fatores tais como estilo de fala, posição do acento, posição da palavra na frase, pausas, limites da sílaba, modo e ponto de articulação e ritmo.

A fim de determinar como estes fatores atuam de modo a prever a duração dos segmentos nas sentenças, seria necessário o desenvolvolvimento de um modelo descritivo capaz de realizar tal função. Dado o número de variáveis envolvidas e a complexidade das relações entre elas, isto torna-se uma tarefa bastante complexa. Uma maneira mais simples de abordar tal problema é realizar o estudo de efeitos duracionais que carregam informação lingüística. Esta é a metodologia utilizada por Klatt [35], e cujo modelo serve de base para a realização de estudo análogo realizado por Simões [36] para as vogais {*a*, *i* ,*u*} do Português brasileiro. O objetivo do modelo é gerar um conjunto de regras capazes de prever a duração dos segmentos em sentenças, dado um determinado contexto fonético.

O modelo de Klatt baseia-se nas seguintes suposições:

a) Cada segmento possui uma duração intrínseca. Esta duração intrínseca corresponde ao valor médio da distribuição de valores que a duração daquele segmento pode assumir.
b) Cada regra tenta prever uma variação porcentual a fim de efetuar um aumento ou diminuição da duração do segmento
c) Os segmentos não podem ser reduzidos a valores menores do que uma certa duração mínima.

O modelo pode ser expresso por:

$$Do = Dmin + K \times (Di - Dmin)$$

onde

Do é a duração prevista em qualquer ponto no texto
Di é a duração intrínseca de cada segmento
K é o fator que produz a variação da duração de acordo com a aplicação das regras

O primeiro passo para a aplicação do modelo é a obtenção da duração intrínseca do segmento em questão. Os valores obtidos por Simões para as vogais *a* , *i* e *u* foram os seguintes: [a] 64 ms, [i] 56 ms, [u] 48 ms.

Em seguida aplicam-se as regras para variação da duração. As regras obtidas por Simões são reproduzidas abaixo. As regras são aplicadas em série e portanto, o valor de Do para uma regra serve como valor de Di para aplicação da próxima regra.

Nível 1 (Domínio do segmento acústico)

Regra 1: Inicialização. O valor intrínseco da vogal em questão é obtido.

Regra 2: Se a fricativa surda [s] segue a vogal dentro da mesma sílaba, reduza a vogal de 65%. Caso a consoante seja [x], reduza a vogal de 25%.

Regra 3: Se uma consoante sonora segue a vogal dentro da mesma sílaba, não efetuar alteração.

Nível 2 (Domínio da palavra)

Regra 4: Se a vogal está em posição postônica, reduza a vogal de 25%.

Regra 5: Se a vogal está em posição pré-tônica, não precedida por uma consoante, aumente-a de 10%. Caso a vogal seja precedida por uma consoante, aumente-a de 42%.

Regra 6: Se a vogal estiver em posição imediatamente pré-tônica, aumente-a de 13%.

Regra 7: Se a vogal está em posição tônica, aumente-a de 90%.

Nível 3 (Domínio da sentença)

Regra 8: Se a vogal está no começo de uma sentença ou uma pausa, não efetuar alteração.

Regra 9: Se a vogal está em uma posição no meio da sentença, reduza-a de 13%.

Regra 10: Se a vogal está em posição final na sentença sem pausa física, aumente a vogal de 20%. Se a vogal estiver em posição final principal, e uma pausa física se segue, aumente a vogal de 32%.

Nível 4 (Domínio semântico)

Regra 11: Se a vogal está dentro de uma palavra foco, aumente a vogal de 60%.

Regra 12: Se a vogal está em uma palavra exclamatória, aumente a vogal de 80%.

De acordo com os resultados obtidos por Simões para o Português Brasileiro, não existe uma duração mínima para os segmentos estudados, podendo estes ser reduzidos completamente. Neste caso o modelo pode ser equacionado simplesmente por:

$$Do = K \times Di$$

Para o método de síntese que estamos utilizando (síntese por concatenação), a duração intrínseca dos fones será determinada pela duração dos elementos armazenados, sendo no nosso caso obtida no momento da segmentação dos difones. Portanto, se desejamos obter uma duração intrínseca, para que a partir dela possamos aplicar as regras e chegar à duração final, é necessário segmentarmos os difones de maneira que a duração desejada para os segmentos seja conseguida. É necessário lembrar que o difone contém apenas parte de cada segmento e levar em conta que a duração final é afetada pelo superposição utilizada na interpolação.

A variação da duração é feita através da repetição ou retirada de quadros de análise LPC correspondentes à vogal. No caso em que a variação é diferente de um número inteiro de quadros, o tamanho de um quadro, que tipicamente dura 10 ms, é alterado de modo a fornecer a variação desejada.

O aumento da duração é feito através da repetição ou alteração de um quadro pertencente à região de interpolação entre os dois difones que contém a vogal em questão. No caso de vogal inicial ou final, pode-se utilizar qualquer quadro.

O caso da redução da duração é um pouco mais complicado, pois à medida que vamos retirando os quadros, é preciso saber qual o limite da vogal dentro do difone. Portanto, é necessário possuirmos em cada difone a informação acerca do início da consoante ou da vogal do difone em questão. Para que isso possa ser realizado, é necessário sabermos quais são os quadros que correspondem à vogal e quais correspondem à consoante. Essa marca deverá ser determinada na fase de desenvolvimento do dicionário de difones. No sistema que desenvolvemos não foi realizada tal marcação. Isto se deve ao fato de que, por ser essa a primeira implementação do sistema, o nosso objetivo principal foi direcionado no desenvolvimento do dicionário de difones sem preocupação com informações necessárias para a implementação de variação dos parâmetros prosódicos. Apesar disso, implementamos algumas regras de duração de forma aproximada. Isso foi feito considerando que uma vez conhecida a duração intrínseca das vogais, é possível estimar em qual quadro a vogal se inicia. Isto pode ocasionar um erro em alguns casos

pois a duração do mesmo segmento não é exatamente a mesma para diferentes difones. O erro porém não chega a ser excessivo e procedendo deste modo podemos avaliar pelo menos aproximadamente o efeito da incorporação das regras de duração.

No sistema desenvolvido, com exceção das regras 11 e 12, todas as demais regras foram aplicadas.

6.3.2 Freqüência Fundamental (Fo)

Dos parâmetros prosódicos, a freqüência fundamental desempenha o papel mais importante no que se refere à capacidade de transmitir informação lingüística. Sua variação no tempo constitui o chamado padrão de entonação. De acordo com a sua variação no tempo, Fo pode carregar informação sobre a estrutura sintática ou de padrões de acentuação. Essa informação auxilia o ouvinte a delimitar e segmentar a mensagem a fim de decodificá-la. Outra função importante da entonação é de indicar a proeminência das palavras mais importantes no discurso. Esta proeminência pode ser indicada por subidas ou quedas de Fo. Portanto, um modelamento correto da entonação é um importante fator determinante no aumento da inteligibilidade e melhora da qualidade em um sistema de síntese de voz a partir de texto.

Além das funções relacionadas acima, o padrão de Fo ainda desempenha um papel complexo na codificação de informação para o ouvinte, indicando o estado psicológico, sentimento em relação ao que está sendo falado e ênfase. A entonação pode auxiliar também em alguns casos a resolver possíveis ambigüidades sintáticas.

Para sistemas de síntese de entonação por regras, é necessária uma teoria que possa prever quando Fo subirá ou descerá, e quais valores ela atingirá na sentença em função de informações tais como estrutura sintática, padrão de acentuação, localização da palavra mais importante na sentença. A partir desta teoria é possível criar um modelo de modo a tentar gerar de maneira automática um padrão de entonação que seja o mais próximo daquele que ocorreria em um enunciado natural.

Vários modelos têm sido propostos para gerar contornos de entonação sintéticos para o inglês. Como exemplo podemos citar dois algoritmos utilizados em sistemas de voz a partir de texto, desenvolvidos por Pierrehumbert [37] e O'Shaughnessy [30]. Estes dois algoritmos supõem que o contorno de entonação ao

longo do enunciado pode ser visto como uma superposição de efeitos, sendo estes global ou local. Para enunciados assertivos, o efeito global apresenta uma tendência decrescente de F0, conhecida como declinação. Esta declinação pode ser descrita por duas linhas, uma linha de topo e uma linha de base, as quais possuem inclinações ambas negativas, porém distintas. Superposta a esta tendência decrescente de F0, teremos variçÕes internas cujos valores excursionam delimitados pelas duas linhas de declinação.

Os estudos da entonação do português brasileiro ainda são bastante restritos, sendo que não temos conhecimento de um estudo descritivo suficientemente completo para que um modelo aplicável em síntese de voz a partir de texto pudesse ser desenvolvido. Devido a esta limitação, não pudemos incorporar em nosso sistema um algoritmo de controle da freqüência fundamental.

CAPÍTULO 7

IMPLEMENTAÇÃO EM TEMPO REAL

Neste capítulo descreveremos os procedimentos utilizados a fim de permitir que o sistema opere em tempo real.

7.1 INTRODUÇÃO

Para poder efetuar a síntese em tempo real o filtro de síntese LPC foi implementado utilizando uma placa baseada no processador de sinais TMS320C30.

Os sistemas de síntese de voz a partir de texto podem ser divididos basicamente em duas partes principais. A primeira parte trata do processamento do texto e a geração de parâmetros de controle de um sintetizador por meio da aplicação de um conjunto de regras de base lingüística. A segunda parte consiste na geração do sinal acústico propriamente dito, por meio de um sintetizador que recebe os parâmetros de controle e produz o sinal de voz correspondente. Esta divisão sugere que uma maneira interessante de implementação seja adotada, utilizando um microcomputador de uso genérico para realizar os processamentos da primeira parte e implementando o sintetizador por meio de um processador de sinais digitais (DSP). O sintetizador é implementado em DSP por ser a parte do sistema que exige a maior quantidade de cálculos e quenão sofre alterações uma vez escolhido um modelo de síntese. O computador por ser mais flexível em termos de entrada e saída de dados facilita o processo de alteração das regras.

A vantagem da implementação do sistema do modo sugerido, é podermos realizar a síntese em tempo real, no sentido de efetuar o cálculo de novas amostras de saída à medida que as anteriores vão sendo enviadas para o conversor D/A, e ao mesmo tempo possuir uma flexibilidade para testar e modificar constantemente as regras de processamento de texto e geração de parâmetros. Em um sistema dedicado, visando portabilidade, pode-se substituir o microcomputador por circuitos mais específicos utilizando um microprocessador como controlador.

Neste capítulo tentaremos apresentar algumas características do processador TMS320C30, bem como a placa de processamento de sinais e o procedimento utilizado na implementação do sistema de síntese em tempo real.

7.2 O PROCESSADOR DE SINAIS TMS320C30

Apesar dos detalhes técnicos sobre o TMS320C30 poderem ser encontrados no Guia do Usuário [39], o objetivo desta seção é apresentar de maneira resumida algumas de suas características principais que o tornam uma opção atraente para aplicações em sistemas de síntese. O TMS320C30 é um processador de sinais digitais de 32 bits e que opera em ponto flutuante. Seu barramento interno e conjunto de instruções permitem a execução de até 33 MFLOPS (milhões de operações em ponto flutuante por segundo). Este alto desempenho é obtido através da implementação de funções em hardware que outros processadores implementam em software ou através de micro-código.

7.2.1 Arquitetura

A arquitetura do TMS320C30 permite que um alto desempenho possa ser obtido. Algumas de suas principais partes são descritas genericamente a seguir. Na fig. 7.1 temos representado um diagrama em blocos da arquitetura do TMS320C30.

7.2.2 Unidade de Processamento Central (UPC)

A UPC consiste dos seguintes componentes:

Multiplicador

O multiplicador realiza multiplicações em um único ciclo de instrução de 60 ns. As multiplicações de inteiros são efetuadas com 24 bits e as multiplicações em ponto flutuante são efetuadas com 32 bits.

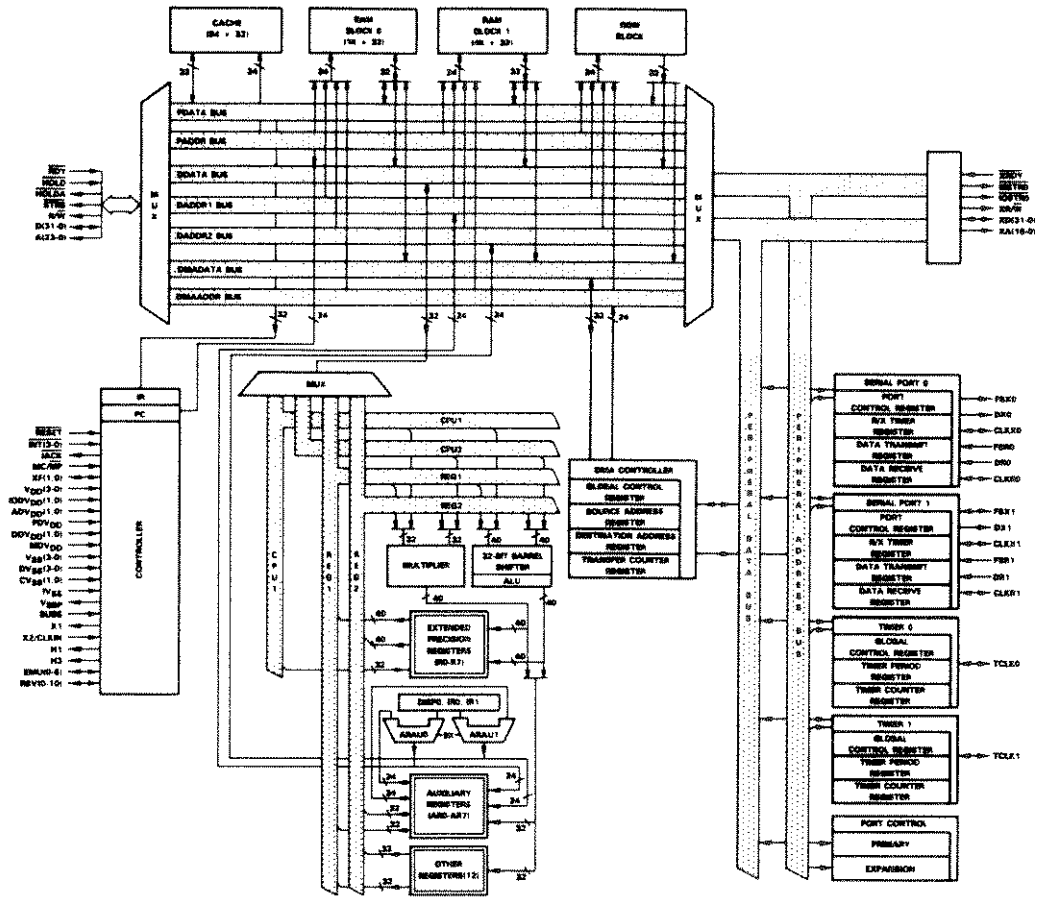


Fig. 7.1. Diagrama em blocos da arquitetura do TMS320C30

Unidade Lógica e Aritmética (ULA)

A ULA realiza operações em único ciclo com dados inteiros de 32 bits e ponto flutuante de 40 bits. Os resultados da ULA são sempre mantidos nos formatos de inteiro de 32 bits e ponto flutuante de 40 bits.

Unidades Aritmética de Registrador Auxiliar (UARAs)

A função das UARAs (UARA0 e UARA1) é realizar operações aritméticas em paralelo com o multiplicador e a ULA para geração de endereços. É possível realizar em um único ciclo a geração de dois endereços.

Arquivo de Registradores da UPC

O TMS320C30 possui 28 registradores sobre os quais o multiplicador e a ULA podem operar, e que podem ser usados como registradores de propósito geral. Contudo, os registradores também têm algumas funções especiais. Por exemplo, os

oito registradores de precisão estendida (R0-R7) são especialmente adequados para manter resultados em ponto flutuante de precisão estendida. Os oito registradores auxiliares (AR0-AR7) suportam uma variedade de modos de endereçamento e podem ser usados como registradores de propósito geral para dados lógicos e inteiros de 32 bits. Os registradores restantes provêm funções de sistema tais como endereçamento, gerenciamento de pilha, status do processador, interrupções e repetição de bloco. Um diagrama da UPC é mostrada na fig. 7.2

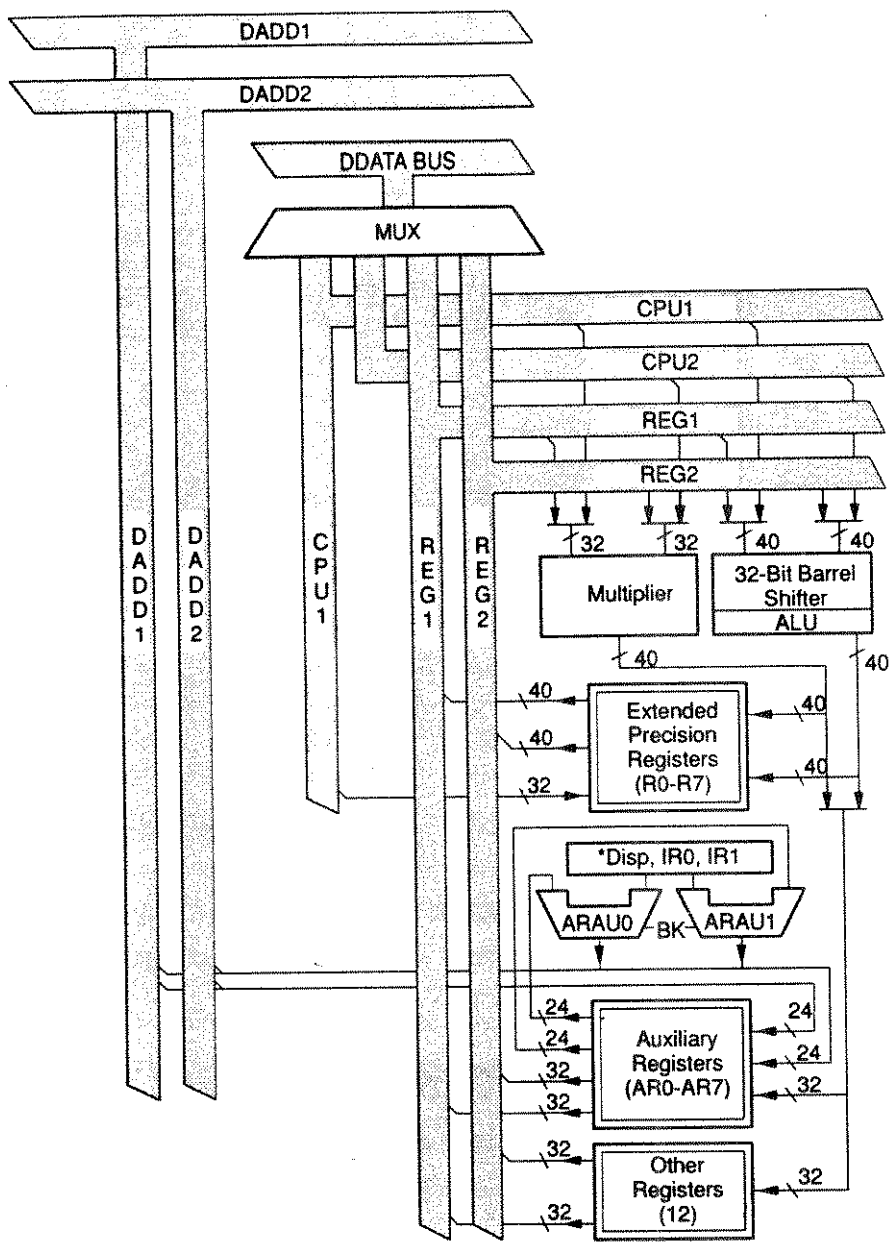


Fig. 7.2. Unidade de Processamento Central (UPC)

7.2.3 Organização de Memória

O espaço total de memória do TMS320C30 é de 16M palavras de 32 bits. Dois blocos de memória RAM interna, cada um de 1k x 32 bits são disponíveis. Um bloco de memória ROM de 4K x 32 bits também é disponível. Um cache de instrução de 64 x 32 bits é provido para armazenar seções de códigos freqüentemente repetidas, reduzindo o número de acessos à memória externa. A fig. 7.3 mostra a organização de memória do TMS320C30.

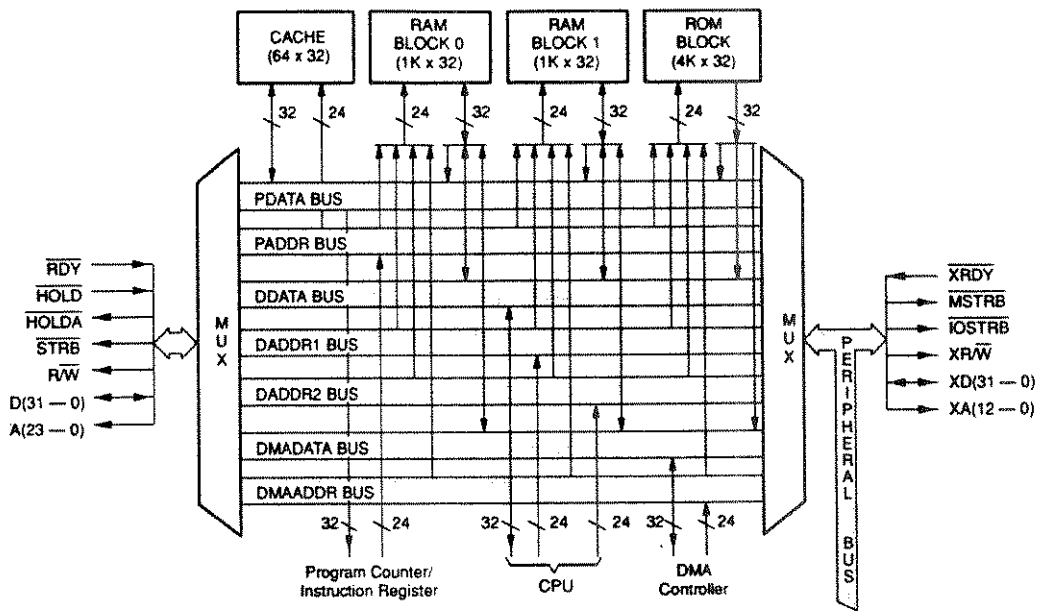


Fig. 7.3. Organização de memória do TMS320C30

7.2.4 Periféricos

Os periféricos do TMS320C30 incluem dois timers e duas portas seriais. Os dois módulos timer são contadores de timer/evento de propósito geral de 32 bits com dois modos de sinalização e controle de clock interno ou externo. Cada timer tem um pino de I/O que pode ser usado como uma entrada de clock para o timer ou como um sinal de saída acionada pelo timer.

As duas portas seriais são totalmente independentes. Cada porta serial pode ser configurada para trasferir 8, 16, 24 ou 32 bits de dados por palavra. O clock para cada porta pode se originar tanto interna como externamente. A fig. 7.4 mostra os periféricos com seus respectivos barramentos e sinais de controle.

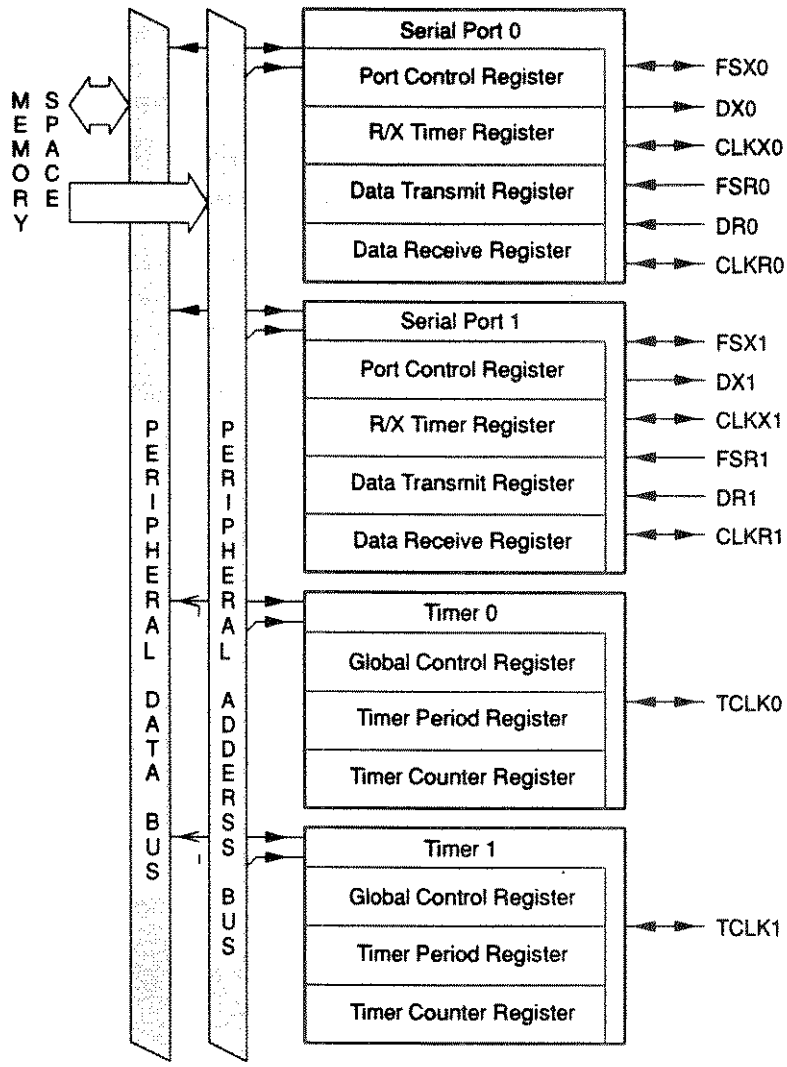


Fig. 7.4. Periféricos

7.2.5 Acesso Direto a Memória (ADM)

O controlador de ADM pode ler de ou escrever em qualquer posição de memória acessível sem interferir com a operação da UPC. A fig. 6.5 mostra o controlador de ADM e seus respectivos barramentos.

7.2.6 Ferramentas de desenvolvimento de software para o TMS320C30

Um compilador C da Texas Instruments é disponível para o desenvolvimento de programas aplicativos para o TMS320C30. Este compilador aceita código fonte C baseado no padrão ANSI e produz código fonte em linguagem assembly do TMS320C30

[40]. Em seguida um programa assembler é utilizado para transformar os arquivos fonte em linguagem assembly para arquivos objeto em linguagem de máquina no formato COFF (Common Object File Format). O linker combina arquivos objeto em um único módulo objeto executável [41].

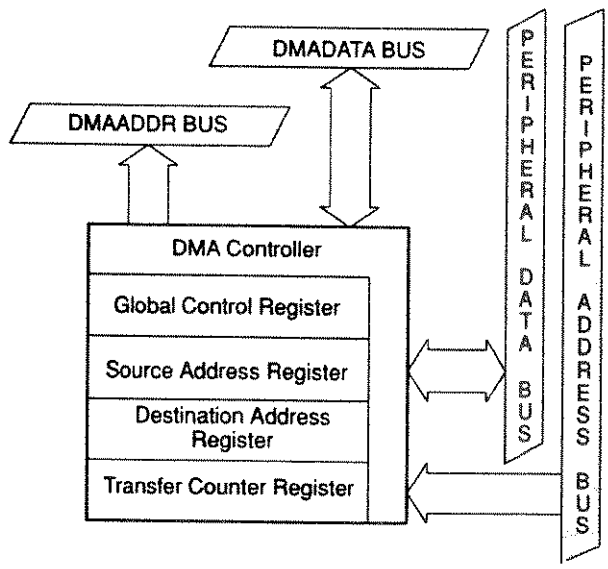


Fig. 7.5. Controlador de ADM

7.3. A PLACA PARA PC UTILIZANDO O TMS320C30

No estágio de síntese e conversão foi utilizada uma placa produzida pela Loughborough Sound Images Limited. Esta placa é baseada no processador TMS320C30 e funciona acoplada a um microcomputador do tipo PC. Abaixo temos uma especificação da configuração da placa em termos de hardware e software disponível.

7.3.1 Hardware

Memória

Duas áreas de memórias são providas. Em uma área estão disponíveis 64K words de 32 bits, normalmente utilizadas para guardar vetores de interrupção e o código do programa executável. Em outra área temos disponíveis 64K words de 32 bits compartilhada entre o PC e o TMS320C30.

Interface Analógica

Sistemas A/D e D/A de 16 bits para dois canais são incluídos na placa. Taxas de amostragem de até 200 kHz são possíveis. Os canais de entrada incluem amplificadores sample/hold, e ambos canais de entrada e saída são bufferizados por filtros passa-baixas Sallen-Key de quarta ordem. As freqüências de corte podem ser selecionadas efetuando a troca de conjunto de resistores. Estes filtros não são muito seletivos e optamos por realizar uma filtragem externa utilizando filtros elípticos de corte mais abrupto com a freqüência de corte em 3.4 kHz.

7.3.2 Software

O Monitor Debug

Dois programas monitores são providos para controlar a placa e auxiliar na depuração de programas. Com estes programas é possível carregar programas do TMS320C30 na placa, executá-los e executar comandos para auxiliar na depuração. Durante o processo de depuração é possível executar o programa passo a passo, ou determinar pontos de parada. Após a execução de uma instrução ou após os pontos de parada, é possível examinar ou alterar o conteúdo da memória e dos registradores do TMS ou o conteúdo da memória da placa. Os programas monitores diferem na apresentação da informação, sendo uma versão mais simplificada com instruções de linha de comando e outra utilizando um sistema de menus e janelas.

Biblioteca de Interface

A fim de poder controlar a execução de programas a partir do microcomputador, uma biblioteca de interface é disponível. Esta biblioteca, cujos arquivos fonte se encontram escritos em linguagem C, são capazes de realizar funções tais como inicializar a placa, transferir dados para a memória ou ler dados da memória da placa.

Por meio desta biblioteca é possível desenvolver aplicações onde a execução do programa no processador pode correr em interação com o PC. Esta é uma situação bastante interessante no caso dos sistemas de voz a partir de texto, pois podemos implementar a parte do sintetizador que exige um maior quantidade de cálculos em DSP, deixando a parte de análise do texto para ser efetuada no computador. Desse modo, o computador, por meio da análise do texto, determina os

parâmetros correspondentes ao texto analizado e envia-os ao sintetizador. Algumas das funções que foram utilizadas e que permitiram esta interface do computador com a placa são apresentadas a seguir:

SelectBoard()

Esta função é utilizada para inicializar a placa do TMS320C30 para operação com as outras funções da biblioteca. A comunicação da placa com o PC é feita através do espaço de I/O do PC. Para o programador, a placa é vista como um bloco de oito palavras de 16 bits dentro do espaço de I/O. O endereço base deste bloco pode ser selecionado entre 256 possibilidades por meio de links na placa. Ao chamar a função SelectBoard(), especifica-se o endereço base selecionado.

LoadObjectFile()

Esta função transfere um arquivo executável no formato COFF do PC para a memória da placa do TMS320C30. Após a execução desta função, o programa pode ser executado pelo processador.

WrBlkFlt()

Preenche um dado número de posições de memória da placa com valores em ponto flutuante de um dado vetor. A posição de memória inicial é dada pelo endereço fornecido. Os valores são convertidos do formato IEEE para formato do TMS320C30 antes de serem colocados na memória.

PutInt()

Coloca um inteiro de 32 bits na memória da placa do TMS320C30 na posição indicada.

GetInt()

Lê um inteiro de 32 bits da memória da placa do TMS320C30 no endereço indicado.

Reset()

Pulsa o sinal de RESET de hardware da placa, fazendo o chip começar execução com um salto para a posição de memória dada pelo vetor de reset (posição 0).

Esta função pode ser usada para controlar o início da execução do programa na placa. Para isso, inicializa-se o vetor de reset com o endereço inicial do programa e executa-se a função Reset() no momento que a execução do programa deve começar.

A Interface Analógica

A interface para sinais analógicos provê dois canais para entrada e dois canais para saída, chamados canais A e B. A interface é acessada por meio de três registradores de 16 bits, que são conectados ao barramento de expansão do TMS320C30. Seus endereços estão na região entre 804000h e 804008h. Embora cada registrador possua 32 bits de comprimento, apenas os 16 bits do topo são usados.

Os primeiros dois registradores acessam os conversores A/D e D/A nos dois canais analógicos de I/O. Os conversores operam com dados no formato de complemento de dois de 16 bits. Para saída de dados em um canal deve-se mover dados para o registrador mapeado em I/O localizado em 804000h. Para entrada de dados do conversor no canal B, deve-se ler os dados do registrador mapeados no mesmo endereço. A entrada e saída de dados para o canal A é feita através do endereço 804001h.

Os A/Ds e D/As usam o mesmo registrador para saída e entrada. Se não houver intervenção do processador, o valor do A/D será deslocado para dentro do registrador, com o conteúdo prévio do registrador sendo simultaneamente deslocado para o D/A. Portanto, o sinal analógico de entrada em cada canal será por default ecoado diretamente para o correspondente canal analógico de saída, com um atraso de um intervalo de amostragem. Caso se deseje ler um valor do A/D e escrever um valor diferente no mesmo canal durante o mesmo intervalo de amostragem, então deve-se ler o A/D antes de escrever para o D/A.

Controle de tempos

As conversões A/D e D/A em ambos canais podem ser iniciadas tanto por um sinal de trigger externo ou pelo contador/timer interno do TMS320C30. A última opção normalmente é selecionada. O intervalo de amostragem é ajustado pela programação de um contador/timer interno ao chip. Os dois canais de I/O analógicos estão sempre sincronizados de modo que não é possível ter duas taxas de amostragem diferentes para os dois canais.

O timer 1 do TMS320C30 é usado para controle do intervalo de amostragem. Ele é composto por um contador progressivo de 32 bits e um registrador de período de 32 bits. O valor no contador é continuamente incrementado a uma taxa de 8.33 Mhz (uma vez a cada 120 ns). Quando o contador se iguala ao registrador de período, ele produz um pulso que inicia as conversões A/D e D/A, e zera o valor de contagem do contador. Para colocar o timer no modo correto, inicialmente es-

creve-se 6C1h no endereço 808030h. Isto habilita o timer 1 para usar o clock interno e produzir um pulso negativo. A taxa de amostragem é então ajustada escrevendo um valor para o Registrador de Período na posição 808038h. Para encontrar o valor de contagem, primeiro calcula-se o período de amostragem (o tempo entre amostras). O valor de contagem é dado pelo período de amostragem dividido pelo período de contagem (120 ns). A melhor maneira de tratar com os A/Ds e D/As é através de uma rotina de serviço de interrupçaõ. Após o timer iniciar uma conversão A/D, o A/D realizará a conversão e então produzirá na saída um sinal de fim-de-conversão. Este sinal de fim-de-conversão é conectado ao TMS320C30 como um pedido de interrupção INT1.

7.4 IMPLEMENTAÇÃO DO SISTEMA EM TEMPO REAL

Para implementar o sistema de síntese em tempo real, a parte da síntese LPC foi feita utilizando-se a placa de sistema TMS320C30. Duas versões do sistema, utilizando de maneira diferente o processador, foram desenvolvidas. Em ambas as versões, um processamento inicial do texto para determinação dos parâmetros necessários para o sintetizador é realizado, e em seguida um conjunto de parâmetros é escrito na memória da placa.

A primeira etapa para a criação do programa em tempo real foi a geração de um programa para ser executado pelo TMS320C30. Para isso foi utilizado inicialmente o compilador C para o TMS320C30. A parte do programa implementado na placa consistia do bloco de programa correspondente à geração da excitação e à realização do filtro de síntese LPC, de modo que a entrada para este bloco é composta pelo valor do período de pitch, o ganho, os oito coeficientes de predição e o comprimento do quadro em amostras.Este bloco de programa foi compilado a fim de gerar um arquivo em linguagem assembly.

Juntamente com o algoritmo de síntese, incorporamos a rotina de serviço de interrupção para enviar as amostras calculadas para o conversor D/A para serem reproduzidas. Utilizamos o processo de conversão D/A iniciada pelo contador/ timer. Como utilizamos uma freqüência de amostragem de 8 kHz, o valor carregado no Registrador de Período foi de 1042. Portanto a cada intervalo de 125 μs uma interrupção é originada e a rotina de serviço de interrupção chamada. Esta rotina lê uma amostra de saída que já foi calculada e armazenada na memória e envia-

a para o conversor D/A. Esta rotina que é executada na placa do TMS320C30, é a mesma para as duas versões do sistema de síntese. A diferença entre os dois programas é como o programa executado no PC interage com a placa.

Em uma das versões, à medida que os parâmetros de cada quadro LPC são determinados a partir da análise do texto, eles vão sendo escritos na memória da placa. A entrada de texto neste caso é feita pelo teclado. Após os parâmetros correspondentes ao texto completo terem sido enviados para a placa, a função Reset() é executada, o que faz com que o processo de cálculo e envio de amostras para o conversor se inicie. Neste processo, as amostras ficam armazenadas na memória da placa, podendo ser recuperadas e guardadas em um arquivo se assim for desejado. Pelo processo descrito, fica claro que este tipo de síntese só se aplica para textos pequenos, de tamanho restrito devido a limitação de memória da placa. Este programa possui um pequeno atraso, pois deve-se esperar primeiro que os parâmetros sejam calculados e armazenados para que então a síntese possa começar. Porém, como este tipo de síntese assume que somente textos pequenos serão sintetizados, isto não representa uma limitação séria. Esta versão é mais adequada para testar pequenos pedaços de fala durante a fase de desenvolvimento do sistema, pois permite que o sinal possa ser armazenado para posterior análise.

A segunda versão do sistema permite que textos de tamanho irrestrito, armazenados em arquivo, sejam sintetizados em tempo real. Nesta versão do sistema o sinal de voz não é armazenado. A transferência de parâmetros é feita considerando trechos pequenos do texto de entrada para análise. A escolha dos trechos de texto é feita de modo a coincidir com as pausas naturais marcadas por pontuação. Este pedaço de texto é analisado e a respectiva seqüência de parâmetros transferida para a placa do TMS320C30. Logo após a transferência deste bloco de parâmetros, o processamento na placa começa a síntese do respectivo sinal, enquanto o processamento no PC inicia a leitura e análise do próximo trecho de texto. Desse modo, a reprodução do texto pode ser feita em tempo real, pois enquanto a placa realiza a síntese de uma parte do texto, o PC está processando a análise e geração dos parâmetros do fragmento de texto seguinte. Para sincronizar a transferência utilizamos uma posição da memória da placa, de tal modo que quando a placa termina o processamento correspondente ao trecho de texto atual, ela escreve um valor na posição de memória indicando que o processamento terminou e fica esperando um novo conjunto de parâmetros. O programa no PC, lê então esta posição de memória antes de escrever o próximo bloco de parâmetros na memória da placa.

Os difones, por ocuparem uma quantidade muito grande de memória, tiveram que permanecer armazenados na forma de arquivo. Todos os difones foram reunidos em um único arquivo a fim de facilitar a manipulação dos dados. Este arquivo permanecia guardado em um disco virtual, criado de modo a permitir o armazenamento em memória, e portanto um acesso rápido aos parâmetros do difone.

CAPÍTULO 8

CONCLUSÕES

Neste capítulo discutiremos os resultados obtidos e propostas para futuros estudos.

8.1 CONSIDERAÇÕES SOBRE O TRABALHO DESENVOLVIDO

A maneira mais eficiente de se avaliar um sistema de conversão texto-voz é através de testes de inteligibilidade. As medidas de inteligibilidade podem ser feitas considerando palavras isoladas, palavras em sentenças ou compreensão de parágrafos inteiros [42]. A psicolingüística atua na preparação dos estímulos perceptuais, e na avaliação dos resultados. Estes testes geralmente são realizados em sistemas completos, em estágio avançado.

No trabalho desenvolvido, não procuramos realizar qualquer tipo de teste de inteligibilidade pelo fato de ser um sistema implementado visando somente iniciar estudos na área e adquirir alguma experiência, para posteriormente realizar um trabalho mais completo. O sistema é capaz de gerar voz, a partir de um texto digitado pelo teclado ou a partir de um arquivo armazenado, sem controle de freqüência fundamental. A seguir consideraremos algumas dificuldades encontradas durante o desenvolvimento do trabalho e as conclusões obtidas.

A maior dificuldade que tivemos ao iniciar o desenvolvimento do sistema de conversão texto-voz para o Português, foi de encontrar estudos lingüísticos básicos sobre o Português do Brasil que pudessem auxiliar no desenvolvimento dos módulos que realizam as diversas tarefas do sistema de síntese. A simulação do processo de produção natural da fala é de grande complexidade e exige que estudos fonéticos experimentais extensivos sejam realizados, a fim de se obter uma descrição das propriedades particulares da língua considerada. Partindo desta descrição, o sistema de síntese procura implementar módulos que simulem a ação dos diversos processos envolvidos. Quanto mais detalhada for a descrição obtida,

melhor será o processo de aproximação da fala sintetizada à fala natural. Embora um grande número destes estudos venha sendo realizado há vários anos para diversas línguas, no Brasil o número de trabalhos ainda é bastante restrito.

Diante da situação apresentada, tivemos que escolher entre duas opções: tentar realizar um trabalho de base, concentrando o estudo em um problema específico relacionado à síntese de voz a partir de texto (estudo de entonação p.ex.), ou partir para a implementação de um sistema mais abrangente, incluindo diversas etapas de processamento. Embora o caminho mais natural fosse adotar a primeira opção como passo inicial, resolvemos adotar a segunda opção pois permitiria adquirir uma visão mais global de sistema, o que auxiliaria no direcionamento das pesquisas futuras. Além disso, permitiria entrar em contato com problemas práticos de implementação, adquirindo experiência no desenvolvimento de metodologias de estudo e de ferramentas computacionais.

8.2 PROPOSTAS PARA FUTUROS TRABALHOS

Com relação ao desenvolvimento posterior do trabalho, uma das principais mudanças propostas é com relação ao método de síntese.

Uma das conclusões a que chegamos utilizando o método de síntese por concatenação, é que existe uma limitação que ele impõe na qualidade do sistema. O fato do método de síntese por concatenação trabalhar com unidades armazenadas, limita o sistema com relação a possíveis mudanças desejadas a fim de implementar algumas variações que ocorrem na fala contínua. Talvez isso não seja uma limitação para algumas aplicações, para as quais a geração de uma fala controlada, do tipo leitura, é aceitável. Porém, à medida que se deseja imitar a fala natural, como a de uma conversação por exemplo, mais detalhes têm que ser considerados. Isto no sistema por regras é possível, pois a fala é gerada a partir de regras que levam em consideração variações contextuais. Desde que se conheça as variações que ocorrem e as regras capazes de controlá-las, é possível incorporar na fala sintetizada detalhes adicionais. Já para a síntese por concatenação existe uma limitação na possibilidade de alteração das unidades armazenadas. Portanto, à medida que os estudos lingüísticos avançam e que uma melhor descrição dos processos fonéticos e fonológicos da língua é obtida, a tendência é que a qualidade no sistema de síntese por regras se torne cada vez mais superior.

Tendo em vista as considerações apresentadas, pretende-se dar continuidade ao trabalho substituindo o sistema de síntese por concatenação por um sistema de síntese por regras utilizando um sintetizador por formantes.

Uma outra área básica que deverá ser explorada é com relação ao estudo dos fenômenos prosódicos. Em particular, pretende-se realizar estudos sobre entonação visando o possível desenvolvimento de um modelo de geração automática de freqüência fundamental. A implementação deste módulo deve fornecer ganhos bastante significativos na qualidade do sistema. Conforme destacamos na seção anterior, a realização de estudos lingüísticos de base é de fundamental importância e a única maneira pela qual os sistemas de conversão texto-voz poderão evoluir.

BIBLIOGRAFIA

1. Makhoul, J. (1975). "Linear Prediction: A Tutorial Review", Proc. IEEE **63**, 561-580.

2. Flanagan, J.L. (1976). "Computers that Talk and Listen: Man-Machine Communication by Voice", Proc. IEEE **64**, 405-415.

3. Olive, J.P., and Nakatani, L.H. (1974). "Rule Synthesis of Speech by Word Concatenation: A First Step", J. Acoust. Soc. Am. **55**, 660-666.

4. Kurzweil, R. (1976). "The Kurzweil Reading Machine: A Technical Overview", in *Science, Technology and the Handicapped,* edited by M.R. Redden and Schwandt (American Association for the Advancement of Science, Report 76-R-11, Washington, DC), pp. 3-11.

5. Dudley, H., Riesz, R.R., and Watkins, S.A. (1939). "A Synthetic Speaker", J. Franklin Inst. **227**, 739-764.

6. Cooper, F.S., Liberman, A.M., and Borst, J.M. (1951). "The Interconversion of Audible and Visible Patterns as a Basis for Research in the Perception of Speech", Proc. Natl. Acad. Sci. (US) **37**, 318-325.

7. Cooper, F.S., Delattre, P.C., Liberman, A.M., Borst, J.M., and Gerstman L.J. (1952). "Some Experiments on the Perception of Synthetic Speech Sounds", J. Acoust. Soc. Am. **24**, 597-606.

8. Klatt, D.H. (1987). "Review of Text-to-Speech Conversion for English", J. Acoust. Soc. Am. **82**, 737-793.

9. Fant, G. (1960). *Acoustic Theory of Speech Production* (Mouton, 's-Gravenhage, The Netherlands).

10. Flanagan, J.L. (1972). *Speech Analysis Synthesis and Perception* (Springer, New York).

11. Fujimura, O. (1962). "Analyis of Nasal Consonants", J. Acoust. Soc. Am. **34**, 1865-1875.

12. Holmes, J.N. (1983). "Formant Synthesizers: Cascade or Parallel", Speech Commun. **2**, 251-273.

13. Rabiner, L., and Schaffer R. (1979). *Digital Processing of Speech Signals* (Prentice Hall: Englewood Cliffs, NJ).

14. Klatt, D.H. (1980). "Software for a Cascade/Parallel Formant Synthesizer", J. Acoust. Soc. Am. **67**, 971-995.

15. Heinz, J.M., and Stevens, K.N. (1961). "On the Properties of Voiceless Fricative Consonants", J. Acoust. Soc. Am. **33**, 589-596.

16. Oppenheim A. and Schaffer (1975). *Digital Signal Processing* (Prentice Hall: Englewood Cliffs, NJ).

17. Markel, J., and Gray, A. (1976). *Linear Prediction of Speech* (Springer-Verlag: New York).

18. Atal, B.S., and Remde, J.R. (1982). "A New Model of LPC Excitation for Producing Natural-Sounding Speech at Low Bit Rates", Proc. Int. Conf. Acoust. Speech Signal Process. **ICASSP-82**, 614-617.

19. Caspers B.E., Atal, B.S. (1983). "Changing Pitch and Duration in LPC Synthesized Speech Using Multi-Pulse Excitation", J. Acoust. Soc. Am. Suppl. 1 **73**, 55.

20. Coker C.H. (1976). "A Model of Articulatory Dynamics and Control", Proc. IEEE **64**, 452-459.

21. O'Shaughnessy, D. (1987). *Speech Communication* (Addison-Wesley, New York).

22. Maia, Eleonora M. (1986). *No Reino da Fala* (Editora Ática: São Paulo, SP).

23. Lehiste, I. (1970). *Suprasegmentals* (MIT Press, Cambridge, MA).

24. Crystal, D. (1969). *Prosodic Systems and Intonation in English* (Cambridge University Press: London).

25. Couper-Kuhlen, E. (1986). *An Introduction to English Prosody* (E. Arnold: London).

26. Cruttenden, A. (1986). *Intonation* (Cambridge University Press: London).

27. Liberman, A.M., Ingemann, F., Lisker, L., Delattre, P., and Cooper, F. (1959). "Minimal Rules for Synthesizing Speech", J.Acoust. Soc. Am. **31**, 1490-1499.

28. Chollet, G., Galliano, J.F., Lefevre, J.P., and Viara, E., (1983) "On the Generation and Use of a Segment Dictionary for Speech Coding, Synthesis and Recognition", Proc. Int. Conf. Acoust. Speech Signal Process. **ICASSP-83**, 1328-1331.

29. Kurt Schäfer- Vincent (1983). "Pitch Period Detection and Chaining: Method and Evaluation", Phonetica **40**, 177-202.

30. Fallside, F. and Woods, W.A. (1985). *Computer Speech Processing* (Prentice Hall: Englewood Cliffs, N.J.) p. 421.

31. Violaro, Fábio (1989). "Uma Nova Versão do Sistema de Análise e Processamento Digital de Voz", 7º Simpósio Brasileiro de Telecomunicações, Florianópolis-SC.

32. Lesk, M.E., and Schmidt E. "Lex-A Lexical Analyzer Generator", Bell Laboratories, Murray Hill, NJ 07974.

33. Spelta, L.L., "Um compilador para a Língua Portuguesa". Laboratório de Computação Científica, CNPq, Rio de Janeiro..

34. Rabiner, L.R., Schaffer, R.W., and Flanagan, J.L. (1971). "Computer Synthesis of Speech by Concatenation of Formant-Coded Words", Bell System Tech. J.50, 1541-1558.

35. Klatt, D.H. (1976). "Linguistic Uses of Segmental Duration in English: Acoustic and Perceptual Evidence", J. Acoust. Soc. Am. 59, 1208-1221.

36. Simões, A.R.M. (1990). "Predicting Sound Segment Duration in Connected Speech: an Acoustical Study of Brazilian Portuguese", In the First International Conference on Speech Synthesis, ETRW-GALF90, September 23-28. Autrans, France: Institut de la communication parlée.

37. Pierrumbert, J. (1981). "Synthesizing Intonation", J. Acoust. Soc. Am. 70, 985-995.

38. O'Shaughnessy, D. (1977). "Fundamental Frequency by Rule for a Text-to-Speech System", Proc. Int. Conf. Acoust. Speech Signal Process. ICASSP-77, 571-574.

39. Texas Instruments (1990). **Third Generation TMS320C30 User's Guide** (literature number SPRU031).

40. Texas Instruments (1990). **TMS320C30 C Compiler Reference Guide** (literature number SPRU034).

41. Texas Instruments (1988). **TMS320C30 Assembly Language Tools User's Guide** (literature number SPRU035).

42. Allen, J., Hunnicutt, S., and Klatt, D.H. (1987). *From Text to Speech: The MITalk System* (Cambridge U.P., Cambridge UK).

APÊNDICE

REGRAS DE CONVERSÃO ORTOGRÁFICA-FONÉTICA E

PRÉ-PROCESSAMENTO

(LEX)

```
%{
/* ---------------------------- Define categorias ----------------------------*/
%}
%a 2400
%e 2000
%k 1500
%o 4000
%p 2700

vogal [aeiouAEIOU]
consoante [bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ]
fim_de_palavra [\n!" "¡\%¡\(¡\)¡\-¡\,¡\;¡\.\¡\?\"]
sonoro [aeioubdgjlmnvAEIOUBDGJLMNV]
consoante_sonora [bdgjlmnvBDGJLMNV]

a [aA]
b [bB]
c [cC]
d [dD]
e [eE]
f [fF]
g [gG]
h [hH]
i [iI]
j [jJ]
k [kK]
l [lL]
m [mM]
n [nN]
o [oO]
p [pP]
q [qQ]
r [rR]
s [sS]
t [tT]
u [uU]
v [vV]
w [wW]
x [xX]
y [yY]
z [zZ]

%{
struct fon_vogal {
        int vogal;
        char vog_ton;
        };
extern int num_fon;
extern char fonema[250][3];
extern struct fon_vogal fon[250];
#define FONEMA guarda_fonema
#define VOGAL fon[num_vog++].vogal = num_fon - 1; vog_pal++;
#define VOGAL_TONICA fon[num_vog].vogal = num_fon - 1;\
fon[num_vog++].vog_ton = SIM; vog_pal++;
#define COPIA fonema[num_fon][0] = yytext[yyleng-1];\
fonema[num_fon++][1] = '\x0'
#define SIM 1
#define NAO 0
#include "alloc.h"
#include "string.h"
%}
```

```
%%
%{
int num_vog = 0;
char acento = NAO;
int vog_pal = 0;
num_fon = 0;
inicializa();
%}
%{
/* ------------------------------ Vogais ------------------------------------*/

/* 1 --------------- Regras de conversão para a letra 'a' ------------------ */
%}

{a}"~"  {
         FONEMA( "an" );
         VOGAL_TONICA;
         acento = SIM;
        }

{a}"'"  ;

{a}"'"  {
         FONEMA( "a_" );
         VOGAL_TONICA;
         acento = SIM;
        }

{a} {
      FONEMA( "a" );
      VOGAL;
    }

{a}iu  {
        FONEMA( "a" );
        VOGAL;
        FONEMA( "i" );
        VOGAL_TONICA;
        FONEMA( "w" );
        VOGAL;
        acento = SIM;
       }

{a}m{fim_de_palavra} {
                      fon[num_vog-1].vog_ton = SIM;
                      parox(fon[num_vog-1].vogal);
                      FONEMA( "an" );
                      VOGAL;
                      FONEMA( "w" );
                      VOGAL;
                      unput(yytext[yyleng-1]);
                      acento = SIM;
                     }

{a}nh {
        FONEMA( "an" );
        VOGAL;
        FONEMA( "nh" );
       }
```

```
({a}m|{a}{n}){consoante} {
                            FONEMA( "an" );
                            VOGAL;
                            unput(yytext[yyleng-1]);
                           }

{a}m{vogal}{fim_de_palavra} {
                              FONEMA( "an" );
                              VOGAL;
                              FONEMA( "m" );
                              unput(yytext[yyleng-1]);
                              unput(yytext[yyleng-2]);
                             }

{a}m{vogal}{s}{fim_de_palavra} {
                                 FONEMA( "an" );
                                 VOGAL;
                                 FONEMA( "m" );
                                 unput(yytext[yyleng-1]);
                                 unput(yytext[yyleng-2]);
                                 unput(yytext[yyleng-3]);
                                }

{a}n{vogal}{fim_de_palavra} {
                              FONEMA( "an" );
                              VOGAL;
                              FONEMA( "n" );
                              unput(yytext[yyleng-1]);
                              unput(yytext[yyleng-2]);
                             }

{a}n{vogal}{s}{fim_de_palavra} {
                                 FONEMA( "an" );
                                 VOGAL;
                                 FONEMA( "n" );
                                 unput(yytext[yyleng-1]);
                                 unput(yytext[yyleng-2]);
                                 unput(yytext[yyleng-3]);
                                }

{a}"^"m {
          FONEMA( "an" );
          VOGAL_TONICA;
          FONEMA( "m" );
          acento = SIM;
         }

{a}"^"n {
          FONEMA( "an" );
          VOGAL_TONICA;
          FONEMA( "n" );
          acento = SIM;
         }

{a}r{fim_de_palavra} {
                       FONEMA( "a_" );
                       VOGAL_TONICA;
                       FONEMA( "r" );
                       acento = SIM;
                       unput(yytext[yyleng-1]);
                      }
```

```
%{
/* 2 ---------------- Regras de conversão para a letra 'e' ---------------- */
%}

E"'"" " {
          FONEMA( "eh" );
          unput(yytext[yyleng-1]);
         }

e"'"" " {
          FONEMA( "eh" );
          unput(yytext[yyleng-1]);
          if(strcmp(fonema[num_fon-1], " ") && strcmp(fonema[num_fon-1], ",")){
               VOGAL_TONICA;
               acento = SIM;
          }
         }

^E ;

e {
   FONEMA( "e" );
   VOGAL;
  }

E {
   if(pauant(fonema[num_fon-1])){
        FONEMA( "e" );
        VOGAL;
  }
   else{
        FONEMA( "eh" );
        VOGAL_TONICA;
        acento = SIM;
   }
  }

e"'" ;

{e}h {
       FONEMA( "eh" );
       VOGAL_TONICA;
       acento = SIM;
      }

{e}nh {
        FONEMA( "en" );
        VOGAL;
        FONEMA( "nh" );
       }

{e}m({consoante}|{fim_de_palavra}) ;

{e}n{consoante} {
                   FONEMA( "en" );
                   VOGAL;
                   unput(yytext[yyleng-1]);
                  }
```

```
{e}"^" {
         FONEMA( "e" );
         VOGAL_TONICA;
         acento = SIM;
        }

{e}"^"n{vogal} {
                  FONEMA( "e" );
                  VOGAL_TONICA;
                  FONEMA( "n" );
                  acento = SIM;
                  unput(yytext[yyleng-1]);
                 }

e'n ;

{e}"'"m ;

{e}"^"n {
          FONEMA( "en" );
          VOGAL_TONICA;
          acento = SIM;
         }

{e}[\.","] ;

{e}{fim_de_palavra} {
                       FONEMA( "y" );
                       VOGAL;
                       unput(yytext[yyleng-1]);
                      }

er{fim_de_palavra} {
                      FONEMA( "e" );
                      VOGAL_TONICA;
                      FONEMA( "r" );
                      acento = SIM;
                      unput(yytext[yyleng-1]);
                     }

{e}s[\.","] ;

{e}s{fim_de_palavra} {
                        FONEMA( "y" );
                        VOGAL;
                        unput(yytext[yyleng-1]);
                        unput('s');
                       }


%{
/* 3 ---------- Regras de conversão para a letra 'i' ---------- */
%}

{i}nh {
        FONEMA( "in" );
        VOGAL;
        FONEMA( "nh" );
       }
```

```
{i}[mn]{consoante} {
                    FONEMA( "in" );
                    VOGAL;
                    unput(yytext[yyleng-1]);
                   }

{i}m{fim_de_palavra} {
                      FONEMA( "in" );
                      VOGAL_TONICA;
                      acento = SIM;
                      unput(yytext[yyleng-1]);
                     }

{i}"'" {
        FONEMA( "i" );
        VOGAL_TONICA;
        acento = SIM;
       }

{i}"'"[mn]{consoante} {
                       unput(yytext[yyleng-1]);
                       FONEMA( "in" );
                       VOGAL_TONICA;
                       acento = SIM;
                      }

{i} {
     FONEMA( "y" );
     VOGAL;
    }

{i}r{fim_de_palavra} {
                      FONEMA( "i" );
                      VOGAL_TONICA;
                      FONEMA( "r" );
                      acento = SIM;
                      unput(yytext[yyleng-1]);
                     }

{i}{fim_de_palavra} {
                     unput(yytext[yyleng-1]);
                     if((fon[num_vog-1].vogal != num_fon - 1) && !acento){
                         unput(0x27);
                         unput('i');
                     }
                     else{
                         FONEMA( "y" );
                         VOGAL;
                     }
                    }


%{
/* 4 ---------- Regras de conversão para a letra 'o' ---------- */
%}

^O :
```

```
o {
   FONEMA( "o" );
   VOGAL;
  }
" "O {
       unput('h');
       unput('o');
       unput(' ');
     }

O {
   if(pauant(fonema[num_fon-1])){
       FONEMA( "o" );
       VOGAL;
   }
   else{
       FONEMA( "oh" );
       VOGAL_TONICA;
       acento = SIM;
   }
  }

{o}"'" ;

{o}h {
      FONEMA( "oh" );
      VOGAL_TONICA;
      acento = SIM;
     }

{o}"^" {
        FONEMA( "o" );
        VOGAL_TONICA;
        acento = SIM;
       }

{o}"~" {
        FONEMA( "on" );
        VOGAL_TONICA;
        acento = SIM;
       }

{o}nh {
       FONEMA( "on" );
       VOGAL;
       FONEMA( "nh" );
      }

{o}m({consoante}|{fim_de_palavra}) ;

{o}[mn]{consoante} {
                    FONEMA( "on" );
                    VOGAL;
                    unput(yytext[yyleng-1]);
                   }

{o}[\.","] ;
```

```
{o}{fim_de_palavra} {
                        FONEMA( "w" );
                        VOGAL;
                        unput(yytext[yyleng-1]);
                       }

oo unput('o');
or{fim_de_palavra} {
                        FONEMA( "o" );
                        VOGAL_TONICA;
                        FONEMA( "r" );
                        acento = SIM;
                        unput(yytext[yyleng-1]);
                       }

{o}s[\.","] ;

os{fim_de_palavra} {
                        FONEMA( "w" );
                        VOGAL;
                        unput(yytext[yyleng-1]);
                        unput('s');
                       }


%{
/* 5 ---------- Regras de conversão para a letra 'u' ---------- */
%}

{u}nh {
        FONEMA( "un" );
        VOGAL;
        FONEMA( "nh" );
       }

{u}[mn]{consoante} {
                        FONEMA( "un" );
                        VOGAL;
                        unput(yytext[yyleng-1]);
                       }
{u}m{fim_de_palavra} {
                          FONEMA( "un" );
                          VOGAL_TONICA;
                          acento = SIM;
                          unput(yytext[yyleng-1]);
                         }
{u}"'" {
          FONEMA( "u" );
          VOGAL_TONICA;
          acento = SIM;
         }

{u} {
      FONEMA( "w" );
      VOGAL;
     }
```

```
%{
/* 6 ---------- Regras de conversão para a letra 'c' ---------- */
%}

{c}[aoOurInt] {
                FONEMA( "k" );
                unput(yytext[yyleng-1]);
              }

{c}[eEi] {
           FONEMA( "s" );
           unput(yytext[yyleng-1]);
         }

{c}h FONEMA( "x" );

{c}"," FONEMA( "s" );


%{
/* 7 ---------- Regras de conversão para a letra 'g' ---------- */
%}

{g}[eEi] {
           FONEMA( "j" );
           unput(yytext[yyleng-1]);
         }

{g}u[eEi] {
            FONEMA( "g" );
            unput(yytext[yyleng-1]);
          }

{g}u\" {
         FONEMA( "g" );
         FONEMA( "u" );
         VOGAL;
       }


%{
/* 8 ---------- Regras de conversão para a letra 'h' ---------- */
%}

" "{h} unput(' ');

","{h} unput(',');

"."{h} unput('.');


%{
/* 9 ---------- Regras de conversão para a letra 'l' ---------- */
%}

{l}{fim_de_palavra} {
                      FONEMA( "w" );
                      VOGAL;
                      unput(yytext[yyleng-1]);
                    }
```

```
{l}h {
       FONEMA( "lh" );
     }

{l}{vogal} {
             FONEMA( "l" );
             unput(yytext[yyleng-1]);
           }

{l} {
     FONEMA( "w" );
     VOGAL;
    }


%{
/* 10 ---------------- Regras de conversão para a letra 'n' ---------------- */
%}

{n}h {
       FONEMA( "nh" );
     }


%{
/* 11 ---------------- Regras de conversão para a letra 'q' ---------------- */
%}

q

{q}u[eEi] {
            FONEMA( "k" );
            unput(yytext[yyleng-1]);
          }

{q}[a-zA-Z] {
              FONEMA( "k" );
              unput(yytext[yyleng-1]);
            }

{q}u\" {
         FONEMA( "k" );
         FONEMA( "w" );
         VOGAL;
       }


%{
/* 12 ---------------- Regras de conversão para a letra 'r' ---------------- */
%}

"=>"{r} ;

^{r} ;

"("r ;

rr {
    FONEMA( "rr" );
   }
```

```
[" "]+{r} {
            unput('r');
            unput('>');
            unput('=');
            unput(' ');
           }

","[" "]*{r} {
               unput('r');
               unput('>');
               unput('=');
               unput(' ');
               unput(',');
              }


%{
/* 13 ---------- Regras de conversão para a letra 's' ---------- */
%}

^{s}[a-zA-Z] {
               unput(yytext[yyleng-1]);
               FONEMA( "s" );
              }

[" "]+{s} {
            unput(',');
            unput('c');
            unput(' ');
           }

[","][" "]*{s} {
                 unput(',');
                 unput('c');
                 unput(' ');
                 unput(',');
                }

{a}"'"s{vogal} {
                 unput(yytext[yyleng-1]);
                 unput('z');
                 unput(0x27);
                 unput('a');
                }

{a}s{vogal} {
              unput(yytext[yyleng-1]);
              FONEMA( "a" );
              VOGAL;
              FONEMA( "z" );
             }


{e}"'"s{vogal} {
                 unput(yytext[yyleng-1]);
                 unput('z');
                 unput(0x27);
                 unput('e');
                }
```

```
es{vogal} {
            unput(yytext[yyleng-1]);
            FONEMA( "e" );
            VOGAL;
            FONEMA( "z" );
           }

Es{vogal} :

{e}hs{vogal} {
               unput(yytext[yyleng-1]);
               unput('z');
               unput('E');
              }

{i}"'"s{vogal} {
                 unput(yytext[yyleng-1]);
                 unput('z');
                 unput(0x27);
                 unput('i');
                }

{i}s{vogal} {
              unput(yytext[yyleng-1]);
              FONEMA( "y" );
              VOGAL;
              FONEMA( "z" );
             }

{o}"'"s{vogal} {
                 unput(yytext[yyleng-1]);
                 unput('z');
                 unput(0x27);
                 unput('o');
                }

os{vogal} {
            unput(yytext[yyleng-1]);
            FONEMA( "o" );
            VOGAL;
            FONEMA( "z" );
           }

Os{vogal} :

{o}hs{vogal} {
               unput(yytext[yyleng-1]);
               unput('z');
               unput('O');
              }


{u}"'"s{vogal} {
                 unput(yytext[yyleng-1]);
                 unput('z');
                 unput(0x27);
                 unput('u');
                }
```

```
{u}s{vogal} {
                unput(yytext[yyleng-1]);
                FONEMA( "w" );
                VOGAL;
                FONEMA( "z" );
                }

{s}[" "¦\n]{sonoro} {
                        unput(yytext[yyleng-1]);
                        unput(' ');
                        FONEMA( "z" );
                       }

{s}[" "¦\n]h{vogal} {
                        unput(yytext[yyleng-1]);
                        unput(' ');
                        FONEMA( "z" );
                       }

{s}{consoante_sonora} {
                          unput(yytext[yyleng-1]);
                          FONEMA( "z" );
                         }

{s}ce {
        unput(yytext[yyleng-1]);
        FONEMA( "s" );
       }

{s}ci {
        unput(yytext[yyleng-1]);
        FONEMA( "s" );
       }

ss FONEMA( "s" );

{s}h FONEMA( "x" );


%{
/* 14 ---------- Regras de conversão para a letra 'x' ---------- */
%}

{x}[cpt] {
           FONEMA( "s" );
           unput(yytext[yyleng-1]);
           }

{e}x{vogal} {
              FONEMA( "e" );
              VOGAL;
              if(pauant(fonema[num_fon-2]))
                   FONEMA( "z" );
              else
                   FONEMA( "x" );
              unput(yytext[yyleng-1]);
             }
```

```
%{
/* 15 ---------- Regras de conversão para a letra 'z' ---------- */
%}

{a}z{fim_de_palavra} {
                        FONEMA( "a_" );
                        VOGAL_TONICA;
                        acento = SIM;
                        unput(yytext[yyleng-1]);
                        unput('s');
                       }

ez{fim_de_palavra} {
                      FONEMA( "e" );
                      VOGAL_TONICA;
                      acento = SIM;
                      unput(yytext[yyleng-1]);
                      unput('s');
                     }

Ez{fim_de_palavra} ;

{e}hz{fim_de_palavra} {
                         FONEMA( "eh" );
                         VOGAL_TONICA;
                         acento = SIM;
                         unput(yytext[yyleng-1]);
                         unput('s');
                        }

{i}z{fim_de_palavra} {
                        FONEMA( "i" );
                        VOGAL_TONICA;
                        acento = SIM;
                        unput(yytext[yyleng-1]);
                        unput('s');
                       }

oz{fim_de_palavra} {
                      FONEMA( "o" );
                      VOGAL_TONICA;
                      acento = SIM;
                      unput(yytext[yyleng-1]);
                      unput('s');
                     }

Oz{fim_de_palavra} ;

{o}hz{fim_de_palavra} {
                         FONEMA( "oh" );
                         VOGAL_TONICA;
                         acento = SIM;
                         unput(yytext[yyleng-1]);
                         unput('s');
                        }

{u}z{fim_de_palavra} {
                        FONEMA( "u" );
                        VOGAL_TONICA;
                        acento = SIM;
                        unput(yytext[yyleng-1]);
                        unput('s');
                       }
```

```
%{
/* 16 ---------------------- Sinais de pontuação ---------------------------- */
%}

[" "]+ {
      FONEMA( " " );
      if(!acento){
         if(vog_pal > 1){
             if((fon[num_vog-2].vogal==(fon[num_vog-3].vogal+1))
                          && ((!strcmp(fonema[fon[num_vog-2].vogal],"y"))
                          || (!strcmp(fonema[fon[num_vog-2].vogal],"w")))
                          && ((!strcmp(fonema[fon[num_vog-3].vogal],"a"))
                          || (!strcmp(fonema[fon[num_vog-3].vogal],"e"))
                          || (!strcmp(fonema[fon[num_vog-3].vogal],"o"))
                          || (!strcmp(fonema[fon[num_vog-3].vogal],"w")))){
                 parox(fon[num_vog-3].vogal);
                 fon[num_vog-3].vog_ton = SIM;
             }
             else{
                 parox(fon[num_vog-2].vogal);
                 fon[num_vog-2].vog_ton = SIM;
             }
         }
      }
      acento = NAO;
      vog_pal = 0;
      }

[" "]*[\n][" "]+ ;

[" "]+[\n][" "]* unput('\n');

\n {
     FONEMA( " " );
     if(!acento){
         if(vog_pal > 1){
             if((fon[num_vog-2].vogal==(fon[num_vog-3].vogal+1))
                          && ((!strcmp(fonema[fon[num_vog-2].vogal],"y"))
                          || (!strcmp(fonema[fon[num_vog-2].vogal],"w")))
                          && ((!strcmp(fonema[fon[num_vog-3].vogal],"a"))
                          || (!strcmp(fonema[fon[num_vog-3].vogal],"e"))
                          || (!strcmp(fonema[fon[num_vog-3].vogal],"o"))
                          || (!strcmp(fonema[fon[num_vog-3].vogal],"w")))){

                 parox(fon[num_vog-3].vogal);
                 fon[num_vog-3].vog_ton = SIM;
             }
             else{
                 parox(fon[num_vog-2].vogal);
                 fon[num_vog-2].vog_ton = SIM;
             }
         }
     }
     }

\"[","] ;

[","]+[" "]*"-"[" "]* ;

[" "]*[","]*[" "]*"(" ;
```

```
")"[" "]*[","]*[" "]* ;

[" "]*","[" "]+ ;

[" "]+","[" "]* unput(',');

"," {
            FONEMA( "," );
            if(!acento){
                if(vog_pal > 1){
                    if((fon[num_vog-2].vogal==(fon[num_vog-3].vogal+1))
                            && ((!strcmp(fonema[fon[num_vog-2].vogal],"y"))
                            || (!strcmp(fonema[fon[num_vog-2].vogal],"w")))
                            && ((!strcmp(fonema[fon[num_vog-3].vogal],"a"))
                            || (!strcmp(fonema[fon[num_vog-3].vogal],"e"))
                            || (!strcmp(fonema[fon[num_vog-3].vogal],"o"))
                            || (!strcmp(fonema[fon[num_vog-3].vogal],"w")))){
                        parox(fon[num_vog-3].vogal);
                        fon[num_vog-3].vog_ton = SIM;
                    }
                    else{
                        parox(fon[num_vog-2].vogal);
                        fon[num_vog-2].vog_ton = SIM;
                    }
                }
            }
            acento = NAO;
            vog_pal = 0;
            }

[" "]*"."[" "]+ ;

\"["."] ;

[" "]+"."[" "]* unput('.');

"." {
        FONEMA( "." );
        if(!acento){
            if(vog_pal > 1){

                if((fon[num_vog-2].vogal==(fon[num_vog-3].vogal+1))
                            && ((!strcmp(fonema[fon[num_vog-2].vogal],"y"))
                            || (!strcmp(fonema[fon[num_vog-2].vogal],"w")))
                            && ((!strcmp(fonema[fon[num_vog-3].vogal],"a"))
                            || (!strcmp(fonema[fon[num_vog-3].vogal],"e"))
                            || (!strcmp(fonema[fon[num_vog-3].vogal],"o"))
                            || (!strcmp(fonema[fon[num_vog-3].vogal],"w")))){
                    parox(fon[num_vog-3].vogal);
                    fon[num_vog-3].vog_ton = SIM;
                }
                else{
                    parox(fon[num_vog-2].vogal);
                    fon[num_vog-2].vog_ton = SIM;
                }
        }
    }
    acento = NAO;
    vog_pal = 0;
    }
```

```
[" "]+[\n]*[" "]*"-" unput(',');

"-" unput(' ');

\"[" "]+ unput(' ');


%{
/* 17 ------------------------ Algarismos ---------------------------------- */
%}

0 entrada("zEro");

[0-9]+ algarismo(0);

-[0-9]+ {
          yyleng--;
          algarismo(1);
          entrada("menos ");
         }

{c}{r}"$"[0-9]+ {
                   yyleng-=3;
                   entrada(" cruzeiros ");
                   algarismo(3);
                  }

{c}{r}"$ "[0-9]+ {
                    yyleng-=4;
                    entrada(" cruzeiros ");
                    algarismo(4);
                   }

[0-9]+","[0-9]+ {
                   yyleng--;
                   entrada(" vi'rgula ");
                   algarismo(0);
                  }


-[0-9]+","[0-9] {
                    yyleng-=2;
                    algarismo(1);
                    entrada("menos ");
                    entrada(" vi'rgula ");
                   }

[0-9]+h {
          yyleng-=1;
          entrada(" hOras ");
          algarismo(0);
         }

[0-9]+h[0-9]+ {
                  yyleng--;
                  entrada(" ");
                  algarismo(0);
                 }
```

```
%{
/*  ---------------------------- Sem conversão ------------------------------ */
%}

[a-zA-Z] {
          COPIA;
          }


%{
/* 18 ------------------------ Abreviaturas ------------------------------- */
%}

{a}v"." entrada("avenida");

cm{fim_de_palavra} {
                     unput(yytext[yyleng-1]);
                     entrada("centi'metros");
                   }

{d}r"." entrada("doutor");

{d}ra"." entrada("doutora");

{e}{t}{c}"." entrada("etcehtra");
{j}r"." entrada("ju'nior");

kg entrada("quilos");

km entrada("quilo^metros");

m entrada("mEtros");

mm entrada("mili'metros");

{p}rof"." entrada("professor");

r"." entrada("rua");

s entrada("segundos");

"%" entrada(" porcento");

{s}r"." entrada("senhor");

{s}ra"." entrada("senhohra");


%{
/* 19 ---------------------------- Siglas ------------------------------- */
%}

[A-Z]+{fim_de_palavra} {
                         unput(yytext[yyleng-1]);
                         sigla();
                       }

%%
```

```
pauant(ant)
char * ant;
{
     if(!strcmp(ant, " ") || !strcmp(ant, ",")
        || !strcmp(ant, ".") || !num_fon)
     return(1);
     else
          return(0);
}

guarda_fonema(fon)
char *fon;
{
     strcpy(fonema[num_fon], fon);
     num_fon++;
}

parox(num_vog)
int num_vog;
{
     int temp;

     temp = num_fon;
     num_fon = num_vog;
     if(!strcmp(fonema[num_vog], "a")){
          guarda_fonema("a_");
          num_fon = temp;
          return;
     }
     if(!strcmp(fonema[num_vog], "y")){
          guarda_fonema("i");
          num_fon = temp;
          return;
     }
     if(!strcmp(fonema[num_vog], "w")){
          guarda_fonema("u");
          num_fon = temp;
          return;
     }
     num_fon = temp;
}

algarismo(alg)
int alg;
{
     switch(yyleng){
          case 1:
               unidade(yytext[alg+0]);
               break;
          case 2:
               dezena(yytext[alg+0], yytext[alg+1]);
               break;
          case 3:
               centena(yytext[alg+0], yytext[alg+1], yytext[alg+2]);
               break;
          case 4:
               milhar(yytext[alg+0], yytext[alg+1], yytext[alg+2],
               yytext[alg+3]);
               break;
```

```
        case 5:
            dez_mil(yytext[alg+0], yytext[alg+1], yytext[alg+2],
            yytext[alg+3], yytext[alg+4]);
            break;
        case 6:
            cen_mil(yytext[alg+0], yytext[alg+1], yytext[alg+2],
            yytext[alg+3], yytext[alg+4], yytext[alg+5]);
            break;
        case 7:
            milhao(yytext[alg+0], yytext[alg+1], yytext[alg+2],
            yytext[alg+3], yytext[alg+4], yytext[alg+5], yytext[alg+6]);
            break;
        case 8:
            dez_milhao(yytext[alg+0], yytext[alg+1], yytext[alg+2],
            yytext[alg+3], yytext[alg+4], yytext[alg+5], yytext[alg+6],
            yytext[alg+7]);
            break;
        case 9:
            cen_milhao(yytext[alg+0], yytext[alg+1], yytext[alg+2],
            yytext[alg+3], yytext[alg+4], yytext[alg+5], yytext[alg+6],
            yytext[alg+7], yytext[alg+8]);
            break;
    }
}

entrada(alg)
char *alg;
{
    int i;
    for(i=strlen(alg)-1; i>=0; i--)
        unput(alg[i]);
}

unidade(num)
char num;
{
    switch(num){
        case '0':
            return(0);
        case '1':
            entrada("um");
            return(1);
        case '2':
            entrada("dois");
            return(1);
        case '3':
            entrada("treys");
            return(1);
        case '4':
            entrada("quatro");
            return(1);
        case '5':
            entrada("cinco");
            return(1);
        case '6':
            entrada("seis");
            return(1);
        case '7':
            entrada("sehte");
            return(1);
```

```
        case '8':
            entrada("oito");
            return(1);
        case '9':
            entrada("nohve");
            return(1);
    }
}

dezena(dez, uni)
char dez, uni;
{
    if(dez == '0' && uni == '0')
        return(0);
    if(dez != '1')
        if(unidade(uni) && (dez != '0'))
            entrada(" e ");
    switch(dez){
        case '1':
            switch(uni){
                case '0':
                    entrada("dehz");
                    return(1);
                case '1':
                    entrada("onze");
                    return(1);
                case '2':
                    entrada("doze");
                    return(1);
                case '3':
                    entrada("treze");
                    return(1);
                case '4':
                    entrada("catorze");
                    return(1);
                case '5':
                    entrada("quinze");
                    return(1);
                case '6':
                    entrada("dezesseis");
                    return(1);
                case '7':
                    entrada("dezessehte");
                    return(1);
                case '8':
                    entrada("dezoito");
                    return(1);
                case '9':
                    entrada("dezenohve");
                    return(1);
            }
        case '2':
            entrada("vinte");
            break;
        case '3':
            entrada("trinta");
            break;
        case '4':
            entrada("quarenta");
            break;
```

```
        case '5':
            entrada("cinqu\"enta");
            break;
        case '6':
            entrada("sessenta");
            break;
        case '7':
            entrada("setenta");
            break;
        case '8':
            entrada("oitenta");
            break;
        case '9':
            entrada("noventa");
            break;
    }
}

centena(cen, dez, uni)
char cen, dez, uni;
{
    if(cen == '0' && dez == '0' && uni == '0')
        return(0);
    if(dezena(dez, uni) && (cen != '0'))
        entrada(" e ");
    switch(cen){
        case '1':
            if(dez == '0' && uni == '0')
                entrada("cem");
            else
                entrada("cento");
                return(1);
        case '2':
            entrada("duzentos");
            return(1);
        case '3':
            entrada("trezentos");
            return(1);
        case '4':
            entrada("quatrwcentos");
            return(1);
        case '5':
            entrada("quinhentos");
            return(1);
        case '6':
            entrada("seicentos");
            return(1);
        case '7':
            entrada("sehtycentos");
            return(1);
        case '8':
            entrada("oitwcentos");
            return(1);
        case '9':
            entrada("nohvycentos");
            return(1);
    }
}
```

```
milhar(mil, cen, dez, uni)
char mil, cen, dez, uni;
{
    if(mil == '0' && cen == '0' && dez == '0' && uni == '0')
        return(0);
    if(centena(cen, dez, uni) && (cen == '0') ! (dez == '0' && uni == '0')
                              && (mil != '0'))
        entrada(" e ");
        entrada("mil ");
    if(mil != '1')
        unidade(mil);
}

dez_mil(dez_mil, mil, cen, dez, uni)
char dez_mil, mil, cen, dez, uni;
{
    if(dez_mil == '0' && mil == '0' && cen == '0' && dez == '0' && uni == '0')
        return(0);
    if((centena(cen, dez, uni)) && (cen == '0') ! (dez == '0'&& uni == '0')
        && (dez_mil != '0'))
        entrada(" e ");
    entrada("mil ");
    dezena(dez_mil, mil);
}

cen_mil(cen_mil, dez_mil, mil, cen, dez, uni)
char cen_mil, dez_mil, mil, cen, dez, uni;
{
    if(dez_mil == '0' && mil == '0' && cen == '0' && dez == '0' && uni == '0'
            && cen_mil == '0')
        return(0);
    if((centena(cen, dez, uni)) && (cen == '0') ! (dez == '0'&& uni == '0'))
        entrada(" e ");
    entrada("mil ");
    centena(cen_mil, dez_mil, mil);
}

milhao(milhao, cent_mil, dez_mil, mil, cen, dez, uni)
char milhao, cent_mil, dez_mil, mil, cen, dez, uni;
{
    if(!cen_mil(cent_mil, dez_mil, mil, cen, dez, uni))
        entrada("de ");
    if(milhao == '1')
        entrada(" milha~o ");
    else
        entrada(" milho~es ");
    unidade(milhao);
}

dez_milhao(dez_milhao, milhao, cent_mil, dez_mil, mil, cen, dez, uni)
char dez_milhao, milhao, cent_mil, dez_mil, mil, cen, dez, uni;
{
    if(!cen_mil(cent_mil, dez_mil, mil, cen, dez, uni))
        entrada("de ");
    entrada(" milho~es ");
    dezena(dez_milhao, milhao);
}
```

```
cen_milhao(cen_milhao, dez_milhao, milhao, cent_mil, dez_mil, mil, cen, dez,
uni)
char cen_milhao, dez_milhao, milhao, cent_mil, dez_mil, mil, cen, dez, uni;
{
    if(!cen_mil(cent_mil, dez_mil, mil, cen, dez, uni))
        entrada("de ");
    entrada(" milho~es ");
    centena(cen_milhao, dez_milhao, milhao);
}

inicializa()
{
    int i;
    for(i=0; i<200; i++){
        fon[i].vogal = 0;
        fon[i].vog_ton = 0;
    }
}

sigla()
{
    int i;
    for(i=yyleng-1; i>=0; i--){
        switch(yytext[i]){
            case 'A':
                entrada( "a" );
                break;
            case 'B':
                entrada( "be^" );
                break;
            case 'C':
                entrada( "ce^" );
                break;
            case 'D':
                entrada( "de^" );
                break;
            case 'E':
                entrada( "e^" );
                break;
            case 'F':
                entrada( "ehfi" );
                break;
            case 'G':
                entrada( "ge^" );
                break;
            case 'H':
                entrada( "aga'" );
                break;
            case 'I':
                entrada( "i'" );
                break;
            case 'J':
                entrada( "johta" );
                break;
            case 'K':
                entrada( "ka'" );
                break;
            case 'L':
                entrada( "ehli" );
                break;
```

```
                case 'M':
                    entrada( "e^mi" );
                    break;
                case 'N':
                    entrada( "e^ni" );
                    break;
                case 'O':
                    entrada( "o'");
                    break;
                case 'P':
                    entrada( "pe^" );
                    break;
                case 'Q':
                    entrada( "que^" );
                    break;
                case 'R':
                    entrada( "ehrri" );
                    break;
                case 'S':
                    entrada( "ehssi" );
                    break;
                case 'T':
                    entrada( "te^" );
                    break;
                case 'U':
                    entrada( "u'" );
                    break;
                case 'V':
                    entrada( "ve^" );
                    break;
                case 'W':
                    entrada( "da'bliu" );
                    break;
                case 'X':
                    entrada( "xis" );
                    break;
                case 'Y':
                    entrada( "i'psolon" );
                    break;
                case 'Z':
                    entrada( "ze^" );
                    break;
            }
        }
    }
```