Signal Coding

Pulse Modulation:

\[
SNR = \frac{\sigma_x^2}{\sigma_e^2} = \frac{\sum x^2(k)}{\sum e^2(k)} = \frac{3.2B}{x_{max}^2/\sigma_x^2}
\]

\[
SNR(dB) = 10\log(\frac{\sigma_x^2}{\sigma_e^2}) = 6B + 4.77 - 20\log(x_{max}/\sigma_x)
\]

\[
SNR(dB) = 6B - 7.2, \text{ if } x_{max} = 4\sigma_x
\]
Companded PCM

\[
\hat{y}(k) = Q[\log|x(k)|] = \log|x(k)| + \epsilon(k)
\]

\[
\hat{x}(k) = \exp[\hat{y}(k)] \text{sign}[x(k)] = x(k) \exp[\epsilon(k)]
\]

If \( \epsilon(k) \) is small:

\[
\hat{x}(k) = x(k)[1 + \epsilon(k)] = x(k) + \epsilon(k)x(k) = x(k) + n(k)
\]

\[
\text{SNR} = \frac{\sigma_x^2}{\sigma_n^2} = \frac{\sigma_x^2}{\sigma_x^2 + \sigma^2} = \frac{1}{\sigma^2}
\]

\[\mu - \text{law:}\]

\[
y(k) = x_{\max} \frac{\ln(1 + \mu \frac{|x(k)|}{x_{\max}})}{\ln(1 + \mu)} \text{sign}[x(k)]
\]

\[A = 87.6\]

\[\mu = 255\]
DPCM

\[ d(k) = x(k) - \hat{x}(k) \]

\[ \hat{d}(k) = d(k) + e(k) \]

\[ \hat{x}(k) = \hat{x}(k) + \hat{d}(k) = x(k) + e(k) \]

\[ SNR = \frac{\sigma_x^2}{\sigma_e^2} = \frac{\sigma_x^2 \cdot \sigma_d^2}{\sigma_d \cdot \sigma_e^2} = G_d \cdot SNR_Q \]

For linear predictor, P:

\[ \hat{x}(k) = \sum_{n=1}^{P} \alpha_n \hat{x}(k-n) \]

\[ \Rightarrow H(z) = \frac{1}{1 - \sum_{n=1}^{P} \alpha_n z^{-n}} \]

\[ \alpha_n \] can be fixed or adaptive
Finding Predictor Coefficients

\[ \sigma_d^2 = E[d^2(k)] = E[(x(k) - \hat{x}(k))^2] \]

\[ = E[(x(k) - \sum_{n=1}^{P} \alpha_n \hat{x}(k-n))^2] \]

\[ = E[(x(k) - \sum_{n=1}^{P} \alpha_n x(k-n) - \sum_{n=1}^{P} \alpha_n e(k-n))^2] \]

To find the \( \alpha_j, 1 \leq j \leq P \), that minimise \( \sigma_d^2 \):

\[ \frac{\partial \sigma_d^2}{\partial \alpha_j} = -2E[(x(k) - \sum_{n=1}^{P} \alpha_n (x(k-n)e(k-n))) (x(k-j)e(k-j))] = 0 \quad (5.3) \]

\[ \Rightarrow E[x(k-j)x(k)] + E[e(k-j)x(k)] = \sum_{n=1}^{P} \alpha_n E[x(k-j)x(k-n)] + \sum_{n=1}^{P} \alpha_n E[e(k-j)e(k-n)] + \sum_{n=1}^{P} \alpha_n E[e(k-j)e(k-n-n)] \quad (5.4) \]

If quantization is fine, e(k) is uncorrelated with x(k) and e(k) is white noise.

\[ \Rightarrow E[x(k-j)e(k-n)] = 0 \text{ and } E[e(k-j)e(k-n)] = \sigma_e^2 \delta(j-n) \]

So Eq (5.4) becomes:

\[ R_{xx}(j) = \sum_{n=1}^{P} \alpha_n (R_{xx}(j-n) + \sigma_e^2 \delta(j-n)), 1 \leq j \leq P \quad (5.5) \]
Delta Modulation

a) linear DM: \( \hat{x}(k) = x(k) - 1(k) \)
If \( \alpha = 1 \), \( d(k) = x(k) - \hat{x}(k - 1) = x(k) - x(k - 1) - e(k - 1) \)
For \( \hat{x}(k) \) to increase as fast as \( x(t) \) at max slope:
\[
\frac{\Delta}{T} \geq \max \left| \frac{dx(t)}{dt} \right|
\]
If \( \Delta \) is too small - slope overload
If \( \Delta \) is too big - large granular noise

b) adaptive DM: step size is varied:
i) Song algorithm: min stepsize is \( \delta \);
\[
\Delta(k) = \Delta(k - 1) \pm \delta
\]
ii) CVSD:
\( \Delta(k) = \beta \Delta(k - 1) + D_{\text{max}} \) if \( c(k) = c(k-1) = c(k-2) \)
\( \Delta(k) = \beta \Delta(k - 1) + D_{\text{min}} \) otherwise
\( 0 < \beta < 1 \) and \( D_{\text{max}} \gg D_{\text{min}} > 0 \)

12 kbit/s CVSD is used by Motorola’s SECURENET line of digitally encrypted two-way radio products.
64 kbit/s CVSD is one of the options to encode voice signals in telephony-related Bluetooth service profiles, e.g. between mobile phones and wireless headsets. The other options are PCM with logarithmic a-law or \( \mu \)-law quantization.
One of the most popular parametric representations of the LPC filter uses the line spectral frequencies (LSFs), also known as line spectrum pairs (LSPs). Consider the polynomials $P(z)$ and $Q(z)$ given by:

\[
P(z) = A(z) + z^{-(p+1)}A(z^{-1})
\]

\[
Q(z) = A(z) - z^{-(p+1)}A(z^{-1})
\]

It follows that: $A(z) = 0.5[P(z) + Q(z)]$.

The zeroes of $P(z)$ and $Q(z)$ can be expressed as $e^{j\omega i}$. These frequencies are called the line spectral frequencies and they uniquely characterise the LPC inverse filter $A(z)$.

$A(z)$ is minimum phase if and only if all the zeros of the LSF polynomials $P(z)$ and $Q(z)$ are interlaced on the unit circle. The LSFs consist of the angular positions of these zeros. Only $p/2$ zeros are needed to specify each LSF polynomial since the zeros come in complex conjugate pairs.

The LSFs have a number of interesting properties that have made them common spectral parameters:

- A stable synthesis filter is guaranteed when the zeros are interlaced on the unit circle. This is simple to verify when the LSFs are quantized. The LSF coefficients allow interpretation in terms of formant frequencies. If two neighbouring LSFs are close in frequency, it is likely that they correspond to a narrow bandwidth spectral resonance in that frequency region; otherwise, they usually contribute to the overall tilt of the spectrum. Shifting the LSF frequencies has a localized spectral effect; quantization errors in an LSF will primarily affect the region of the spectrum around that frequency.
**LPC drawbacks**

Unfortunately, things are not so simple. One reason is that there are speech sounds which are made with a combination of buzz and hiss sources (for example, the initial consonants in "this zoo" and the middle consonant in "azure"). Speech sounds like this will not be reproduced accurately by a simple LPC encoder.

Another problem is that any inaccuracy in the estimation of the formants means that more speech information gets left in the residue. The aspects of nasal sounds that don’t match the LPC model, for example, will end up in the residue. There are other aspects of the speech sound that don’t match the LPC model; side branches introduced by the tongue positions of some consonants, and tracheal (lung) resonances are some examples.

Therefore, the residue contains important information about how the speech should sound, and LPC synthesis without this information will result in poor quality speech. For the best quality results, we could just send the residue signal, and the LPC synthesis would sound great. Unfortunately, the whole idea of this technique is to compress the speech signal, and the residue signal takes just as many bits as the original speech signal, so this would not provide any compression.

**Hybrid Coding:** Inverse LPC filtering \( A(z) = 1/H(z) \) removes short term correlations from \( s(k) \) to leave residual \( e(k) \):

```
Input Signal
```

---

95a
1. **APC**: residual is quantized directly with 2-4 bits/sample. (DPCM no use - why?). Good speech quality when bit rate > 16k bits/sec.

![APC encoder and decoder diagram]

2. **RELP**: residual, e(k) is low-pass filtered to 1kH(z) and down sampled so fewer samples need be sent. Fairly good quality when bit rate > 8kb/s.

![RELP encoder and decoder diagram]
**Pitch Prediction**

Glottal period correlation may be exploited using long-term predictor:

\[
d(k) = d_1(k) - \sum_{n=1}^{P} \alpha_n d_1(k - n)
\]

\[
d_1(k) = x(k) - \beta x(k - M)
\]

Difficult to optimize \( \beta \), \( M \) and \( \alpha_n \) to minimise \( \sigma_d^2 \) so optimize \( M \) then \( \beta \) then \( \alpha_n \)

Pitch prediction can also be 3rd order:

\[
H_1(z) = \beta_1 z^{-M-1} + \beta_2 z^{-M} + \beta_3 z^{-M+1}
\]
Analysis by Synthesis Coding

1. **Multipulse Excited LPC**: excitation is small no of non-uniformly spaced pulses of varying amplitudes
   No V/U or pitch detection needed.

---

97
**Perceptual Weighting:** perceived distortion rather than M.S.E is minimized. Trade off increases noise power at formants with decreases in valleys.
2. **Regular Pulse Excited LPC**: pulses are spaced uniformly. Eg. every 4 samples.

```
|   •   |   •   |   •   |
|   •   |   •   |   •   |
|   •   |   •   |   •   |
|   •   |   •   |
```

98a
Code Excited Linear Prediction

Excitation is "innovation sequence", which may be white noise or residuals generated by training, stored in code-book. Optimum sequence is selected by filtering each sequence in code-book in turn; the one which gives min. weighted M.S.E is chosen and its address is sent to decoder.

Code Book size typically 1024
Innovation length typically 32 samples
LPC block length typically 128 ”
CELP bit rate typically 4-8k b/s
Main problem is large amount of computation.

Variants include:
LD-CELP
ACELP
VSELP . . .
Sub-band Coding

Signal is split into frequency bands. Each band is downsampled and quantised separately.

Total tx rate, \( I = \sum_{k=1}^{M} f_{sk} R_k = \frac{2W}{M} \sum_{k=1}^{M} R_k \) for eq width bands.

\[
I = 2WR, \sum_{k=1}^{M} R_k = MR 
\]

For non-overlapping bands, \( \sum_{k=1}^{M} \sigma_{xk}^2 = \sigma_x^2 \) and \( \sum_{k=1}^{M} \sigma_{rk}^2 = \sigma_{rSBC}^2 \)

Assuming error-free tx and PCM for each band (\( \sigma_{r_k}^2 = \sigma_{rSBC}^2 \)):

\[
\sigma_{rSBC}^2 = \sum_{k=1}^{M} \frac{2^{-2Rk} \sigma_{rk}^2}{SNR_k \sigma_{xk}^2} 
\]

For full-band PCM coder with \( R \) bits/sample:

\[
\sigma_{rPCM}^2 = \frac{2^{-2R}}{SNR} \sigma_x^2 
\]

\[\Rightarrow G_{SBC} = \frac{\sigma_{rPCM}^2}{\sigma_{rSBC}^2} = \frac{2^{-2R} \sigma_x^2}{\sum_{k=1}^{M} 2^{-2Rk} \sigma_{xk}^2} = \frac{2^{-2} \sum_{k=1}^{M} R_k/M \sum_{k=1}^{M} \sigma_{xk}^2}{\sum_{k=1}^{M} 2^{-2Rk} \sigma_{xk}^2} \]

\[
SNR_{SBC}(dB) = SNR_{PCM}(dB) + 10\log_{10} G_{SBC} 
\]
$G_{SBC}$ can’t exceed 1 for flat spectra but, for non-flat spectra, $G_{SBC} > 1$ if $R_k$ is matched to $\sigma^2_x$ by Bit Allocation.

Eg1 2-band coding of flat spectrum signal:
$\sigma^2_{x1} = \sigma^2_{x2} = \sigma^2_x/2$, $\Rightarrow G_{SBC} = 2 \frac{2^{-(R_1+R_2)}}{2^{R_1+2^{R_2}}}$

Eg2 2-band coding of signal with non-flat spectrum:
Let $\alpha = 2/17$
$\sigma^2_{x1} = \frac{16}{17} \sigma^2_x$, $\sigma^2_{x2} = \frac{1}{17} \sigma^2_x$, $\Rightarrow G_{SBC} = 17 \frac{2^{-(R_1+R_2)}}{16.2^{R_1+2^{R_2}}}$
For $R_1 = R_2 = 3$, $G_{SBC} = 1$
For $R_1 = 4$, $R_2 = 2$, $G_{SBC} = 17/8$
<table>
<thead>
<tr>
<th>Type of Coder</th>
<th>ITU-T Standard</th>
<th>Bit Rates in kb/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCM</td>
<td>G.711</td>
<td>64</td>
</tr>
<tr>
<td>ADPCM</td>
<td>G.726</td>
<td>16, 24, 32, 40</td>
</tr>
<tr>
<td>Embedded ADPCM</td>
<td>G.727</td>
<td>16, 24, 32, 40</td>
</tr>
<tr>
<td>Low-Delay CELP</td>
<td>G.728</td>
<td>16</td>
</tr>
<tr>
<td>ACELP</td>
<td>G.729</td>
<td>8</td>
</tr>
<tr>
<td>ACELP/MP-LPC</td>
<td>G.723.1</td>
<td>5.3 / 6.3</td>
</tr>
<tr>
<td>Not defined yet</td>
<td>ITU-T G.xxx</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 4: Speech coding for telecommunications

<table>
<thead>
<tr>
<th>Type of Coder</th>
<th>Standard/Product</th>
<th>Bit Rates in kb/s</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPE-LTP</td>
<td>GSM 06.10</td>
<td>13</td>
<td>Europe</td>
</tr>
<tr>
<td>ACELP</td>
<td>GSM EFR, PCS 1900 EFR</td>
<td>12.2</td>
<td>Europe, USA</td>
</tr>
<tr>
<td>ACELP</td>
<td>EFR IS-641</td>
<td>7.4</td>
<td>USA</td>
</tr>
<tr>
<td>QCRLP</td>
<td>IS-96*</td>
<td>≤8.5</td>
<td>USA</td>
</tr>
<tr>
<td>VSELP</td>
<td>IS-54*</td>
<td>7.95</td>
<td>USA</td>
</tr>
<tr>
<td>VSELP</td>
<td>GSM Half Rate 06.20</td>
<td>5.6</td>
<td>Europe</td>
</tr>
<tr>
<td>VSELP</td>
<td>PDC</td>
<td>6.7</td>
<td>Japan</td>
</tr>
<tr>
<td>PSI-CELP</td>
<td>PDC Half Rate</td>
<td>3.45</td>
<td>Japan</td>
</tr>
</tbody>
</table>

RPE = regular pulse excitation; LTP = long-term prediction; ACELP = algebraic CELP; EFR = enhanced full rate; SELP = vector sum excited LP coding; PSI = pitch-synchronous innovation CELP
*These coders will be replaced by the EFR coder IS-641

Table 5: Speech coding for mobile radio
<table>
<thead>
<tr>
<th>Mean opinion score</th>
<th>Impairment scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Imperceptible</td>
</tr>
<tr>
<td>4</td>
<td>Perceptible, but not annoying</td>
</tr>
<tr>
<td>3</td>
<td>Slightly annoying</td>
</tr>
<tr>
<td>2</td>
<td>Annoying</td>
</tr>
<tr>
<td>1</td>
<td>Very annoying</td>
</tr>
</tbody>
</table>

Table 2: 5-point MOS impairment scale

<table>
<thead>
<tr>
<th>Coding scheme</th>
<th>Bit Rate in kb/s</th>
<th>MOS value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FS 1015 LPC-10E</td>
<td>2.4</td>
<td>2.6</td>
</tr>
<tr>
<td>FS 1017 MELP</td>
<td>2.4</td>
<td>3.3</td>
</tr>
<tr>
<td>GSM Enhanced Half Rate</td>
<td>5.6</td>
<td>3.3</td>
</tr>
<tr>
<td>ACELP</td>
<td>8.0</td>
<td>4.0</td>
</tr>
<tr>
<td>GSM Enhanced Full Rate</td>
<td>12.2</td>
<td>3.8</td>
</tr>
<tr>
<td>GSM (RPE-LTP)</td>
<td>13.0</td>
<td>3.5</td>
</tr>
<tr>
<td>Low Delay CELP</td>
<td>16.0</td>
<td>4.1</td>
</tr>
<tr>
<td>ADPCM (DECT)</td>
<td>32.0</td>
<td>4.1</td>
</tr>
<tr>
<td>PCM (ISDN)</td>
<td>64.0</td>
<td>4.3</td>
</tr>
</tbody>
</table>

Table 6: MOS values of various speech coding schemes
MPEG Audio Coding
- based on Psycho-acoustic Model

Ear:

Cochlea:

Basilar membrane:
Equivalent Rectangular Bandwidth (ERB) - Critical Bands.
The auditory system is thought to contain an array of overlapping band-pass filters known as auditory filters. They occur along the basilar membrane (BM) and increase the frequency selectivity of the cochlea. The bandwidth of the auditory filter is called the critical bandwidth, and is the band of frequencies which are passed by the filter. If a signal and masker are presented simultaneously then only the masker frequencies falling within the critical bandwidth contribute to masking of the signal.

\[ \text{ERB} = 24.7 \log(4.37F + 1) \]
where the ERB is in Hz and F is the centre frequency in kHz.

Auditory Masking (Frequency):

MPEG Audio Coder divides 20-20kHz audio spectrum into 32 bands. Say Sub-Band 8 has 1kHz tone at 60dB. Say PsychoAcoustic Model finds masking threshold for SB8 is 36dB below this tone. So acceptable SNR is 60-36 = 24dB (≡ 4 bits instead of 10).
MPEG1 Audio

CD Audio: 44,100kHz x 16bits x 2channels > 1.4Mb/s
MPEG1 Std: 1.5Mb/s (1.2Mb/s for Video, .3Mb/s for Audio)
MPEG1 Audio Compression Ratios: 2.7 to 24
MPEG1 Audio Fs: 32, 44.1 or 48kHz
MPEG1 Audio supports 1 or 2 channels in 1 of 4 modes:
1. Mono
2. Dual Mono
3. Stereo
4. Joint Stereo
Typical Bit Rates:

<table>
<thead>
<tr>
<th>Layer</th>
<th>Ratio</th>
<th>kb/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>3:1</td>
<td>384</td>
</tr>
<tr>
<td>II</td>
<td>6:1</td>
<td>192</td>
</tr>
<tr>
<td>III</td>
<td>10:1</td>
<td>128</td>
</tr>
</tbody>
</table>

![Diagram of MPEG1 Audio encoding process]
PsychoAcoustic Model 1 (used in MP1, MP2):

- FFT, N=512 for L I and 1024 for L II
- SPL is each SB computed
- Quiet Threshold provided
- Discriminate between tonal and non-tonal

\[ SMR_n = \text{Signal in } SB_n - \text{Masking Threshold for } SB_n \]

PsychoAcoustic Model 2 (used in MP3):

- FFT size can be varied, eg 192 for short block, 576 for long block
- Includes Temporal Masking

After analysis, the first levels of 16 of the 32 bands are:

<table>
<thead>
<tr>
<th>Band</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level (db)</td>
<td>0</td>
<td>8</td>
<td>12</td>
<td>10</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>60</td>
<td>35</td>
<td>20</td>
<td>15</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

If the level of the 8th band is 60dB, it gives a masking of 12 dB in the 7th band, 15dB in the 9th.

Level in 7th band is 10 dB ( < 12 dB ), so ignore it.

Level in 9th band is 35 dB ( > 15 dB ), so send it.

Only the amount above the masking level needs to be sent, so instead of using 6 bits to encode it, we can use 4 bits saving 2 bits (= 12 dB).
Temporal Masking

In addition to simultaneous masking two time domain phenomena also play an important role in human auditory perception, pre-masking and post-masking. The temporal masking effects occur before and after a masking signal has been switched on and off, respectively. The duration when pre-masking applies is less than -or as newer results indicate, significantly less than-one tenth that of the post-masking, which is in the order of 50 to 200 msec.

Total effect of both frequency and temporal maskings:

\[
\text{level (dB)}
\]

\[
\text{Masking tone}
\]

\[
\text{time}
\]

\[
\text{freq}
\]

\[
\text{Inaudible tones (under curve)}
\]
MPEG1 Audio Coder:
Layer I Encoder

![Diagram of Layer I Encoder](image-url)
**MP1 Frame Structure**

(valid for 384 [32x12] PCM samples ≡ 8ms at $F_s = 48$kHz)

<table>
<thead>
<tr>
<th>Header</th>
<th>CRC</th>
<th>Bit Alloc</th>
<th>Scale Factor</th>
<th>SB samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>12bit sync</td>
<td>16bit</td>
<td>4bit</td>
<td>6bit</td>
<td>0-15bits</td>
</tr>
<tr>
<td>20bit sys</td>
<td>x32</td>
<td>x32</td>
<td>x32</td>
<td></td>
</tr>
</tbody>
</table>

Layer II similar to L I but uses 3x12x32 = 1152 samples
Layer II bitstream similar to L I but uses Scale Factor Select Info (2 bits) as follows:
To reduce SF bits, SFs are calculated for 3 successive blocks of 12 SB samples and classified into SF patterns. Depending on pattern, 1, 2, or 3 SFs are sent with a 2-bit SFSI

![Block diagram of MPEG1 Layer-II](image-url)
Layer III

MP3 specifies long (18 samples) or short (6) blocks.

Mixed has long blocks for 2 lower SBs, short blocks for 30 hi-freq SBs

MP3 also uses noiseless Huffman VLC to further reduce bit rate
MP3 Decoder

1. BEGIN
2. GET BIT STREAM, FIND HEADER
3. DECODE SIDE INFORMATION
4. DECODE SCALE FACTORS
5. DECODE HUFFMAN DATA
6. REQUANTIZE SPECTRUM
7. REORDER SPECTRUM
   IF (window_switching_flag) AND (block_type==2)
8. JOINT STEREO PROCESSING
    (if applicable)
9. ALIAS REDUCTION
10. SYNTHESIZE VIA IMDCT & OVERLAP-ADD
    (IMDCT either 18 or 6,6,6 depending on
    window_switching_flag and block_type)
11. SYNTHESIZE VIA POLYPHASE FILTERBANK
12. OUTPUT PCM SAMPLES

END
Stereo Coding

All MPEG1 layers support Intensity Stereo: some hi-freq SBs are coded with a single summed signal instead of independent L and R codes for each of the 32 SBs. Decoder reconstructs L and R channels based only on single summed signal and separate L and R SFs, so spectral shape of L and R is same within each intensity coded SB, but magnitude differs.

MP3 also supports Mid/Side joint stereo coding: forming the sum of the left and right channels (L+R) and the difference between the left and right channels (L-R) and encoding these sum and differences separately:
M = L + R [normally large]
S = L - R [normally small, so needs fewer bits]
The decoder then carries out the following equations to return the L and R signals:
Left = M + S = 0.5 x ((L+R) + (L-R)) = 0.5 x 2L = L
Right = M - S = 0.5 x ((L+R) - (L-R)) = 0.5 x 2R = R
Unlike intensity stereo coding, which is a lossy form of joint stereo coding, mid/side joint stereo coding is lossless, which means that the original information is returned without losing any of the original information. Therefore, mid/side joint stereo coding does not have the collapsing and non-existent stereo image problems that MP2 has at low bit rates.
**MPEG2 Backwards Compatible** Provides 5-ch audio (L, R, C, Ls, Rs)

The MPEG-2 BC format defines support for lower sampling rates. In addition the MPEG-2 standard defines an extension to multi-channel that is backwards compatible to MPEG-1, and inclusion of up to seven multi-lingual tracks for mixing with the multi channel audio content. MPEG-2 adds support for the lower sampling frequencies 16, 22.05 and 24 kHz and bit rates down to 8 kb/s.

MPEG-2 audio also adds a compatible extension to MPEG-1 audio encoding, which enables the transmission of stereo and multichannel audio in a single bitstream. It can operate at a wide range of bitrates up to more than 1 Mbit/s. A 5.1 multichannel (five full bandwidth channels plus a Low Frequency Enhancement channel) movie soundtrack requires an average bit rate of 384-640 kbit/s.

The MPEG-2 standard was designed with compatibility as a major consideration. The core of the MPEG-2 bitstream is an MPEG1 bitstream, which enables fully compatible decoding of a multi-channel bitstream by a low complexity MPEG1 audio decoder. An MPEG-1 decoder will decode the stereo part of the MPEG-2 frame, representing an appropriate (stereo) ‘downmix’ of all channels in the multichannel frame, and ignore the multichannel extension.

But need for compatibility with MPEG1 imposes a noticeable loss of quality.
MPEG2 NBC (aka AAC)

Can handle up to 48 audio channels, 15 Lo Freq Enhancement channels, 15 embedded data streams
Allows lower sampling rates: 16, 22.05, and 24 kHz
"Better" audio quality @ 96kb/s than MP3 @ 128kb/s
Main Profile uses: Huffman, Quantisation and Scaling, M/S stereo, Backward Adaptive Prediction, TNS, MDCT (1024 or 128), Gain Control
Low Complexity Profile: no prediction, TNS limited to 12 coeffs
Scalable Sampling Rate Profile: no prediction, no coupling channel, TNS limited to 12 coeffs
AAC uses the coding tools already present in MP3, but uses them in a better way:
* The filter bank is a pure MDCT and not an hybrid filter bank like in MP3
* Long windows are nearly twice as long as MP3 ones, providing better frequency resolution
* Short windows are smaller than MP3 ones, providing better transients handling and less pre-echo
* Ability to toggle middle/side stereo on a subband basis instead of entire frame basis
* Ability to toggle intensity stereo on a subband basis instead of using it only for a contiguous group of subbands.
This first part is mainly a removal of MP3 limitations, as AAC standard doesn’t have to preserve compatibility. AAC also introduces some new tools over previous coding schemes:
* Temporal Noise Shaping (TNS) is a tool designed to control the location, in time, of the quantization noise by transmission of filtering coefficients
* Prediction [of freq coeffs] is a tool designed to enhance compressibility of stationary signals
Temporal Noise Shaping

Figure 3: Transient signal (castanets, uncoded).

Figure 4: Coding noise in decoded castanets signal with (above) and without (below) TNS.
MPEG2 AAC Encoder
MPEG4 Audio Coding  Coding of Audio Objects

MPEG-4 coding of audio objects provides tools for both representing natural sounds (such as speech and music) and for synthesizing sounds based on structured descriptions. The representation for synthesized sound can be derived from text data or so-called instrument descriptions and by coding parameters to provide effects, such as reverberation and spatialization. The representations provide compression and other functionalities, such as scalability or play-back at different speeds.

![Diagram of MPEG-4 audio system](image)

**Fig. 3.** The MPEG-4 audio system, showing the interaction between decoding, scene description, and audiovisual synthesis. The conceptual flow is from the bottom of the figure to the top. At the bottom, two multiplexed MPEG-4 bitstreams, each from a different server, convey several elementary streams containing compressed data. Each bitstream is demultiplexed, a total of four elementary streams are produced. The elementary streams are decoded using various MPEG-4 decoders into four primitive audio objects containing uncompressed PCM audio data. The audio data is manipulated by the AudioBIFS scene graph and presented to the listener as though it emanates from the sound nodes. © 1999 Marcel Dekker [7], used with permission.
1. Synthesized Sound (input = Text, Music Score, . . .)

a) Text to Speech functionalities:
- speech synthesis using prosody
- facial animation control with phonemes
- modes: pause, resume, fwd/bwd
- international language support
- specify age, gender, dialect

The basic TTSI format is extremely low bitrate. In the most compact method, one can send a bitstream that contains only the text to be spoken and its length. In this case, the bitrate is 200 bits per second. The synthesizer will add pre-defined or rule-generated prosody to the synthesized speech (in a nonnormative fashion). The synthesized speech with predefined prosody will deliver emotional content to the listener.

On the other hand, one can send a bitstream that contains text as well as the detailed prosody of the original speech, that is, phoneme sequence, duration of each phoneme, base frequency (pitch) of each phoneme, and energy of each phoneme. The synthesized speech in this case will be very similar to the original speech since it employs the original prosody. Thus, one can send speech with subtle nuances without any loss of intonation using MPEG-4 TTSI.

One of the important features of the MPEG-4 TTSI is the ability to synchronize synthetic speech with the lip movements of a computer-generated avatar or "talking head". In this technique, the TTS synthesizer generates phoneme sequences and their durations, and communicates them to the facial animation visual object decoder so that it can control the lip movement. With this feature, one can not only hear the synthetic speech but also see the synchronized lip movement of the avatar.
b) Score Driven Synthesis:

- decoder i/p is Structured Audio Orchestra Language (SAOL), which defines an "orchestra" made up of "instruments".
SAOL handles wavetable, FM, physical modelling, etc
SAOL can generate audio effects (footsteps, door closures, etc), simulations of natural sounds (rain, wind, etc) and fully synthetic sounds.
SAOL can describe Special Effects like reverb, spatializers, mixers, limiters, filters, chorus, etc

\[Structured\ \text{Audio}\ \text{Decoder}\]
A SAOL orchestra, containing one instrument that makes a ramped complex tone.

```plaintext
global {
  srate 32000;
  krate 1000;
}
instr beep(pitch, amp) {
  asig out;
  ksig env;
  table sound(harm,2048,1,0.5,0,0.2);
  env = kline(0,0.1,amp,dur-0.1,0);
  out = oscil(sound,pitch) * amp;
  output(out);
}
```

A SASL score, which uses the orchestra above to play four notes.

```plaintext
0.0 beep 1.0 440 0.5
1.0 beep 2.0 220 0.2
2.0 beep 1.0 264 0.5
3.0 beep 1.0 440 0.5
4.0 end
```

A musical notation diagram is shown below the score.
2. Natural Sound

The AAC standard (part 7 of MPEG-2) has brought down to 64 kbit/s virtual transparency of single channel music which MPEG-1 Audio had set at 128 kbit/s and MPEG-4 Audio will bring interesting performance even at lower bitrates than 64 kbit/s. AAC is therefore already providing part of the MPEG-4 Audio standard.

MPEG-4 standardises natural audio coding at bitrates ranging from 2 kbit/s up to 64 kbit/s. For the bitrates from 2 kbit/s up to 64 kbit/s, the MPEG-4 standard normalises the bitstream syntax and decoding processes in terms of a set of tools. In order to achieve the highest audio quality within the full range of bitrates and at the same time provide the extra functionalities, three types of coder have been defined. The lowest bitrate range between about 2 and 6 kbit/s, mostly used for speech coding at 8 kHz sampling frequency, is covered by parametric coding techniques. Coding at the medium bitrates between about 6 and 24 kbit/s uses Code Excited Linear Predictive (CELP) coding techniques. In this region, two sampling rates, 8 and 16 kHz, are used to support a broader range of audio signals (other than speech). For the higher bitrates typically starting at about 16 kbit/s, time to frequency (T/F) coding techniques, namely VQ and AAC codecs, are applied. The audio signals in this region typically have bandwidths starting at 8 kHz.

To allow for smooth transitions between the bitrates and to allow for bitrate and bandwidth scalability, a general framework has been defined. This is illustrated in the figure below.
TwinVQ (Transform-domain weighted interleaved Vector Quantisation)

In the context of the MPEG-4 audio (MPEG-4 Part 3), TwinVQ is an audio codec optimized for audio coding at ultra low bitrates around 8 kbit/s. TwinVQ is one of the object types defined in MPEG-4 Audio version 1. This object type is based on a general audio transform coding scheme which is integrated with the AAC coding framework, a spectral flattening module, and a weighted interleave vector quantization module. This scheme reportedly has high coding gain for low bit rate and potential robustness against channel errors and packet loss, since it does not use any variable length coding and adaptive bit allocation. It supports bitrate scalability, both by means of layered TwinVQ coding and in combination with the scalable AAC.

Note that some commercialized products such as Metasound (Voxware), SoundVQ (Yamaha), and SolidAudio (Hagiwara) are also based on the TwinVQ technology, but the configurations are different from the MPEG-4 TwinVQ.

TwinVQ uses Twin vector quantization. The proprietary TwinVQ codec supports constant bit rate encoding at 80, 96, 112, 128, 160 and 192 kbit/s. It was claimed that TwinVQ files are about 30 to 35% smaller than MP3 files of adequate quality. For example, a 96 kbit/s TwinVQ file allegedly has roughly the same quality as a 128 kbit/s MP3 file. The higher quality is achieved at the cost of higher processor usage.
HILN or Harmonic and Individual Lines and Noise is a parametric audio codec for audio. The basic premise of the encoder is that most audio, and particularly speech, can be synthesized from only sinusoids and noise. The encoder describes individual sines with amplitude and frequency, harmonic tones by fundamental frequency, amplitude and the spectral envelope of the partials, and the noise by amplitude and spectral envelope. This type of encoder is capable of encoding audio to between 6 and 16 kilobits per second for a typical audio bandwidth of 8 kHz. The framelength of this encoder is 32 msec.

A typical codec extracts sinusoid information from the samples by applying a Short-time Fourier transform to the samples and using that to find the important harmonic content of a single frame. By matching sines across frames, the encoder is capable of grouping them into harmonic lines and individual sines. The matching can take amplitude, frequency and phase into account when trying to match sinusoids across frames. Differences between amplitude and frequency within a track can be coded with less bits than each individual single sinus would require, thus the longer a track the encoder can find, the better it will be able to reduce the final bitrate.

Synthesizing only the sinusoids sounds artificial and metallic. To mask this, the encoder subtracts the synthesized sinusses from the original audio signal. The residual is then matched to a linear filter that is excited with white noise. The extracted parameters can than be quantized, coded and multiplexed into a bitstream.
The Advanced Audio Coding in MPEG-4 Part 3 was enhanced relative to the previous standard MPEG-2 Part 7, in order to provide better sound quality for a given encoding bitrate.

AAC’s multiple codecs:
* Low Complexity Advanced Audio Coding (LC-AAC)
* High-Efficiency Advanced Audio Coding (HE-AAC)
* Scalable Sample Rate Advanced Audio Coding (AAC-SSR)
* Bit Sliced Arithmetic Coding (BSAC)
* Long Term Predictor (LTP)
AAC-SSR

AAC Scalable Sample Rate was introduced by Sony to the MPEG-4 standard. The audio signal is first split into 4 bands using a 4 band polyphase quadrature filter bank. Then these 4 bands are further split using MDCTs with a size k of 32 or 256 samples. This is similar to normal MPEG-4 AAC which uses MDCTs with a size k of 128 or 1024 directly on the audio signal.

The advantage of this technique is that short block switching can be done separately for every PQF band. So high frequencies can be encoded using a short block to enhance temporal resolution, low frequencies can be still encoded with high spectral resolution.

Why AAC-SSR was introduced

The idea behind AAC-SSR was not only the advantage listed above, but also the possibility of reducing the data rate by removing 1, 2 or 3 of the upper PQF bands. A very simple bitstream splitter can remove these bands and thus reduce the bitrate and sample rate.

Example:

* 4 subbands: bitrate = 128 kbit/s, sample rate = 48 kHz, f-lowpass = 20 kHz
* 3 subbands: bitrate = 120 kbit/s, sample rate = 48 kHz, f-lowpass = 18 kHz
* 2 subbands: bitrate = 100 kbit/s, sample rate = 24 kHz, f-lowpass = 12 kHz
* 1 subband: bitrate = 65 kbit/s, sample rate = 12 kHz, f-lowpass = 6 kHz
BSAC

Bit Sliced Arithmetic Coding is an MPEG-4 standard (ISO/IEC 14496-3 sub-part 4) for scalable audio coding. BSAC uses an alternative noiseless coding to AAC, with the rest of the processing being identical to AAC. This support for scalability allows for nearly transparent sound quality at 64 kbit/s and graceful degradation at lower bit rates. BSAC coding is best performed in the range of 40 kbit/s to 64 kbit/s, though it operates in the range of 16 kbit/s to 64 kbit/s. The AAC-BSAC codec is used in Digital Multimedia Broadcasting (DMB) applications.

- Third alternative quantization and coding scheme defined in Version 2:

BSAC (Bit Sliced Arithmetic Coding)

- Arithmetic coding of vectors (slices) constructed from the LS8, LS8+1, LS8+2 ... of all frequency domain samples
- Small scalability steps possible (~ 1kbit/s)
- Good performance at higher bitrates
- At lower bit rates the other schemes perform better
HE-AAC

HE-AAC is an extension of AAC using spectral band replication (SBR), and Parametric Stereo (PS). It is designed to increase coding efficiency at low bitrates by using partial parametric representation of audio.

AAC+ Codec

AAC+ is the combination of the standard AAC (LC AAC – low complexity AAC) codec with Coding Technologies’ Spectral Band Replication (SBR) technology – the AAC audio codec encodes the bottom half of the audio spectrum, and SBR encodes the top half of the audio spectrum.

AAC is a transform codec, which means that blocks of input audio samples are first transformed into the frequency domain by means of a modified discrete cosine transform (MDCT), and compression takes place in the frequency domain, not the time domain.

SBR is based on the fact that there is a strong correlation between the top and bottom halves of the audio spectrum due to the presence of harmonics of the lower frequencies. SBR works by transposing the bottom half of the audio spectrum to the top half, and then modifying this top half of the spectrum so that it resembles the actual top half of the audio spectrum more closely. The SBR data only consists of the modification information, and this only requires an ultra-low bit rate channel of between 1 - 3 kbps, which is far lower than the bit rate that would be required if the top half of the audio spectrum is encoded by any other audio coding method. This is why AAC+’s official name is HE AAC – High Efficiency AAC.

SBR can enhance the efficiency of perceptual audio codecs by $30\%$ (even more in certain configurations) in the medium to low bitrate range.
**AAC+v2**

**Parametric Stereo**
The Parametric Stereo technology is the next major step to enhance the efficiency of audio compression for low bit rate stereo signals. Parametric Stereo is fully standardized in MPEG-4 and the new component within aacPlus v2. As of today, Parametric Stereo is optimized for the range of 16-40 kbps and provides high audio quality at bit rates as low as 24 kbps.

The Parametric Stereo encoder extracts a parametric representation of the stereo image of an audio signal, whereas only a monaural representation of the original signal is encoded in a conventional fashion. The stereo image information is represented as a small amount of high quality parametric stereo information and transmitted along with the monaural signal in the bit stream. Based on the parametric stereo information, the decoder is capable of regenerating the stereo image.

As a result, the perceived audio quality of a low bit rate audio bit stream of e.g. 24 kbps incorporating Parametric Stereo is significantly higher compared to the quality of a similar bit stream without Parametric Stereo.
**AAC-LC** (Low Complexity AAC) 128 kbit/s (Stereo)

High performance audio codec for excellent audio quality at low bit rates
Apple iPod
iTunes
ISDB television broadcasting (Japan)

**HE-AAC** (High Efficiency AAC) 56 kbit/s (Stereo)
High performance audio codec for good quality at bit rates of 28 kbit/s per channel and below.
XM Radio
Mobile music download
Digital Radio Mondiale

**HE-AAC v2** 48 kbit/s (Stereo)
Highest performance audio codec for good quality at bit rates below of 24 kbit/s per channel
3GPP music download  Digital radio DAB+

**HD-AAC** (High Definition AAC) Roughly half the bit rate of the uncompressed file
Lossless audio codec for better-than-CD-quality with 24 bit and up to 192 kHz
Music distribution /production

**AAC-LD** (Low Delay AAC) 128 kbit/s (Stereo)
AAC encoding with 20 ms algorithmic delay
Video conferencing
VoIP telephony

**MPEG Surround**
Surround Sound extension e.g. for AAC-LC and HE-AAC
Digital Radio in surround
Mobile TV with binaural surround sound
Music distribution 64192 kbit/s (5.1 channels)
Enhanced AACplus

---

1 usage dependent on audio mode
**MPEG-4 Audio Lossless Coding**, also known as MPEG-4 ALS, is an extension to the MPEG-4 Part 3 audio standard to allow lossless audio compression. MPEG-4 ALS defines efficient and fast lossless audio compression techniques for both professional and consumer applications. It offers many features not included in other lossless compression schemes.

* General support for virtually any uncompressed digital audio format (including wav, aiff, au, bwf, raw).
* Support for PCM resolutions of up to 32-bit at arbitrary sampling rate (including 16/44.1, 16/48, 24/48, 24/96, 24/192).
* Multi-channel / multi-track support for up to 65536 channels (including 5.1 surround).
* Support for 32-bit IEEE floating point audio data.
* Fast random access to any part of the encoded data.
* Optional storage in MP4 file format (allows multiplex with video).
* High flexibility of codec parameters for various applications.

MPEG-4 ALS is similar to FLAC in its operation. Simply put it is a quantized LPC predictor with a losslessly coded residual using Golomb Rice Coding or Block Gilbert Moore Coding (BGMC).

---

**Fig. 1. MPEG-4 ALS encoder**

**Fig. 2. MPEG-4 ALS decoder**
MPEG-4 Speech Coding: 1. HVXC

- Bit rate range: 1.2kbit/s 1.7kbit/s variable rate; 2.0kbit/s 4.0kbit/s constant rate
- Voiced frames: harmonic coding
- Pitch frequency: scalar quantization
- Spectral envelope (LPC coefficients): vector quantization
- Unvoiced frames: CELP Coding
- Variable bit rate mode: reduced bit rates for unvoiced frames and background noise

HVXC results
2. CELP

.. Narrowband and wideband mode
.. Bit rate range: 4.0kbit/s - 24kbit/s constant rate
.. Best for speech only applications and speech with background noise at low bit rates

CELP results