

Métodos de Codificação de Voz

Uma Introdução

Este artigo vai explicar as principais técnicas utilizadas para a codificação digital de voz. Ao longo do texto vamos procurar relacionar as técnicas apresentadas com os padrões utilizados nos diversos sistemas de telefonia (fixa e celular).

Como não sou especialista no assunto vou, logo de início, pedir desculpas por eventuais incorreções ou omissões. Sugestões para a melhoria e/ou ampliação do texto (dentro do espírito do objetivo) serão sempre bem-vindas. O público-alvo deste artigo são pessoas sem *background* técnico extenso, portanto peço paciência aos leitores que já tenham um conhecimento mais profundo deste tema.

A apresentação vai ficar, sempre que possível, no limite do conceitual. Vamos recorrer à matemática apenas onde isto for imprescindível. Para acompanhar o texto basta um bom conhecimento da matemática e física do ensino médio (funções e eletromagnetismo).

Na elaboração deste artigo usei apenas referências disponíveis na Internet – ver lista no final, mais algumas coisa que ainda me lembro do tempo da faculdade de engenharia elétrica. Aqueles que desejarem se aprofundar mais no assunto podem, a partir das referências citadas, encontrar indicações de outros sites e livros sobre o tema.

Porque codificar?

Codificação digital de sinais de voz é um dos tópicos de uma categoria mais geral de problemas: *digital signal processing* (processamento digital de sinais). Nesta categoria existem inúmeras aplicações, entre elas:

- ❑ *Comerciais* – áudio e vídeo de alta fidelidade, TV, rádio, telefonia;
- ❑ *Médicas* – Radiografia, ultrassonografia, tomografia computadorizada, tomografia por emissão de pósitrons, ressonância magnética nuclear;
- ❑ *Militares* – RADAR, SONAR.

O problema comum a todas estas aplicações é que a capacidade dos meios de transmissão e/ou armazenamento dos dados é finita, e precisamos encontrar um meio termo entre duas necessidades antagônicas: diminuir a quantidade de bits necessária para a representação da informação (*encoding*), e manter a capacidade de recuperar a informação original (*decoding*) com um nível de distorção aceitável.

Na Teoria da Informação encontramos a *Lei de Shannon*, que nos diz que a capacidade máxima de transmissão C (em bps) de um canal de comunicação, na presença de ruído, é dada pela expressão:

$$C = B \cdot \log_2(1 + S/N)$$

Onde B representa a banda de passagem do canal (em Hz), e S/N é a *relação sinal-ruído*, obtida pela divisão da potência média do sinal S pela potência média do ruído N no canal. Uma vez escolhido o canal de comunicação (que define a banda de passagem), para melhorar a capacidade de transmissão temos que brincar com a relação sinal-ruído.

Em comunicações digitais, o efeito do ruído no canal é provocar erros de interpretação pelo receptor (trocas de zeros por uns, e vice-versa). Imagine o receptor como um carro em uma estrada. Quanto mais fechadas e freqüentes as curvas na estrada, maior a chance de derrapagem. O carro (receptor) precisa grudar na estrada (sinal), para evitar derrapagens (erros de interpretação).

Para minimizar as “derrapagens” de recepção temos que diminuir a quantidade de transições de estado no sinal transmitido (um para zero e vice-versa – esta taxa de transições por segundo é denominada *baud rate*). Este é o papel das técnicas de modulação, que não vamos discutir aqui. O que nos interessa aqui é que, quanto menor o *bit rate* do sinal a ser modulado, mais simples (e baratas) ficam as técnicas de modulação que precisam ser usadas para atingir o *baud rate* desejado. É aqui que entram as técnicas de codificação do sinal.

Métodos de codificação são algoritmos que reduzem o *bit rate*, “comprimindo” a quantidade original de bits do sinal na transmissão, e “descomprimindo” na recepção. Daqui em diante vamos examinar os métodos de codificação aplicáveis para o problema de transmitir e receber os dados que representam voz humana, em um contexto de aplicações de telefonia.

Famílias de Algoritmos

Os diferentes algoritmos para codificação de voz, que também são conhecidos como CODECs (*enCODers/DECoders*) ou VOCODERS (*VOIce enCODers/decodERS*), podem ser agrupados em três grandes “famílias” genéricas:

- ❑ *Codificação por forma de onda (waveform encoding)* – Não são feitas muitas suposições sobre a natureza do sinal original, e consegue-se excelente qualidade, mas pouca compressão;
- ❑ *Codificação por síntese (synthesis encoding)* – A natureza do sinal (voz) é essencial para obter máxima compressão, embora com sensível perda na qualidade;
- ❑ *Codificação híbrida (hybrid encoding)* – Usa conceitos das duas outras famílias, procurando um balanço entre qualidade e taxa de compressão.

Os sistemas de telefonia fixa, telefonia celular de primeira geração e VoIP (*Voice over IP*) usam codificação por forma de onda. Nos sistemas de telefonia celular de segunda e terceira geração usa-se codificação híbrida no enlace de rádio entre o usuário e a BTS (*base transceiver station*) e no enlace entre as BTSs e a BSC (*base station controller*) e codificação por forma de onda da BSC em diante. Codificação por síntese praticamente só é utilizada em aplicações militares, mas é relativamente popular entre os músicos, para criar efeitos de “voz robótica”.

Qualidade da Voz

Uma vez que mencionamos qualidade da voz como um parâmetro de comparação entre os vários tipos de VOCODERS, precisamos definir melhor o que isto significa.

Embora possamos usar fórmulas analíticas para determinar a “qualidade” de um sistema de comunicação de voz, em termos de distorção entre o sinal recebido e o sinal original, nada disto é útil se o usuário não considerar o resultado aceitável.

O que vale, então, é o princípio geral da qualidade: atender às expectativas do cliente. O que o usuário de telefonia deseja? Duas coisas:

- ❑ *Inteligibilidade* – conseguir compreender o significado associado ao áudio reproduzido (entender as palavras);
- ❑ *Discernibilidade* – conseguir distinguir entre diferentes fontes (identificar a pessoa que fala).

Para avaliar como os usuários percebem estas duas características, foram criados dois métodos estatísticos: ACR (*Absolute Category Rating*) e CCR (*Comparison Category Rating*). Em ambos os casos, grupos de pessoas experimentam o sistema, e dão suas opiniões individuais sobre a qualidade do áudio:

- ❑ *ACR* – Os ouvintes dão opinião a respeito da voz reproduzida, expressa em uma escala variando de 1 a 5 (1=muito ruim, 2=ruim, 3=razoável, 4=bom e 5=excelente). O resultado final é a média aritmética das notas individuais atribuídas pelos ouvintes, denominado *índice MOS* (*Mean Opinion Score*);
- ❑ *CCR* – Os ouvintes comparam a voz original com a voz reproduzida, e dão opinião em uma escala variando de -3 a +3 (-3=muito pior, -2=pior, -1=pouco pior, 0=igual, +1=pouco melhor, +2=melhor e +3=muito melhor). O resultado final também é a média aritmética das notas individuais atribuídas pelos ouvintes, denominado *índice CMOS* (*Comparison Mean Opinion Score*).

Dos dois métodos, o ACR é o mais conhecido e difundido. Como as várias técnicas de codificação se saem neste tipo de teste? O desempenho típico das famílias de VOCODERs, em termos de índice MOS e *bit rate* pós compressão, está apresentado na figura 1.

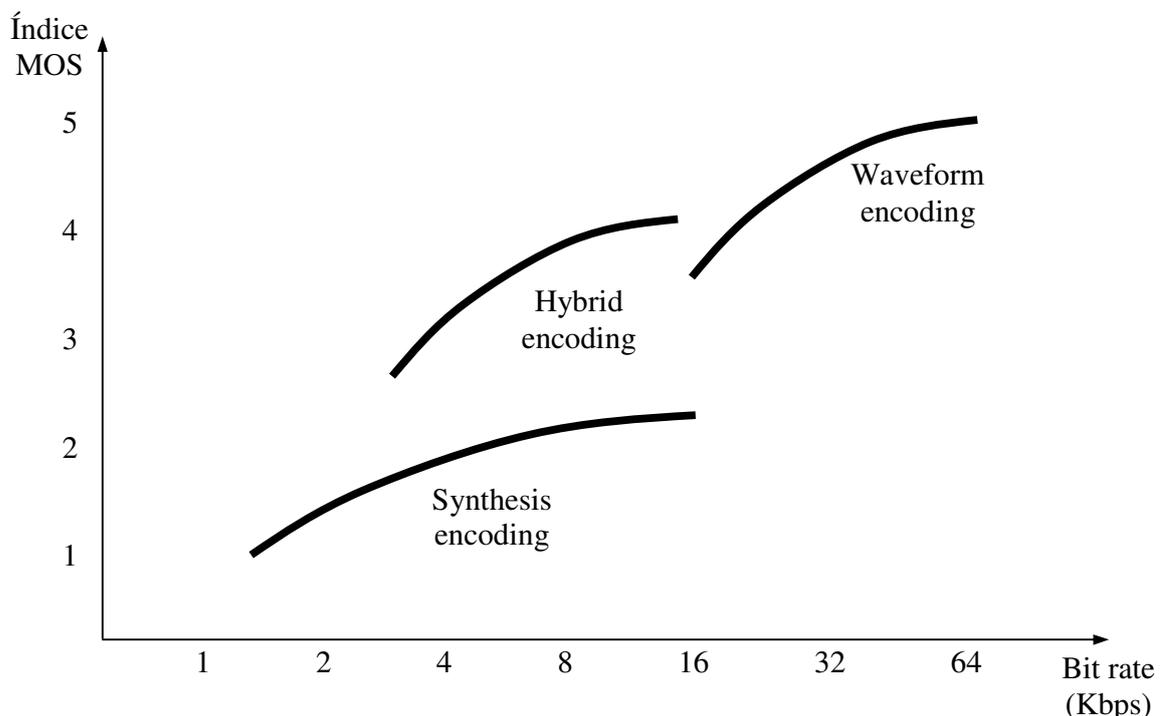


Figura 1 – Desempenho das famílias de VOCODERs

Representação Digital de Sinais Analógicos

Quando tratamos com informação de voz, não podemos fugir do fato que, no final, os sinais são originados e recebidos por mecanismos biológicos (o trato vocal e o aparelho auditivo) que são *analógicos*: a informação é codificada, de forma contínua no tempo, em ondas de pressão do ar.

Nos extremos de qualquer mecanismo para transmissão e recepção de sinais de voz estão os *transdutores* (microfones e alto-falantes), que irão converter sinais analógicos de pressão em sinais analógicos de tensão (ou corrente) elétrica, e vice-versa.

Agora um pouco de matemática. Um sinal qualquer é representado como uma função do tempo $s(t)$. Uma das formas de analisar funções complicadas é representá-las por somas de outras funções mais simples que, dentro de certos limites, produzem o mesmo resultado que a função complicada. Isto é conhecido como expansão em série de funções.

Uma das várias formas possíveis de expansão em série é a *série de Fourier*, que representa a função original por uma soma infinita (e envolvendo números complexos) de funções seno e co-seno. Com alguns algebrismos, podemos simplificar para uma soma infinita de funções co-seno, do tipo:

$$s(t) = \sum A \cdot \cos(\omega t)$$

Não vamos entrar no detalhe de como calcular as amplitudes A (o que envolve cálculo integral). O que nos importa é que, se $s(t)$ for periódica (com frequência angular $\omega_s = 2\pi f$, $f = 1/T$, onde T é o período), somente os termos correspondentes a frequências angulares que sejam múltiplos inteiros da frequência angular de $s(t)$ ($\omega = 0$, $\omega = \pm\omega_s$, $\omega = \pm 2\omega_s$, ...) terão amplitudes A diferentes de zero.

Se $s(t)$ não é periódica, entretanto, podemos encontrar amplitudes A diferentes de zero para qualquer valor real de ω entre menos infinito e mais infinito.

Cada um dos termos do somatório com amplitude A diferente de zero é denominado uma *componente de frequência* de $s(t)$, e contribui para a soma total de acordo com a sua amplitude A .

Se fizermos um gráfico das amplitudes das componentes de frequência de $s(t)$, como função da frequência angular, obtemos uma outra função $S(\omega)$, que representa a contribuição de cada componente de frequência na formação de $s(t)$, e que é conhecida como o *espectro de frequências* de $s(t)$. A função $S(\omega)$ é a *transformada de Fourier* de $s(t)$, e o processo matemático para obter diretamente $S(\omega)$ a partir de $s(t)$ é denominado *transformação de Fourier*. Analogamente, o processo para obter $s(t)$ a partir de $S(\omega)$ é denominado *transformação inversa de Fourier*. A figura 2 ilustra isto.

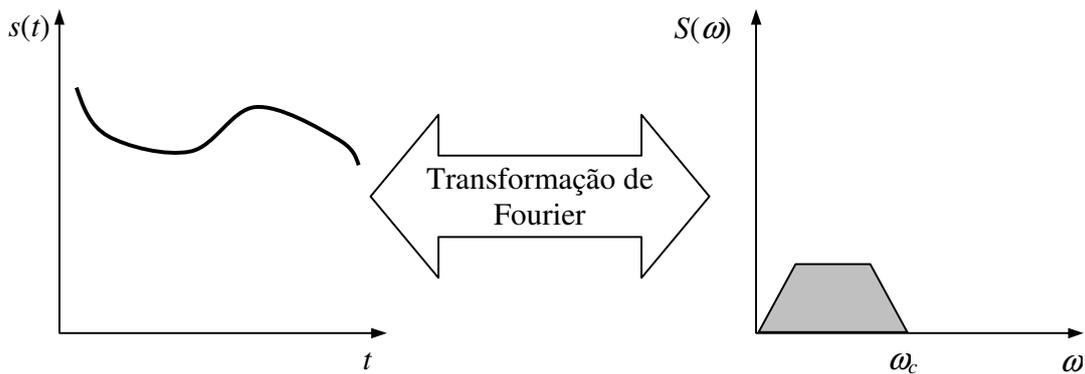


Figura 2 – sinal no tempo e espectro de frequência

Os gráficos apresentados são apenas esquemáticos, para ilustrar os conceitos envolvidos, e não devem ser entendidos como representações matematicamente precisas. Especialmente, a título de simplificação, somente vamos considerar os valores positivos de ω .

Vamos observar o gráfico do espectro de frequência de $s(t)$ apresentado na figura 2. Para todas as frequências maiores que ω_c , $S(\omega)$ assume somente valores nulos. Isto significa que não existem componentes de frequência no espectro de $s(t)$ acima da frequência ω_c . Isto é o mesmo que dizer que $s(t)$ é limitado em faixa na frequência ω_c .

Voltando à Teoria da Informação, encontramos o *teorema de Nyquist*: para qualquer sinal $s(t)$ limitado em faixa na frequência ω_c , o sinal original pode ser recuperado integralmente a partir de amostras discretas de $s(t)$, tomadas com frequência $\omega_a \geq 2 \cdot \omega_c$.

Para entender o significado do teorema, vamos examinar, passo a passo, o que acontece no domínio tempo e no domínio frequência. O processo de amostragem pode ser entendido como a multiplicação, no tempo, do sinal $s(t)$ por uma função *trem de impulsos* $a(t)$, com pulsos de amplitude unitária e frequência angular $\omega_a = 2 \cdot \omega_c$. O resultado é o trem de amostras $p(t)$, conforme mostra a figura 3.

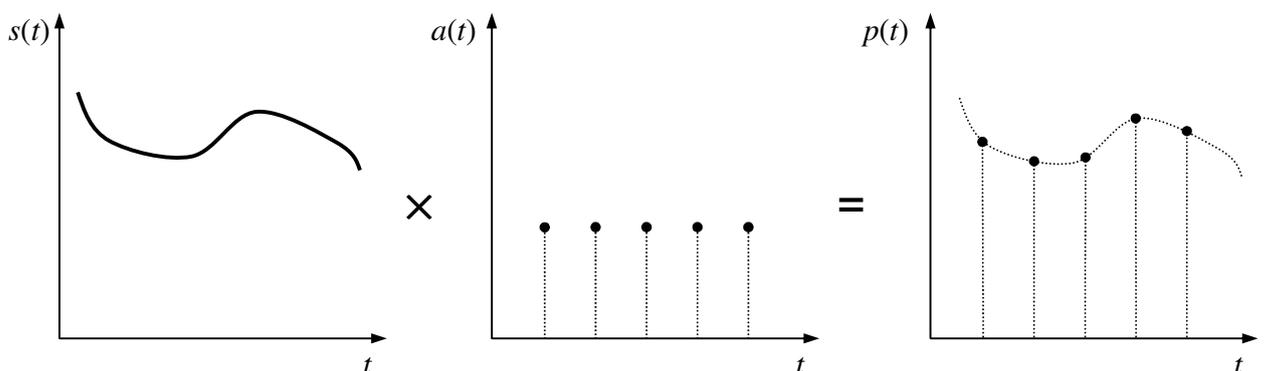


Figura 3 – amostragem

O sinal $p(t)$ é a versão PAM (*pulse-amplitude modulation*) de $s(t)$. Vejamos como fica o espectro de frequência do sinal PAM na figura 4.

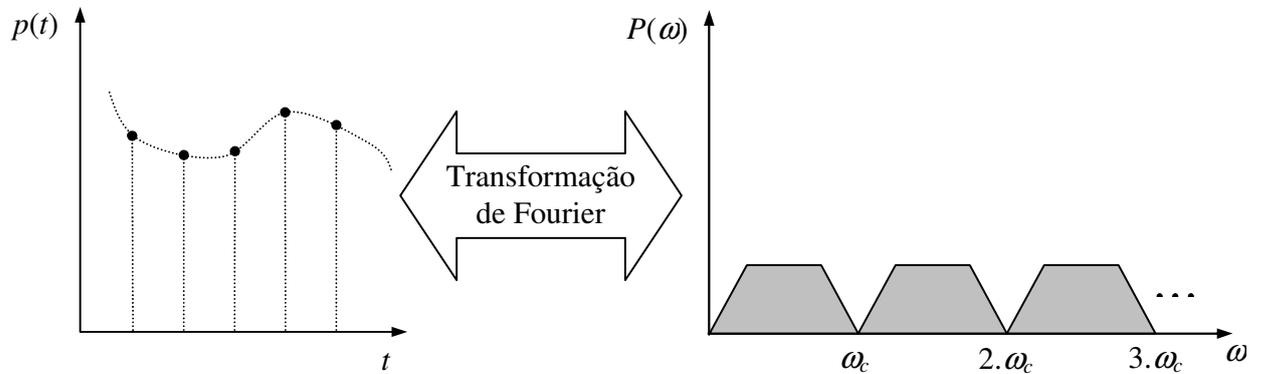


Figura 4 – espectro de frequência do sinal PAM

O que temos são várias “cópias” do espectro de frequências de $s(t)$, localizadas entre as frequências $0, \pm\omega_c, \pm2.\omega_c, \pm3.\omega_c, \dots$ (a parte negativa do espectro de $p(t)$ não é mostrada).

Para recuperar o sinal original, basta passar $p(t)$ por um filtro que isole uma das “cópias” do espectro original. Na figura 4 mostramos o que acontece se a frequência de amostragem for exatamente igual ao dobro da frequência máxima do sinal, como exigido pelo teorema de Nyquist. Se a frequência de amostragem for menor que este valor, ocorre perda de informação, e o sinal original não pode mais ser recuperado sem distorção. Este fenômeno é conhecido como *aliasing*. Se a frequência de amostragem for superior ao mínimo estabelecido pelo teorema de Nyquist, ocorre um “espalhamento” das cópias do espectro original. Esta situação de superamostragem (*oversampling*) é útil, como veremos depois. A figura 5 representa estas situações.

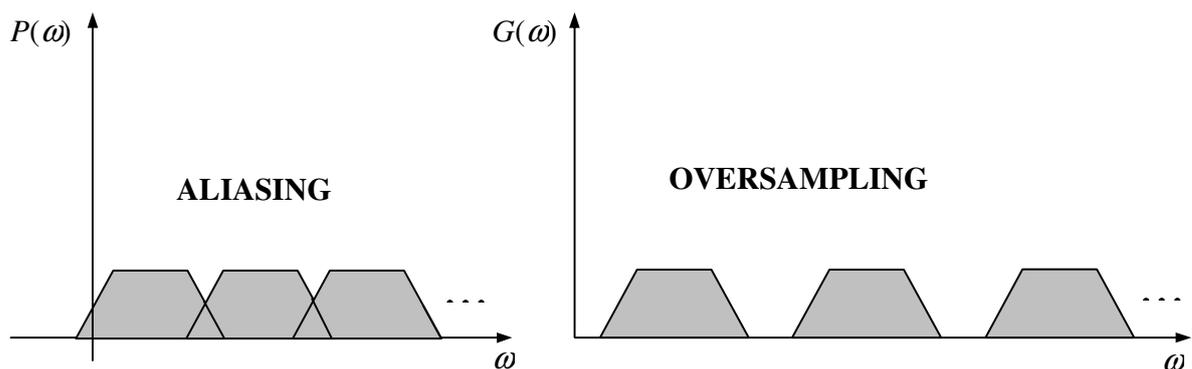


Figura 5 – aliasing e oversampling

O segredo é determinar, de acordo com a natureza da aplicação, a frequência limite ω_c e a frequência de amostragem ω_t . Em sinais de voz existem muitas componentes de frequência

alta, mas estudos determinaram que, para as necessidades de inteligibilidade e discernibilidade em aplicações de telefonia, as componentes de frequência importantes estão contidas na faixa de 500 Hz a 3,5 KHz.

Então, para este tipo de aplicação, vamos forçar o sinal original a ficar limitado em faixa na frequência $f_c=4$ KHz (correspondente a $\omega_c=8.\pi$ Krad/s). Isto pode ser feito passando o sinal de saída do microfone por um filtro passa-baixa, que corta todas as componentes acima de 4 KHz. Este filtro é denominado de *anti-alias* (porque ele vai prevenir a ocorrência de *aliasing* na amostragem). Como é difícil (e caro) construir um filtro analógico perfeito, com atenuação infinita em uma frequência precisa, o que se faz é *oversampling* e depois descarte seletivo das amostras em excesso, num processo conhecido como *dizimação* (*decimation*). O efeito final é de um filtro *anti-alias* digital, com frequência de corte exatamente em 4 KHz.

Resumindo o que vimos até aqui: na transmissão geramos um sinal PAM, e na recepção recuperamos o sinal original, por filtragem simples do sinal PAM. Mas o sinal PAM não é digital. Ele é discreto no tempo, com cada amostra podendo assumir uma faixa contínua de valores. Para transformar o sinal PAM em um sinal digital, fazemos um processo chamado *quantização*.

Para quantizar, primeiro precisamos saber os limites de excursão do sinal PAM gerado na amostragem. Da mesma forma que usamos um filtro para limitar as frequências do sinal a amostrar, usamos um filtro para limitar a amplitude do sinal analógico de entrada, o que torna a excursão máxima do sinal PAM conhecida (dica: não adianta gritar ao telefone, porque o volume máximo transmitido é limitado). Supondo que o sinal PAM nos dê amostras do valor da tensão elétrica na saída do microfone em função do tempo, e que estas amostras variam entre os valores mínimo (V_{\min}) e máximo (V_{\max}), dividindo o intervalo de excursão do sinal PAM em n sub-intervalos iguais, montamos a tabela 1:

Tabela 1 – Intervalos de quantização		
Núm.	Faixa de Valores	
0	$V_{\min} \leq V < V_{\min} + \Delta$	$\Delta = \frac{V_{\max} - V_{\min}}{n}$
1	$V_{\min} + \Delta \leq V < V_{\min} + 2.\Delta$	
2	$V_{\min} + 2.\Delta \leq V < V_{\min} + 3.\Delta$	
⋮	⋮	
k	$V_{\min} + k.\Delta \leq V < V_{\min} + (k + 1).\Delta$	
⋮	⋮	
$n-3$	$V_{\min} + (n - 3).\Delta \leq V < V_{\min} + (n - 2).\Delta$	
$n-2$	$V_{\min} + (n - 2).\Delta \leq V < V_{\min} + (n - 1).\Delta$	
$n-1$	$V_{\min} + (n - 1).\Delta \leq V \leq V_{\min} + n.\Delta$	

Cada amostra do sinal PAM é comparada com esta tabela, para verificar em qual dos sub-intervalos ela se encaixa. Daí em diante, a amostra passa a ser representada pelo número

binário que identifica o sub-intervalo de quantização onde ela se encaixou. O valor de n (número de sub-intervalos) determina a quantidade de bits necessários para representar cada amostra do sinal PAM.

O problema deste método de representação ocorre na recepção. Tudo que o receptor conhece sobre uma determinada amostra é o número (binário) do sub-intervalo onde ela foi classificada pelo transmissor. O receptor não tem como determinar qual o valor exato da amostra dentro daquele sub-intervalo, então ele faz uma aproximação, assumindo que o valor da amostra recuperada é igual ao valor médio do sub-intervalo ao qual ela pertence. Isto causa uma distorção no sinal PAM recuperado, conhecido como *erro de quantização* (*quantization error*) ou *ruído de quantização* (*quantization noise*).

Para minimizar o erro de quantização basta utilizar um número n de sub-intervalos adequado à natureza da aplicação. Para aplicações de telefonia, $n=256$ (amostras representadas por números binários com 8 bits de tamanho) dá resultado satisfatório. Para aplicações de áudio de alta fidelidade (CDs) usa-se $n=65536$ (amostras representadas por números binários com 16 bits de tamanho).

A figura 6 mostra um diagrama de blocos para o processo de conversão A/D (análogo/digital) e D/A (digital/análogo) que foi descrito.

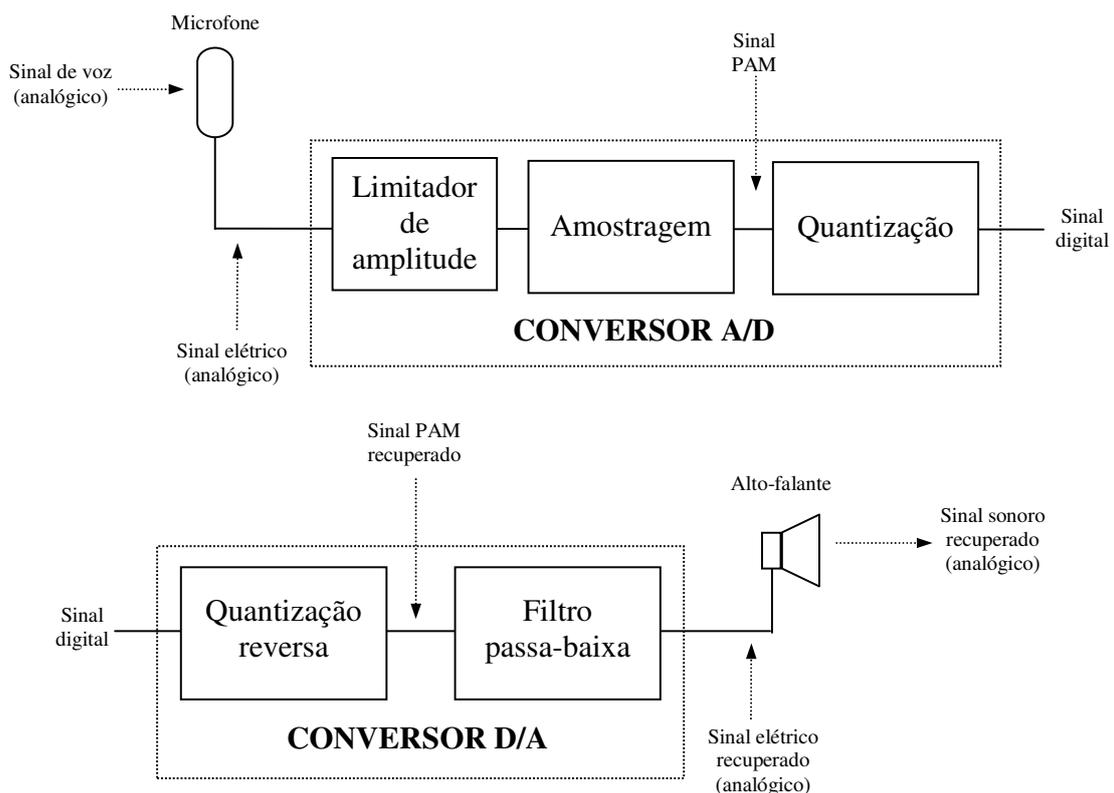


Figura 6 – conversão A/D e D/A

O cálculo do *bit rate* do sinal digital gerado por este processo de conversão é simples. A frequência de amostragem (*sampling rate*) é de 8 KHz (o dobro da frequência de corte de *anti-alias*), portanto o sinal PAM contém 8000 amostras/s, ou, expressando como intervalo de tempo entre duas amostras consecutivas (*sampling interval*), uma amostra a cada 125 μ s (1μ s = 10^{-6} s). Então:

$$\textit{bit rate} = 8000 \text{ amostras/s} \cdot 8 \text{ bits/amostra} = 64000 \text{ bits/s} = \mathbf{64 \text{ Kbps}}$$

Waveform Encoding

As técnicas de *waveform encoding* mapeiam o sinal original no domínio tempo (por isso também são denominadas como técnicas de *time-domain encoding*), usando os bits do sinal digital, que são representações das amplitudes, no tempo, do sinal PAM. Conforme o modo de codificação empregado, estas técnicas produzem *bit rates* de altos a moderados, mas, em contrapartida, obtêm os melhores índices de qualidade (MOS ou CMOS).

PCM

A “mãe” de todas as técnicas de *waveform encoding* é conhecida como PCM (*Pulse Coded Modulation*). Para aplicações de telefonia, o uso do PCM é padronizado na recomendação G.711 da ITU-T (*International Telecommunications Union – Telecommunication standards section*). O termo “recomendação” é ilusório, porque, na verdade, as recomendações da ITU-T são normas.

A diferença do PCM para o processo de conversão A/D e D/A já descrito está no modo de fazer a quantização.

No capítulo anterior, a quantização do sinal PAM foi feita de forma linear (todos os intervalos de quantização com a mesma “largura” – ver tabela 1). Mas o ouvido humano não tem uma curva de resposta dinâmica linear, e sim logarítmica. Isto significa que nossa sensibilidade para perceber diferenças de volume (amplitude) é muito grande para sons de baixa intensidade, e decresce logarítmicamente à medida que a intensidade aumenta. Para termos a sensação que o volume dobrou, a potência do sinal (diretamente proporcional à amplitude) tem de ser multiplicada por 10. É por isso que os audiófilos sempre querem amplificadores com muita potência. Se você achava que isso era bobagem, melhor rever sua posição.

A consequência prática é que os erros de quantização são mais perceptíveis para o ouvinte na parte baixa da escala de quantização, e menos perceptíveis na parte alta. A recomendação G.711 define uma maneira logarítmica para a distribuição das amplitudes dos sub-intervalos de quantização, com sub-intervalos menores na parte baixa da escala, e maiores na parte alta.

Este processo, conhecido como *companding* (*COMPressing and expANDING*) pode ser feito de duas formas, conhecidas como μ -*law* (*mu-law*), usada nos Estados Unidos e Japão, e *A-law*, usada nos demais países. As duas formas são equivalentes, mas a *A-law* exige menos esforço computacional para implementação. O algoritmo, nos dois casos, é simples: primeiro é feita uma quantização linear com um número maior de intervalos (4096 ou 65536), e depois os números binários resultantes desta quantização linear (com 12 ou 16 bits) são transformados um número binário com 8 bits, de acordo com uma função de mapeamento.

Em termos de *bit rate*, os VOCODERs G.711 não oferecem nenhum ganho em relação ao processo linear de digitalização (são os mesmos 64 Kbps). A construção dos VOCODERs G.711 é semelhante à apresentada na figura 6, apenas acrescentando o algoritmo de *companding* no módulo de quantização.

Embora VOCODERs G.711 sejam muito bons em qualidade (índice MOS 4,3), o bit rate gerado é muito elevado para diversas situações, o que limita sua aplicação. Mas este tipo de codificação ainda é o padrão para todo o tráfego de voz digital através das estruturas convencionais de comutação e transmissão nas operadoras de telecomunicações.

DPCM e ADPCM

Observando o comportamento do sinal PCM obtido a partir de sinais de voz, observamos que ele não costuma apresentar variações muito grandes entre duas amostras consecutivas.

Comparando os valores binários que codificam uma amostra e sua antecessora, vemos que a diferença é um número que pode ser codificado com menos de oito bits.

Esta técnica, que é uma variação da modulação delta (*delta modulation* – DM), é conhecida como DPCM (*differential PCM*). O processo de encoding é feito da seguinte forma:

- ❑ O sinal de voz é captado e codificado no formato PCM convencional;
- ❑ O valor binário de cada amostra PCM é passado para dois circuitos, *preditor* e *diferenciador*;
- ❑ O circuito preditor cria um *delay* de um intervalo de amostragem (125 μ s), portanto, na sua saída sempre está o valor binário da amostra anterior;
- ❑ O circuito diferenciador compara os valores binários da amostra corrente e da amostra anterior (na saída do preditor), e calcula a diferença binária entre eles. A saída do diferenciador é o sinal digital a transmitir.

A decodificação segue o processo inverso, como pode ser visto na figura 7.

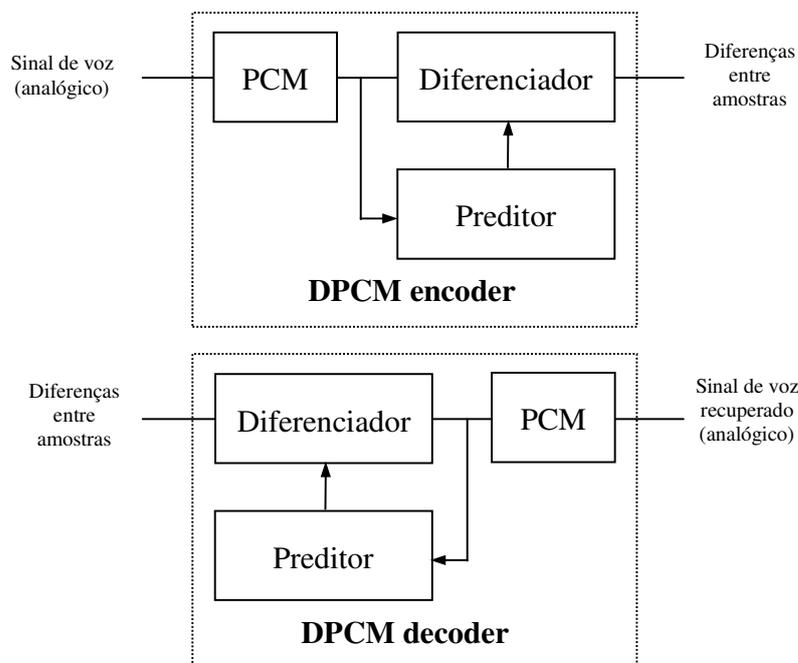


Figura 7 – DPCM encoder e decoder

Embora não tenha encontrado nenhuma referência objetiva, suspeito que as diferenças negativas sejam representadas no formato de complemento de 2.

A qualidade do sinal de voz recuperado é inversamente proporcional à quantidade de bits utilizada para representar as diferenças. Com 4 bits por diferença o índice MOS é 4,1.

O bit rate do sinal digital também dependerá do número de bits usado para representar cada diferença. A tabela 2 apresenta os bit rates associados aos números de bits por diferença possíveis.

Tabela 2 – Bit Rates DPCM	
Bits/Diferença	Bit Rate
7	56 Kbps
6	48 Kbps
5	40 Kbps
4	32 Kbps
3	24 Kbps
2	16 Kbps
1	8 Kbps

Bit Rate=bits/diferença.8000 diferenças/s

A técnica de análise da correlação entre a amostra corrente e a amostra anterior utilizada no DPCM pode ser aprimorada, incorporando o algoritmo de *linear prediction*: durante intervalos curtos de tempo, é possível fazer uma previsão do valor de uma amostra como uma *combinação linear* das suas antecessoras. Isto quer dizer que uma amostra x_n qualquer pode ter o seu valor previsto (\hat{x}_n) calculado como um polinômio das k amostras anteriores:

$$\hat{x}_n = a_1 \cdot x_{n-1} + a_2 \cdot x_{n-2} + a_3 \cdot x_{n-3} + \dots + a_k \cdot x_{n-k}$$

Os coeficientes $a_1, a_2 \dots a_k$ são chamados *coeficientes de adaptação*, e são calculados dinamicamente a partir das k amostras anteriores. Podemos, então, calcular a diferença δ entre o valor real da amostra e o seu valor previsto:

$$\delta = x_n - \hat{x}_n$$

Mantendo o mesmo esquema básico dos VOCODERs DPCM (figura 7), apenas sofisticando o comportamento dos circuitos de predição e diferenciação, fazemos a transmissão das diferenças calculadas entre cada amostra e seu valor previsto. Esta forma de codificação é conhecida como *Adaptive Differential PCM* (ADPCM), e está padronizada pela ITU-T nas recomendações G.721, G.726 e G.727, com bit rates variando de 16 a 40 Kbps.

O índice MOS de VOCODERs DPCM depende do *bit rate*, chegando a 3,9 com *bit rate* de 32 Kbps. As principais aplicações de VOCODERs ADPCM são:

- ❑ “Duplicadores de banda” para canais G.711 convencionais (em um canal de 64 Kbps podem ser transmitidos 2 sinais ADPCM, a 32 Kbps cada);
- ❑ Aplicações de *Voice over IP* (VoIP);
- ❑ Canais de áudio de aplicações de videoconferência.

Características Espectrais da Voz

Para sistemas de telefonia celular, todos os VOCODERs apresentados até agora são dispendiosos demais em termos de *bit rate*. Para entender as técnicas que fundamentam as demais famílias de VOCODERs (*synthesis encoding* e *hybrid encoding*), vamos ter de analisar mais de perto as características espectrais do sinal de voz, e explorar estas características nos algoritmos.

Um modelo matemático simples para simulação do trato vocal, tem os seguintes elementos:

- ❑ Um fole que gera um fluxo de ar (pulmões);
- ❑ Uma membrana que vibra com a passagem do ar (cordas vocais, localizadas na glote);
- ❑ Um tubo, com cerca de 17 cm de comprimento (laringe, faringe e boca).

Os sons produzidos por este aparelho caem, simplificadaamente, em duas categorias: *vocalizados* (*voiced*) e *não vocalizados* (*unvoiced*).

Os sons vocalizados são produzidos pela vibração da membrana em determinada frequência base, que sofre ressonâncias específicas, a depender da forma do tubo (posição da língua, abertura da boca, etc.). Este tipo de som apresenta, durante curtos períodos de tempo, características quase-periódicas. O espectro de frequência deste tipo de sinal apresenta picos característicos em determinadas frequências, denominadas *formantes* (*formants*). Na figura 8 podemos observar um segmento de 30 ms deste tipo de som, e seu espectro de frequência.

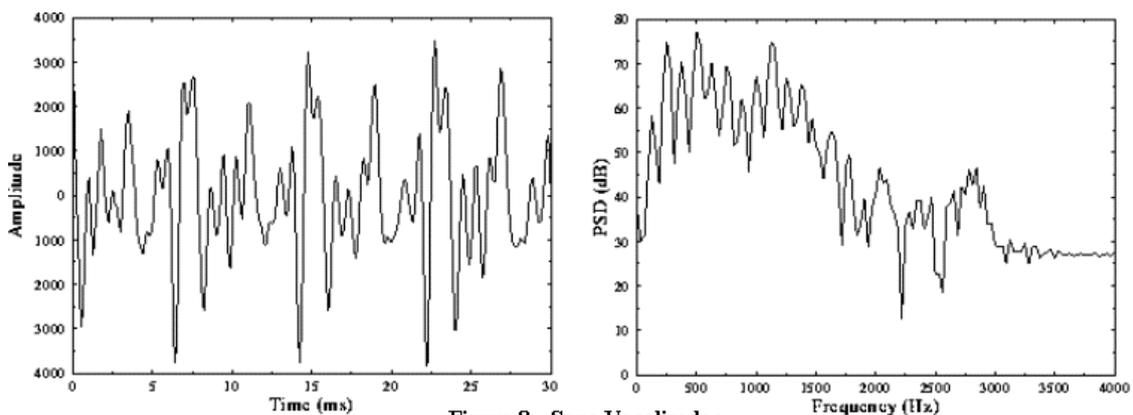


Figura 8 - Sons Vocalizados

Os sons não vocalizados são produzidos pela passagem forçada do ar através de oclusões no tubo (dentes, língua, lábios, etc.), com pouca ou nenhuma vibração da membrana. O resultado é um sinal com características espectrais semelhantes a ruído, como pode ser visto na figura 9.

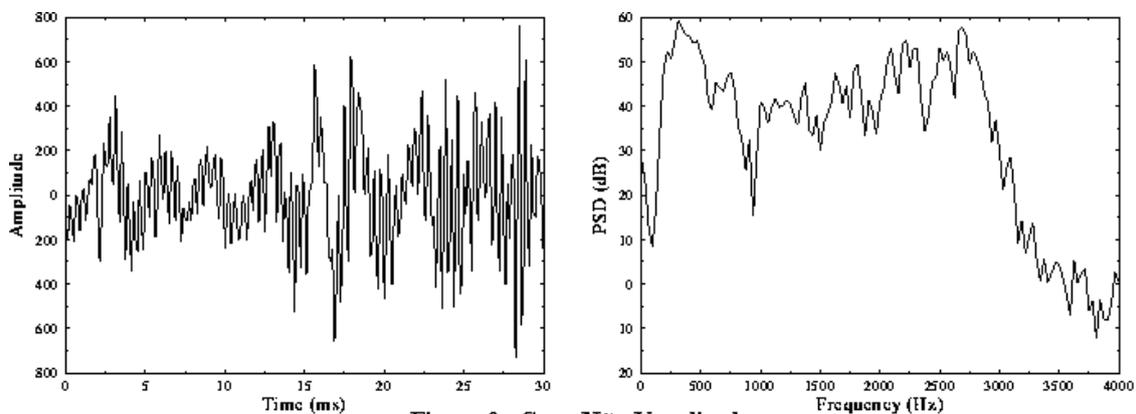


Figura 9 - Sons Não Vocalizados

Synthesis Encoding

Esta técnica (e também a família *hybrid encoding*) só se tornou viável após o brutal aumento de capacidade computacional dos DSPs (*Digital Signal Processors*) a partir da década de 1990. A idéia básica é aproximar o comportamento do trato vocal como um filtro, cujos parâmetros são variáveis no tempo.

A codificação é feita por análise, no domínio freqüência, de pequenos grupos de amostras (*frames*) do sinal PCM. Um *frame* tem, tipicamente de 10 a 20 ms de duração, e contém entre 80 e 160 amostras PCM. Em cada bloco o algoritmo do encoder é:

- ❑ Análise espectral (usando o algoritmo *Fast Fourier Transform* – FFT) para determinar a natureza do *frame* (vocalizado ou não vocalizado), e identificar (se a natureza do *frame* for vocalizada) as freqüências das formantes e os parâmetros para o filtro de simulação das ressonâncias do trato vocal (filtro de síntese);
- ❑ Transmissão dos dados de natureza do *frame*, e, se vocalizado, freqüências das formantes e parâmetros do filtro de síntese.

O decoder, com a informação recebida, executa o seguinte algoritmo:

- ❑ Se o frame for vocalizado,
 - a. Seleciona fonte de sinal de pulsos periódicos, ajustados nas freqüências das formantes, para excitar o filtro de síntese;
 - b. Ajusta os parâmetros do filtro de síntese;
 - c. Transmite para a saída, durante a duração do *frame*, o sinal da fonte de pulsos, através do filtro de síntese;
- ❑ Se o frame for não vocalizado, transmite para a saída sinal de ruído branco durante a duração do *frame*.

Esta categoria de VOCODERs consegue *bit rates* muito baixos, da ordem de 1,2 a 2,4 Kbps, mas, devido à dificuldade em criar um modelo realístico para a simulação do trato vocal, o sinal

recuperado é perceptivelmente artificial (as palavras são inteligíveis, mas a voz é “robotizada”, o que prejudica a discernibilidade).

Aumentar o *bit rate* não melhora a qualidade, porque, neste caso, o *bit rate* é uma consequência do número de variáveis no modelo de simulação do trato vocal. Usar um modelo mais sofisticado aumenta muito o esforço computacional da implementação dos VOCODERs. Do jeito que está, já são necessários DSPs muito potentes para limitar o *delay* de processamento.

Por causa disto, esta família de VOCODERs praticamente só é empregado em aplicações militares, porque, neste caso, *bit rates* baixos importam mais que a qualidade (principalmente por causa dos algoritmos de criptografia que ainda vão ser superpostos à voz codificada), e em música (porque a voz “robótica” é exatamente o que se quer).

Hybrid Encoding

O objetivo desta família de VOCODERs é conseguir um *trade-off* entre as características das famílias de *waveform encoding* e *synthesis encoding*, de forma a conseguir *bit rates* mais baixos, com qualidade razoável, e com esforço computacional moderado. Assim é possível simplificar a construção e diminuir o custo final do produto, o que é importante, se você quer usar o VOCODER “embarcado” em um produto de massa – como aparelhos de telefonia celular.

A idéia é, utilizando *linear prediction* (veja a descrição do ADPCM), simplificar o esforço computacional para encontrar os parâmetros do filtro de síntese e conseguir um sinal de voz recuperado, com qualidade razoável (índice MOS entre 3,7 e 3,9).

O algoritmo de encoding é:

- Geração do sinal PCM e separação das amostras em frames;
- Análise do *frame* no domínio freqüência para determinar uma estimativa para a função de excitação do filtro de síntese;
- Geração de uma simulação do *frame* com a função de excitação estimada;
- Comparação da saída do filtro de síntese com o *frame* original, gerando uma função diferença;
- A função diferença é usada, iterativamente, para aprimorar a estimativa da função de excitação até a função diferença atender o critério desejado;
- Transmissão dos parâmetros da última função de excitação do filtro de síntese para este *frame*.

O decoder usa os dados recebidos para ajustar um gerador da função de excitação, que é passada pelo filtro de síntese para obter o sinal de voz recuperado.

As diferenças entre os vários tipos de VOCODERs desta família está na forma de gerar a função de excitação, e no uso de um único filtro de síntese ou dois filtros de síntese para *short term prediction* (detecção das formantes no frame) e *long term prediction* (detecção de periodicidades entre frames). Os principais tipos (com vários sub-tipos dentro de cada um) são:

- MPE-LPC (Multiple-Pulse Excited Linear Prediction Coding)* – A função de excitação é formada por múltiplos pulsos, com intervalos de freqüência não uniformes entre eles. Utiliza *short-term prediction* e *long-term prediction*;

- ❑ *RPE-LPC (Regular-Pulse Excited Linear Prediction Coding)* – A função de excitação é formada por múltiplos pulsos, com intervalos de frequência uniformes entre eles. Utiliza apenas long-term prediction;
- ❑ *CELP (Codebook-Excited Linear Prediction)* – A função de excitação é obtida por consulta a uma tabela pré-definida de vetores de excitação, denominada *codebook*. Utiliza short-term prediction e long-term prediction;

As principais aplicações desta família são:

- ❑ VOCODERS *full-rate* e *half-rate* para telefonia celular GSM (RPE-LPC);
- ❑ VOCODER EFR (*enhanced full-rate*) para telefonia celular GSM e VOCODER para telefonia celular TDMA IS-136 (ACELP – *Algebraic Codebook-Excited Linear Prediction* – variante do CELP);
- ❑ Recomendação ITU-T G.728 (CELP);
- ❑ VOCODER para telefonia celular CDMA IS-95 (CELP);
- ❑ Recomendações ITU-T G.729 e G.729A (CS-ACELP – *Conjugate Structure Algebraic Codebook-Excited Linear Prediction* – outra variante do CELP);
- ❑ Recomendação ITU-T G.723 para aplicações de videoconferência (incluída nas recomendações ITU-T H.323 e H.324) e VoIP (MP-MLQ/ACELP – *Multipulse-Maximum Likelihood Quantization/Algebraic Codebook-Excited Linear Prediction* – variações, respectivamente, do MPE-LPC e CELP).
- ❑ EVRC (*Enhanced Variable Rate CODEC*) para telefonia celular CDMA2000 IS-127 (Q-CELP – Variante proprietária da Qualcomm para o CELP)
- ❑ VOCODERS AMR (Adaptive Multi-Rate) para telefonia celular GSM (MR-ACELP – *Multi-Rate Algebraic Codebook-Excited Linear Prediction* – variante do CELP)
- ❑ SMV (*Selective Mode VOCODER*) para telefonia celular CDMA2000 (eX-CELP – *eXtended Codebook-Excited Linear Prediction* – mais uma variante do CELP)

Referências

1. <http://mathworld.wolfram.com/FourierSeries.html>
2. <http://en.wikipedia.org/wiki/G.711>
3. <http://en.wikipedia.org/wiki/G.726>
4. <http://www.palowireless.com/bluetooth/docs/BDouglas.pdf>
5. http://engr.smu.Edu/~ebird/Handouts/EETS8306_Lecture4_DigitalCommunicationBasics_2004_RevA.pdf
6. <http://mia.ece.uic.edu/~papers/WWW/MultimediaStandards/chapter3.pdf>
7. <http://www.mat.ucsb.edu/~gggroup/casmagarticlefinal.pdf>
8. <http://cs.haifa.ac.il/~nimrod/Compression/Speech/S4ABYS2004.pdf>
9. <http://61.153.34.35:8002/~kjqk/txxb/980508.htm>
10. <http://dcmc.ee.ncku.edu.tw/pdf/course/MC/MC05.pdf>
11. http://akhisar.sdsu.edu/abut/EE658/CHAP10_2004.pdf
12. http://en.wikipedia.org/wiki/Adaptive_Multi-Rate
13. <http://en.wikipedia.org/wiki/SMV>