Chapter 13
Basic Audio Compression Techniques

13.1 ADPCM in Speech Coding

13.2 G.726 ADPCM

13.3 Vocoders

13.4 Further Exploration

13.1 ADPCM in Speech Coding

- ADPCM forms the heart of the ITU's speech compression standards G.721, G.723, G.726, and G.727.
- The difference between these standards involves the bit-rate (from 3 to 5 bits per sample) and some algorithm details.
- The default input is μ-law coded PCM 16-bit samples.

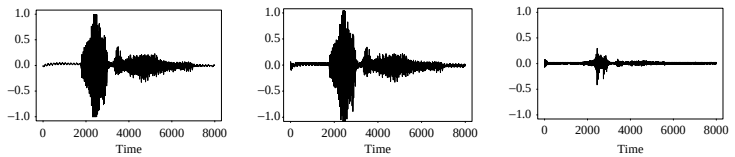

Fig. 13.1 Waveform of Word "Audio": (a) Speech sample, linear PCM at 8 kHz/16 bits per sample. (b) Speech sample, restored from G.721-compressed audio at 4 bits/sample. (c) Difference signal between (a) and (b).

13.2 G.726 ADPCM

- ITU G.726 supersedes ITU standards G.721 and G.723.
- **Rationale**: works by adapting a *fixed* quantizer in a simple way. The different sizes of codewords used amount to bit-rates of 16 kbps, 24 kbps, 32 kbps, or 40 kbps, at 8 kHz sampling rate.

- The standard defines a multiplier constant α that will change for every difference value, e_n, depending on the current scale of signals. Define a scaled difference signal g_n as follows:

$$\begin{aligned} e_n &= s_n - \widehat{s}_n \,, \\ g_n &= e_n/\alpha \,, \end{aligned} \tag{13.1}$$

$\widehat{s}_n$ is the predicted signal value. g_n is then sent to the quantizer for quantization.

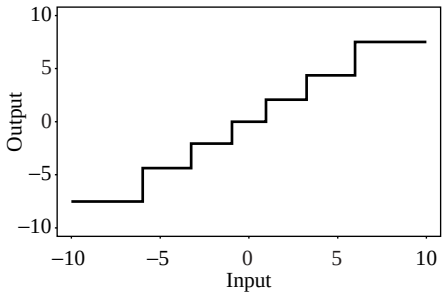

Fig. 13.2: G.726 Quantizer

- The input value is a ratio of a difference with the factor α.
- By changing α, the quantizer can adapt to change in the range of the difference signal — a *backward adaptive* quantizer.

Backward Adaptive Quantizer

- **Backward adaptive** works in principle by noticing either of the cases:
  - too many values are quantized to values far from zero – would happen if quantizer step size in f were too small.
  - too many values fall close to zero too much of the time — would happen if the quantizer step size were too large.
- **Jayant quantizer** allows *one* to adapt a backward quantizer step size after receiving just one single output.
  - Jayant quantizer simply expands the step size if the quantized input is in the outer levels of the quantizer, and reduces the step size if the input is near zero.

The Step Size of Jayant Quantizer

- Jayant quantizer assigns *multiplier values* M_k to each level, with values smaller than unity for levels near zero, and values larger than 1 for the outer levels.
- For signal f_n, the quantizer step size Δ is changed according to the quantized value k, for the previous signal value f_{n-1}, by the simple formula

$$\Delta \leftarrow M_k \Delta \tag{13.2}$$

- Since it is the *quantized* version of the signal that is driving the change, this is indeed a backward adaptive quantizer.

G.726 — Backward Adaptive Jayant Quantizer

- G.726 uses fixed quantizer steps based on the logarithm of the input difference signal, e_n divided by α. The divisor α is:

$$\beta \equiv \log_2 \alpha \tag{13.3}$$

- When difference values are large, α is divided into:
  - *locked* part α_L – scale factor for small difference values.
  - *unlocked* part α_U — adapts quickly to larger differences.
  - These correspond to log quantities β_L and β_U, so that:

$$\beta = A\beta_U + (1-A)\beta_L \tag{13.4}$$

* A changes so that it is about unity, for speech, and about zero, for voice-band data.

- The "unlocked" part adapts via the equation

$$\begin{aligned} \alpha_U &\leftarrow M_k \alpha_U \,, \\ \beta_U &\leftarrow \log_2 M_k + \beta_U \,, \end{aligned} \tag{13.5}$$

where M_k is a Jayant multiplier for the kth level.

- The locked part is slightly modified from the unlocked part:

$$\beta_L \leftarrow (1-B)\beta_L + B\beta_U \tag{13.6}$$

where B is a small number, say 2^{-6}.

- The G.726 predictor is quite complicated: it uses a linear combination of 6 quantized differences and two reconstructed signal values, from the previous 6 signal values f_n.

13.3 Vocoders

- **Vocoders** – voice coders, which cannot be usefully applied when other analog signals, such as modem signals, are in use.
  - concerned with modeling speech so that the salient features are captured in as few bits as possible.
  - use either a model of the speech waveform in time (LPC (Linear Predictive Coding) vocoding), or ... →
  - break down the signal into frequency components and model these (channel vocoders and formant vocoders).
- Vocoder simulation of the voice is not very good yet.

Phase Insensitivity

- A complete reconstituting of speech waveform is really unnecessary, perceptually: all that is needed is for the amount of energy at any time to be about right, and the signal will sound about right.
- **Phase** is a shift in the time argument inside a function of time.
  - Suppose we strike a piano key, and generate a roughly sinusoidal sound $\cos(\omega t)$, with $\omega = 2\pi f$.
  - Now if we wait sufficient time to generate a phase shift $\pi/2$ and then strike another key, with sound $\cos(2\omega t + \pi/2)$, we generate a waveform like the solid line in Fig. 13.3.
  - This waveform is the sum $\cos(\omega t) + \cos(2\omega t + \pi/2)$.

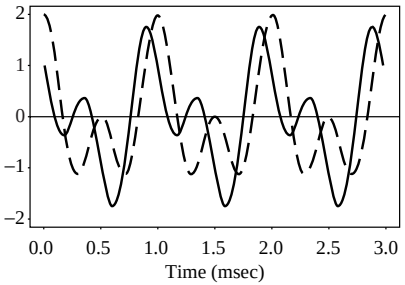

Fig. 13.3: Solid line: Superposition of two cosines, with a phase shift. Dashed line: No phase shift. The wave is very different, yet the sound is the same, perceptually.

- If we did not wait before striking the second note, then our waveform would be $\cos(\omega t) + \cos(2\omega t)$. But perceptually, the two notes would sound the same sound, even though in actuality they would be shifted in phase.

Channel Vocoder

- Vocoders can operate at low bit-rates, 1–2 kbps. To do so, a *channel vocoder* first applies a filter bank to separate out the different frequency components:

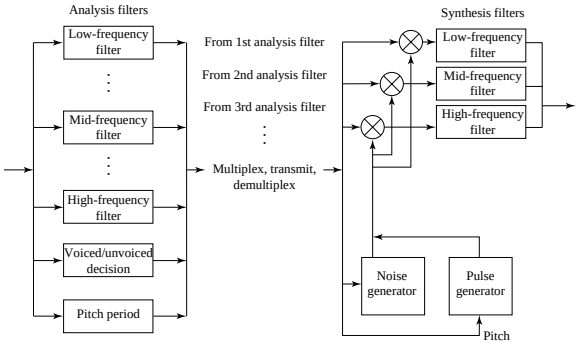

Fig 13.4: Channel Vocoder

- Due to *Phase Insensitivity* (i.e. only the energy is important):
  - The waveform is "rectified" to its absolute value.
  - The filter bank derives relative power levels for each frequency range.
  - A subband coder would not rectify the signal, and would use wider frequency bands.

- A channel vocoder also analyzes the signal to determine the general pitch of the speech (low — bass, or high — tenor), and also the *excitation* of the speech.

- A channel vocoder applies a vocal tract transfer model to generate a vector of excitation parameters that describe a model of the sound, and also guesses whether the sound is *voiced* or *unvoiced*.

Format Vocoder

- **Formants:** the salient frequency components that are present in a sample of speech, as shown in Fig 13.5.
- Rationale: encoding only the most important frequencies.

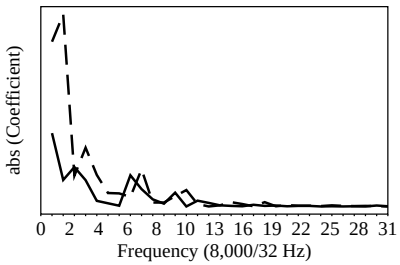

Fig. 13.5: The solid line shows frequencies present in the first 40 msec of the speech sample in Fig. 6.15. The dashed line shows that while similar frequencies are still present one second later, these frequencies have shifted.

Linear Predictive Coding (LPC)

- **LPC vocoders** extract salient features of speech directly from the waveform, rather than transforming the signal to the frequency domain
- **LPC Features:**
  - uses a time-varying model of vocal tract sound generated from a given excitation
  - transmits only a set of parameters modeling the shape and excitation of the vocal tract, not actual signals or differences ⇒ small bit-rate
- About **"Linear":** The speech signal generated by the output vocal tract model is calculated as a function of the current speech output plus a second term linear in previous model coefficients

LPC Coding Process

- **LPC** starts by deciding whether the current segment is voiced or unvoiced:
  - For unvoiced: a wide-band noise generator is used to create sample values $f(n)$ that act as input to the vocal tract simulator
  - For voiced: a pulse train generator creates values $f(n)$
  - Model parameters a_i: calculated by using a least-squares set of equations that minimize the difference between the actual speech and the speech generated by the vocal tract model, excited by the noise or pulse train generators that capture speech parameters

LPC Coding Process (cont'd)

- If the output values generate $s(n)$, for input values $f(n)$, the output depends on p previous *output* sample values:

$$s(n) = \sum_{i=1}^{p} a_i s(n-i) + Gf(n) \qquad (13.7)$$

G – the "gain" factor coefficients; a_i – values in a linear predictor model

- LP coefficients can be calculated by solving the following minimization problem:

$$\min E\{[s(n) - \sum_{j=1}^{p} a_j s(n-j)]^2\} \qquad (13.8)$$

LPC Coding Process (cont'd)

- By taking the derivative of a_i and setting it to zero, we get a set of J equations:

$$E\{[s(n) - \sum_{j=1}^{p} a_j s(n-j)]s(n-i)\} = 0, \qquad i = 1...p, \quad (13.9)$$

- Letting $\phi(i,j) = E\{s(n-i)s(n-j)\}$, then:

$$\begin{bmatrix} \phi(1,1) & \phi(1,2) & ... & \phi(1,p) \\ \phi(2,1) & \phi(2,2) & ... & \phi(2,p) \\ .. & ... & ... & ... \\ \phi(p,1) & \phi(p,2) & ... & \phi(p,p) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ ... \\ a_p \end{bmatrix} = \begin{bmatrix} \phi(0,1) \\ \phi(0,2) \\ ... \\ \phi(0,p) \end{bmatrix}, \qquad (13.10)$$

LPC Coding Process (cont'd)

- An often-used method to calculate LP coefficients is the autocorrelation method:

$$\phi(i,j) = \sum_{n=p}^{N-1} s_w(n-i)s_w(n-j) / \sum_{n=p}^{N-1} s_w^2(n), \quad i = 0...p,\ j = 1...p. \tag{13.11}$$

$s_w(n) = s(n+m)w(n)$ – the windowed speech frame starting from time m

LPC Coding Process (cont'd)

- Since $\phi(i,j)$ can be defined as $\phi(i,j) = R(|i-j|)$, and when $R(0) \geq 0$, the matrix $\{\phi(i,j)\}$ is positive symmetric, there exists a fast scheme to calculate the LP coefficients:

$E(0) = R(0),\ i = 1$
while $i \leq p$
  $k_i = [R(i) - \sum_{j=1}^{i-1} a_j^{i-1} R(i-j)]/E(i-1)$
  $a_i^{i-1} = k_i$
  for $j = 1$ to $i-1$
    $a_j^i = a_j^{i-1} - k_i a_{i-j}^{i-1}$
  $E(i) = (1-k_i^2)E(i-1)$
  $i \leftarrow i+1$
for $j = 1$ to p
  $a_j = a_j^J$

LPC Coding Process (cont'd)

- Gain G can be calculated:

$$\begin{aligned} G &= E\{[s(n) - \textstyle\sum_{j=1}^{p} a_j s(n-j)]^2\} \\ &= E\{[s(n) - \textstyle\sum_{j=1}^{p} a_j s(n-j)]s(n)\} \\ &= \phi(0,0) - \textstyle\sum_{j=1}^{p} a_j \phi(0,j) \end{aligned} \tag{13.12}$$

- The pitch P of the current speech frame can be extracted by correlation method, by finding the index of the peak of:

$$v(i) = \sum_{n=m}^{N-1+m} s(n)s(n-i) / [\sum_{n=m}^{N-1+m} s^2(n) \sum_{n=m}^{N-1+m} s^2(n-i)]^{1/2} \tag{13.13}$$
$$i \in [P_{\text{min}}, P_{\text{max}}]$$

Code Excited Linear Prediction (CELP)

- **CELP** is a more complex family of coders that attempts to mitigate the lack of quality of the simple LPC model
- CELP uses a more complex description of the excitation:
  - An entire set (a codebook) of excitation vectors is matched to the actual speech, and the index of the best match is sent to the receiver
  - The complexity increases the bit-rate to 4,800-9,600 bps
  - The resulting speech is perceived as being more similar and continuous
  - Quality achieved this way is sufficient for audio conferencing

The Predictors for CELP

- CELP coders contain two kinds of prediction:
  - LTP (Long time prediction): try to reduce redundancy in speech signals by finding the basic periodicity or pitch that causes a waveform that more or less repeats
  - STP (Short Time Prediction): try to eliminate the redundancy in speech signals by attempting to predict the next sample from several previous ones

Relationship between STP and LTP

- STP captures the formant structure of the short-term speech spectrum based on only a few samples
- LTP, following STP, recovers the long-term correlation in the speech signal that represents the periodicity in speech using whole frames or subframes (1/4 of a frame)
  - LTP is often implemented as "adaptive codebook searching"
- Fig. 13.6 shows the relationship between STP and LTP

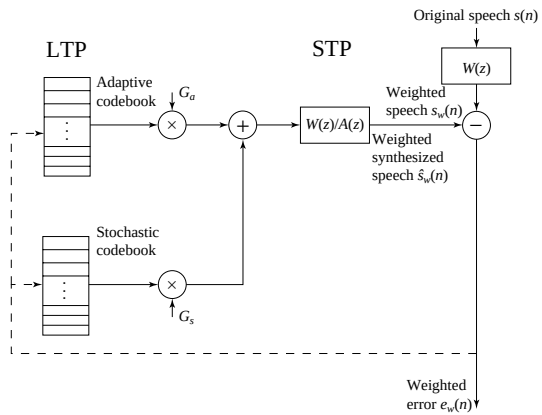

Fig 13.6 CELP Analysis Model with Adaptive and Stochastic Codebooks

Adaptive Codebook Searching

- **Rationale:**
  - Look in a codebook of waveforms to find one that matches the current subframe
  - *Codeword:* a shifted speech residue segment indexed by the by the lag τ corresponding to the current speech frame or subframe in the adaptive codebook
  - The gain corresponding to the codeword is denoted as g_0

Open-Loop Codeword Searching

- Tries to minimize the long-term prediction error but not the perceptual weighted reconstructed speech error,

$$E(\tau) = \sum_{n=0}^{L-1} [s(n) - g_0 s(n-\tau)]^2 . \quad (13.14)$$

By setting the partial derivative of g_0 to zero, $\partial E(\tau)/\partial g_0 = 0$, we get

$$g_0 = \frac{\sum_{n=0}^{L-1} s(n)s(n-\tau)}{\sum_{n=0}^{L-1} s^2(n-\tau)} , \quad (13.15)$$

and hence a minimum summed-error value

$$E_{\min}(\tau) = \sum_{n=0}^{L-1} s^2(n) - \frac{[\sum_{n=0}^{L-1} s(n)s(n-\tau)]^2}{[\sum_{n=0}^{L-1} s(n-\tau)]^2} \quad (13.16)$$

LZW Close-Loop Codeword Searching

- Closed-loop search is more often used in CELP coders — also called *Analysis-By-Synthesis* (A-B-S)
- speech is reconstructed and perceptual error for that is minimized via an adaptive codebook search, rather than simply considering sum-of-squares
- The best candidate in the adaptive codebook is selected to minimize the distortion of locally reconstructed speech
- Parameters are found by minimizing a measure of the difference between the original and the reconstructed speech

Hybrid Excitation Vocoder

- **Hybrid Excitation Vocoders** are different from CELP in that they use model-based methods to introduce multi-model excitation
- includes two major types:
  - MBE (Multi-Band Excitation): a blockwise codec, in which a speech analysis is carried out in a speech frame unit of about 20 msec to 30 msec
  - MELP (Multiband Excitation Linear Predictive) speech codec is a new US Federal standard to replace the old LPC-10 (FS1015) standard with the application focus on very low bit rate safety communications

MBE Vocoder

- MBE utilizes the A-B-S scheme in parameter estimation:
  - The parameters such as basic frequency, spectrum envelope, and sub-band U/V decisions are all done via closed-loop searching
  - The criterion of the closed-loop optimization is based on minimizing the perceptually weighted reconstructed speech error, which can be represented in frequency domain as

$$\varepsilon = \frac{1}{2\pi} \int_{-\pi}^{+\pi} G(\omega)|S_w(\omega) - S_{wr}(\omega)|d\omega \quad (13.29)$$

$S_w(\omega)$ – original speech short-time spectrum
$S_{wr}(\omega)$ – reconstructed speech short-time spectrum
$G(\omega)$ – spectrum of the perceptual weighting filter

MELP Vocoder

- **MELP:** also based on LPC analysis, uses a multiband soft-decision model for the excitation signal
- The LP residue is bandpassed and a voicing strength parameter is estimated for each band
- Speech can be then reconstructed by passing the excitation through the LPC synthesis filter
- Differently from MBE, MELP divides the excitation into five fixed bands of 0-500, 500-1000, 1000-2000, 2000-3000, and 3000-4000 Hz

MELP Vocoder (Cont'd)

- A voice degree parameter is estimated in each band based on the normalized correlation function of the speech signal and the smoothed rectified signal in the non-DC band
- Let $s_k(n)$ denote the speech signal in band k, $u_k(n)$ denote the DC-removed smoothed rectified signal of $s_k(n)$. The correlation function:

$$R_x(P) = \frac{\sum_{n=0}^{N-1} x(n)x(n+P)}{[\sum_{n=0}^{N-1} x^2(n) \sum_{n=0}^{N-1} x^2(n+P)]^{1/2}} \qquad (13.33)$$

P – the pitch of current frame
N – the frame length
k – the voicing strength for band (defined as $\max(R_{s_k}(P), R_{u_k}(P))$)

MELP Vocoder (Cont'd)

- MELP adopts a jittery voiced state to simulate the marginal voiced speech segments – indicated by an aperiodic flag
- The jittery state is determined by the peakiness of the full-wave rectified LP residue $e(n)$:

$$peakiness = \frac{[\frac{1}{N}\sum_{n=0}^{N-1} e(n)^2]^{1/2}}{\frac{1}{N}\sum_{n=0}^{N-1} |e(n)|} \qquad (13.34)$$

- If peakiness is greater than some threshold, the speech frame is then flagged as jittered

13.4 Further Exploration

⟶ Link to Further Exploration for Chapter 13.

- A comprehensive introduction to speech coding may be found in Spanias's excellent article in the textbook references.
- Sun Microsystems, Inc. has made available the code for its implementation on standards G.711, G.721, and G.723, in C. The code can be found from the link in the Chapter 13 file in the Further Exploration section of the text website
- The ITU sells standards; it is linked to from the text website for this Chapter
- More information on speech coding may be found in the speech FAQ files, given as links on the website; also, links to where LPC and CELP code may be found are included