Mobile Systems

Workshop 1

Narrow band speech coding for mobile phones

COMP28512

Steve Furber & Barry Cheetham

Speech coding for mobile phones

- 64 kbit/s too high for mobile phones.
- Originally, they used 24.7 kbit/s , but this included extra bits for correcting bit-errors.
- Bit-rate available for narrow-band speech ≈ 13 kbit/s.
- More recently, AMR speech coder is used.
- Encodes narrowband speech at bit-rates ranging from 4.75 to 12.2 kbit/s.
- 'Toll' quality speech is achieved at 12.2 kbit/s.
- How?

AMR file recorded on a mobile phone (≈10s, 16kBytes) Converted to wav.

The problem

- If Fs = 8 Hz, only have 1.5 bits per sample at 12 kb/s.
- If we reduce Fs to 4 kHz, can have 3 bits/sample
- But we must filter off all sound above 2 kHz.
- Speech will sound 'muffled'
- And 3 bits/sample is still not enough for uniform quantisation with reasonable accuracy.

Operator_short_orig 8kHz unif 12 bits/sample

Lowpass filtered <2kHz)

LP filtered (<1kHz)

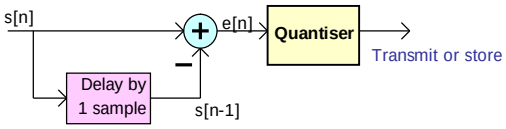
Operator_long_orig 8kHz uniform 12 bits/sample

Low-pass filtered (<2 kHz)

Low-pass filtered (<1kHz)

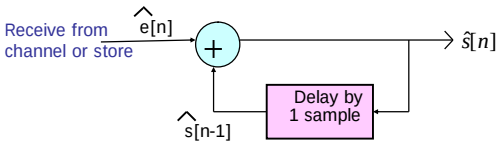
An idea: differential encoding

- Encode differences between samples:



- If successive samples are close together, differences e[n] will be smaller & easier to encode.
- If we send differences, can we get back to orig speech?

Differential decoder



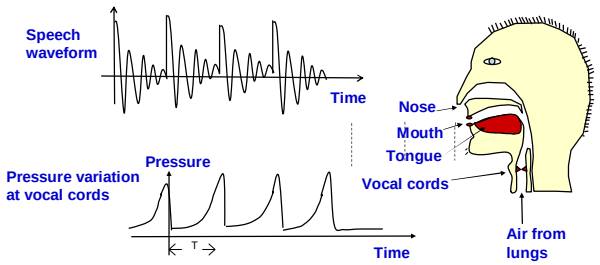
- Can be made to work at bit-rates of 32 & 16 kbit/s
- Lots of quite complicated tricks needed
- Known as G726-ADPCM (adaptive differential PCM)
- Still not good enough for mobile phones
- Any other ideas?

Waveform & parametric speech coders

- So far, all our ideas have tried to encode the shape of the signal (its 'waveform') by sampling it.
- These are 'waveform coders
- Simple, but no good at very low bit-rates.
- Parametric coders try to model the human speech production process:
  - –the vibration of the vocal cords, and
  - –the shape of the 'vocal tract' (mouth, lips etc)
- These do not change as quickly as the wave-form
- So can be sampled more slowly without losing accuracy.

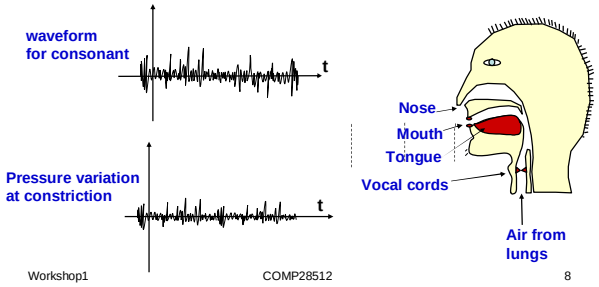
Human speech production: voiced

- Air-flow causes vocal cords to open & close periodically.
- Causes pressure variation (sound).
- Modified (filtered) by vocal tract to produce vowel sound.

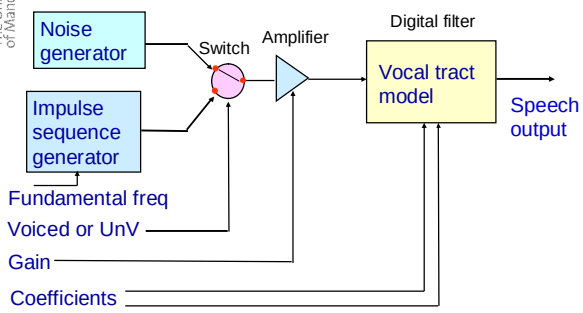


Human speech production: unvoiced

- Vocal cords held open – no vibration.
- Constriction in air-flow creates 'turbulence'.
- Turbulent flow is chaotic & random – sounds like white noise



Model of human speech production



Linear predictive speech coding (LPC)

- Similar model implemented in all mobile phones.
- Sender & receiver
- Sender derives parameters every 20 ms (50 times/s):
  - Coeffs of a digital filter which models effect of the vocal tract
  - How loud it the speech is
  - Whether it is voiced or unvoiced
  - If voiced, measures the fundamental frequency
- These parameters are sent to represent the speech.
- Receiver reconstructs speech from these parameters.

LPC-10

- 2400 b/s coder once widely used in military comms.
- Not used in mobile phones
- Encodes 20 ms speech frames by deriving:
  - 10 digital filter coeffs by LPC analysis,
  - unvoiced/voiced decision (1 bit)
  - gain (or amplitude): a single number: 8 bits say
  - fundamental frequency: a single number: 8 bits say)
- Each frame has 48 bits which affords 37 bits for the 10 digital filter coeffs.
- Quite simple to understand and implement,
- But its quality is far from good (run & listen to demo)

LPC10 demo

- Simple LPC10 encoder & decoder written in MATLAB & Python.
- Encoder derives filter coeffs & vocal tract excitation for 20ms segments.
  - Voiced' & 'excitation freq' determined by very simple method
  - Stores parameters in a file
- Decoder implements the model
  - uses parameters read from file
- Run the encoder, but don't worry abt how it works yet.
- Then run the decoder, & modify the code to investigate:
  - what happens if it is forced to be voiced with constant pitch-period?
  - what happens if the speech is always forced to be unvoiced?

Orig
modoperamp.wav

PP=25 samples
LPC10SpeechNPP25.wav

PP=50 samples
LPC10SpeechNPP50.wav

PP=100 samples
LPC10SpeechNPP100.wav

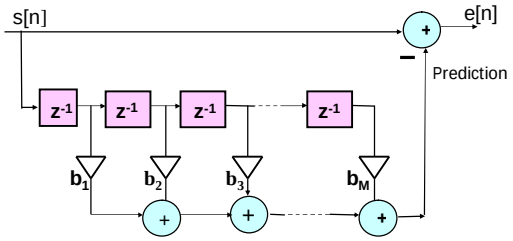
Unvoiced
LPC10SpeechWsp.wav

LPC10
LPC10SpeechPP.wav

More on linear prediction (LP)

- How are the coeffs b_1, b_2, ..., b_M calculated at the encoder?
- Need new set of coeffs for each block of speech: $\{s[n]\}_{1,N}$
- Code & transmit e[n] & coeffs b_1, b_2, ..., b_M for each block.
- Typically M=10.

Prediction filter



$e[n] = s[n] - b_1 s[n-1] - b_2 s[n-2] - \ldots - b_M s[n-M]$

For each block of speech $\{s[n]\}_{1,N}$ choose b_1, b_2, ... b_M to minimise: $E = \sum_{n=1}^{\infty} (e[n])^2$

Finding b_1, b_2, ... b_M

$$E = \sum_{n=1}^{\infty} (e[n])^2 = \sum_{n=1}^{\infty} (s[n] - b_1 s[n-1] - \ldots - b_M s[n-M])^2$$

$$\frac{\partial E}{\partial b_1} = \sum_{n=1}^{\infty} (s[n] - b_1 s[n-1] - \ldots - b_M s[n-M])\, s[n-1]$$

$$= \quad C_{10} - C_{11} b_1 - C_{12} b_2 - \ldots - C_{1M} b_M$$

where $C_{ij} = \sum_{n=1}^{\infty} s[n-i] s[n-j]$

Similarly, $\frac{\partial E}{\partial b_2} = C_{20} - C_{21} b_1 - C_{22} b_2 - \ldots - C_{2M} b_M$

$$\frac{\partial E}{\partial b_3} = C_{30} - C_{31} b_1 - C_{32} b_2 - \ldots - C_{3M} b_M$$

and so on

Finding b_1, b_2, ... b_M (cont)

Setting $\frac{\partial E}{\partial b_1} = 0, \ \frac{\partial E}{\partial b_2} = 0, \ \ldots, \ \frac{\partial E}{\partial b_M} = 0$ we get :

$$C_{10} = C_{11} b_1 + C_{12} b_2 + \ldots + C_{1M} b_M$$
$$C_{20} = C_{21} b_1 + C_{22} b_2 + \ldots + C_{2M} b_M$$
$$C_{30} = C_{31} b_1 + C_{32} b_2 + \ldots + C_{3M} b_M \quad \text{and so on}$$

$$\begin{bmatrix} C_{10} \\ C_{20} \\ \vdots \\ C_{M0} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \ldots & C_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ C_{M1} & C_{M2} & \ldots & C_{MM} \end{bmatrix} \times \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_M \end{bmatrix}$$

$\underline{c} = [C] \times \underline{b}$

$\therefore \underline{b} = [C]^{-1} \times \underline{c}$

That's it!!

In MATLAB

```
for i=1:M
  c(i)=0;
  for n=1:N, if n>i, c(i) = c(i)+s(n)*s(n-i); end;  end;
  for j=1:M
    C(i,j)=0;
    for n = j+1 : j+N
      if n > i && n < N+i+1, C(i,j) = C(i,j)+s(n-i)*s(n-j); end;
    end; % n loop
  end; % j loop
end; % i loop
b = inv(C)*c';
```


Voiced & Unvoiced speech

•Voiced speech (vowels):

- Resonances (formants) which change as person speaks.
- These determine the vowel sound.

• Unvoiced speech (consonants):

- Vocal cords do not vibrate.
- Turbulent air flow produces "hissing" sound.
- Vocal tract excitation is random noise-like signal.

Effect of prediction at the transmitter

- With correctly adapted coeffs, subtracting prediction at the transmitter removes resonances (formants).
- Remaining 'prediction error' (or 'residual') signal {e[n]} becomes high-pass filtered excitation signal:
  - periodic series of pulses (voiced),
    or
  - spectrally white random signal (unvoiced).

Vector-quantisation of e[n]

- Instead of transmitting e[n] sample-by-sample, send several samples at once as a 'vector'.
- Store frequently occurring patterns in code-books at transmitter & receiver.
- Transmitter chooses pattern closest to the one it needs to transmit, & just sends its code-book index.
- This is 'code-book' quantisation.
- Idea like this is used in Code excited LPC (CELP).

Codebook excited LPC (CELP)

- Instead of having fixed noise & impulse generators, CELP model has a selection of different ones stored in a code-book.
- Sender tells it which ones to use for each frame.
- Just needs to send a code-book index.
- Sender has a copy of the code-book & uses 'analysis-by synthesis' to find the best excitation to use.
- Tries them one-by one & compares what they produce with the original speech.

LPC10, CELP & AMR

- LPC-10 coder at 2400 b/s used in military comms.
- CELP is used in mobile telephony (13 kbit/s & lower).
- Adaptive multi-rate (AMR) coder uses CELP at various bit-rates:

  4.75 … 7.4 , 7.95, 10.2, 12.2 kbit/s
- Uses vector (code-book) quantisation.
- Better quality than LPC10.

Waveform & parametric coding.

- Waveform coding techniques such as PCM & ADPCM try to preserve exact shape of waveform as far as possible.
- Simple to understand & implement, but cannot achieve very low bit-rates.
- Parametric techniques such as LPC10 & CELP do not aim to preserve exact wave-shape.
- Instead they represent features expected to be perceptually significant by sets of parameters,
  i.e. by filter coeffs & params of excitation signal.
- Parametric more complicated to understand & implement than waveform coding, but achieves lower bit-rates.

'Comfort noise'

- In a 2-way telephone conversation each person may be listening or waiting about 60% of the time.
- Discontinuous transmission (DTX) is an option for not transmitting 'silence'.
  - Saves transmission power but receiver's phone may sound 'dead.'
  - No background noise heard.
- So receiver inserts some artificial background noise.
- Needs 'voice activity detector' (VAD) at transmitter.
  - Determines when talker is 'silent'
  - Characterises the background noise by some basic measurements.
  - Transmits these measurements (e.g. power) using very few bits.
- Allows receiver to synthesise 'comfort noise' that sounds approximately like the background noise at the transmitter.

Laboratory (Reminder)

- Task 1 ends on Thursday 5 Feb 2015
- Submit to MOODLE notebook or zipped folder containing:
  - Your Python code
  - Short report showing & evaluating your results
  - Deadline is 11:55pm
- Please book a slot during the lab session to:
  - demonstrate your code
  - answer a few questions
- Task 2 will be available on Tuesday 10 Feb

Summary

- Differential coding & ADPCM.
- Modelling human speech mechanism .
- Concept of linear prediction coding (LPC)
- Closely based on characteristics of human voice.
- LPC10 illustrated.
- CELP (as used in mobile telephony).
- Waveform & parametric speech coding.