Mobile Systems

Lecture 2– analogue & digital signals

COMP28512
Steve Furber & Barry Cheetham

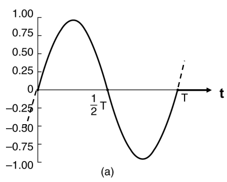
Signals in mobile systems

- Real world signals...
  - sound: speech, music, ...
  - vision: photos, natural scenes, ...
  - radio transmissions
  - taste, smell, touch?
    (we won't be worrying about these!)
- All arrive as physical quantities that vary continuously over time and space.
- Such signals can carry an infinite amount of information
- How can we represent & process these digitally?

Laboratory (Week 2)

- Laboratory Task 1 ends this week.
- Please submit, to MOODLE, a folder or notebook with:
  - Your Python code
  - A short report showing & summarising results
  - Deadline is 11:55pm Friday 6 Feb
- 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.

Analogue signal: sine-wave



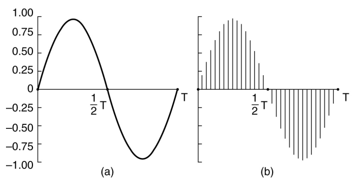
For a sin-wave of amplitude 1 & period T seconds:

$$y = \sin(2\pi t / T)$$

Its frequency is *1/T* Hz.

- Continuous in time – value can be measured at any point in time
- Continuous in value - takes all real values between ±1

Sampling an analogue signal



- Measure the value of the signal at regular points in time
  - now becomes a discrete time signal
  - amplitude is still continuous

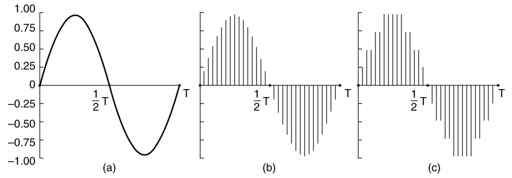
Sampling theorem

- What have we lost?
- Answer:
  - nothing provided we sample frequently enough

Henry Nyquist 1889-1976

- If the sampling frequency is more than twice the maximum frequency we want to capture, nothing is lost.
- If we know the sampled values exactly, we can always recreate original analogue signal exactly from these samples
- This is 'Sampling Theorem' or 'Nyquist criterion'

Quantisation



Analogue Sampled Quantised

- Limited number of bits available for storing each sample
- So each sample is ***quantised.***
- Sine-wave becomes a series of integers:
  0 1 2 2 3 3 4 4 4 4 4 3 3 2 2 1 0 -1 -2 -2 -3 -3 ...

Quantisation noise

- Quantisation introduces errors in sample values.
- So, in practice we will not know these values exactly.
- With many bits per sample (e.g. 16) the error will be small.
- But this requires high storage or transmission capacity.
- If we reduce the number of bits per sample,
  error becomes larger & it becomes noticeable.
- In sound recordings or transmissions,we hear it as 'quantisation noise'
- In pictures, we see it a quality degradation.

Aliasing

- Suppose we use a sampling frequency of *Fs* Hz
- This is good for input frequencies up to *Fs/2* Hz
- What happens to higher frequencies?
- They should not be there – should have been filtered out.
- Assume a sine-wave of frequency *F* > *Fs*/2 remains.
- After sampling, it becomes a sine-wave of freq *Fs* – *F*.
  (Assume *Fs*/2 < *F* < *Fs*)
- Example: If *Fs* = 8 *kHz* & *F* = 5 kHz, we get 3 kHz.
- It's the wrong frequency.
- If it's a harmonic within a musical note, will be out of tune.
- And it goes down when it's supposed to be going up.

Demonstration of aliasing

- Simple experiment in Task 1
- Do not use 'resample' for this experiment.
- Listen to this demo.

Orig

Aliased

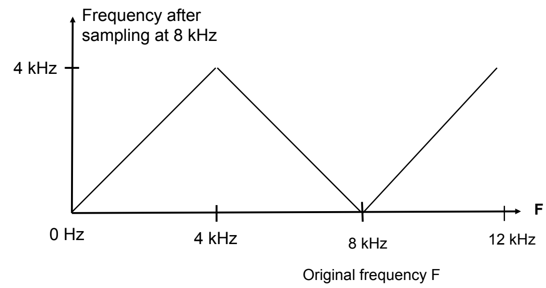
- Generates C-major piano scale over 4 octaves (orig file)
- From 261.6 Hz (middle C) up to 2093 Hz
- Down-sampled (no filtering) to Fs = 2kHz (aliased file).
- Two wav files: orig & aliased.
- Observe in aliased file:
  (1) out of tune harmonics > 1 kHz
  (2) Note starts decreasing as fundamental freq goes above 1kHz

Effect of aliasing on sine-waves

- Assume *Fs* = 8 kHz

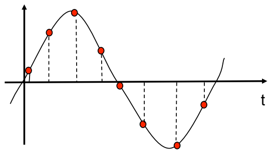
Before sampling	After sampling
1 kHz	1 kHz
3 kHz	3 kHz
4 kHz	strange effects
5 kHz	3 kHz
7 kHz	1 kHz
9 kHz	1 kHz
11 kHz	3 kHz
etc.	

Effect expressed as a graph



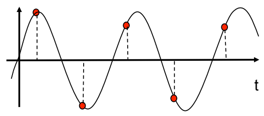
Explanation of aliasing 1

- Consider 1 kHz sine-wave sampled at 8 kHz:



- 8 samples per cycle.
- Easy to reconstruct sine-wave

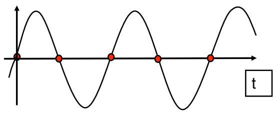
- Now consider 3.9 kHz sine-wave sampled at 8 kHz:



- Just over 2 samples/cycle
- Can still reconstruct sine-wave (just about)

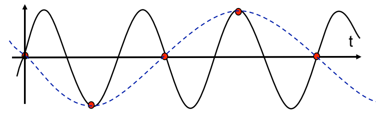
Explanation of aliasing 2

- Now consider 4 kHz sine-wave sampled at 8 kHz:



- With exactly 2 samples/cycle
- This can happen – no sine-wave

- Finally, consider 6 kHz sine-wave sampled at 8 kHz:



- Less than 2 samples per cycle
- Looks like lower frequency sine-wave

(Becomes 4 sample/cycle = 4 kHz)

Decibel (dB) scale

- I can shout twice as loud as you!
- Power of my sound in Watts is twice your power
- My voice is 3 dB louder than yours

$$\text{Power ratio (dB)} = 10 \times \log_{10}\left(\frac{\text{Power of my voice (Watts)}}{\text{Power of your voice (Watts)}}\right)$$

$$= 10 \times \log_{10}(2) = 10 \times 0.3 = 3 \text{ dB}$$

Power ratio	dB	Power ratio	dB
1	0	1000	30
2	3	10000	40
1/2	-3	10^5	50
4	6	10^{10}	100
10	10	10^{12}	120
100	20		

CD quality digital audio

- Humans can hear sound over a frequency range 20 Hz to 20 kHz
- Therefore CD sampling frequency is 44.1 kHz
- Dynamic range of human hearing is power ratio of loudest audible sound we wish to hear (without risking hearing damage) to the quietest sound we can hear.

$$Dy(dB) = 10 \times \log_{10}\left(\frac{\text{power of loudest sound we wish to hear}}{\text{power of quietest sound we can hear}}\right)$$

- It is about 120 dB, which is a power ratio of 10^{12} Wow!
- Power $\propto$ mean squared voltage as obtained from a microphone.
- How many bits/sample do we need?
- With uniform quantisation (see later), we get $\approx$ 6 dB per bit. (discuss).
- For 120 dB, we need $\approx$ 20 bits – too many when CDs were invented.
- Settled for 16 bits per sample & the use of dy range compression (DCR).
- Make quieter parts louder – then we can turn volume down at home.
- DCR is necessary, controversial & good commercially. LOUD SELLS!!
- CD data rate (stereo) = 16 x 44,100 x 2 = 1,411 kbit/s

Telephone quality digital speech

- Band-limited from 300 Hz – 3.4 kHz – narrow-band. (or 50 Hz – 3.4 kHz)
- Loses naturalness but not intelligibility (in principle)
- In practice, sometimes cannot distinguish "S" from "F"!
- Sampled at 8 kHz with 8 bits per sample
- 64 kbit/s bit-rate, but needs non-uniform quantization
  - mu-law or A-law
  - known as the ITU-G711 standard

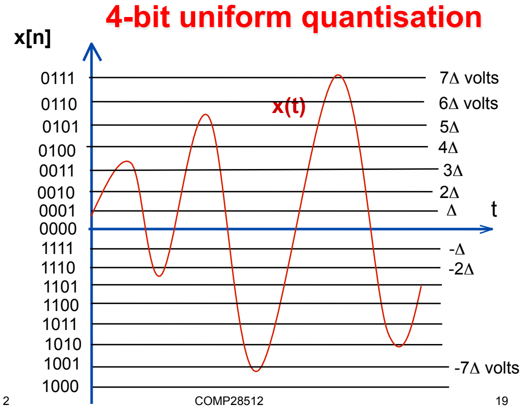
Uniform quantisation

- Each sample of speech x(t) is represented by a binary number x[n].
- Each binary number represents a voltage.
- Constant voltage difference Δ between the voltages for adjacent binary numbers:
- e.g. between 0001 and 0010
- Call delta (Δ) the quantisation step-size

MANCHESTER 1824

The University of Manchester

4-bit uniform quantisation



Lecture 2 COMP28512 19

MANCHESTER 1824

The University of Manchester

Problem with uniform quantisation

- Problem is choosing the right value of delta (Δ)

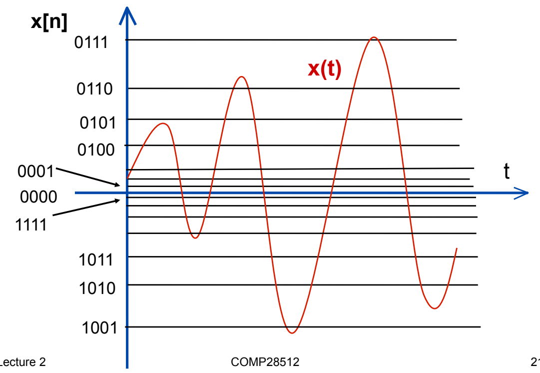
- One solution is to adjust Δ according to amplitude of sample.
  - For larger amplitudes use larger Δ.
  - Make 'step-size' Δ change from sample to sample.

Lecture 2 COMP28512 20

MANCHESTER 1824

The University of Manchester

4-bit non-uniform quantisation



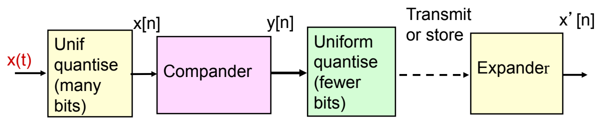
Lecture 2 COMP28512 21

MANCHESTER 1824

The University of Manchester

Implementation

- Digitise **x(t)** accurately with uniform quantisation to give x[n].
- Apply 'companding' formula to x[n] to give y[n].
- Uniformly quantise y[n] using fewer bits
- Store or transmit the companding result.
- Pass it thro' **expander to** reverse effect of companding.



Lecture 2 COMP28512 22

MANCHESTER 1824

The University of Manchester

Effect of compander & expander

- Compander increases smaller amplitudes of x[n] & reduces larger ones.
- Uniform quantiser is then applied with fixed Δ.
- Expander decreases smaller amplitudes of x[n] & also Δ, & increases larger ones along with Δ.
- Effect is non-uniform quantisation as illustrated before.

- Famous companding formulas: A-law & Mu-law (G711)
- These normally require 8-bits per sample.
- Companding is like 'compression', but it is done for coding purposes, not for listening to directly.

Lecture 2 COMP28512 23

MANCHESTER 1824

The University of Manchester

More on quantisation error

- Uniform quantisation produces error in each sample.
- Random in range ±Δ/2 (assuming no overflow).
- When samples are converted back to analogue form, error is heard as 'white noise' sound added to x(t).
- 'Noise' is an unwanted signal.
- White noise is spread evenly across all frequencies.
- Sounds like a waterfall or the sea.
- Not a car or house alarm, or a car revving its engine.
- Error in each sample has uniform probability between ±Δ/2.
- Mean square value (MSV) of noise is:

$$\Delta^2 / 12 \qquad \text{(famous result)}$$

- MSV is a measure of power

Lecture 2 COMP28512 24

Signal-to-quantisation noise ratio (SQNR)

$$\text{SQNR} = 10\log_{10}\left(\frac{\text{signal power}}{\text{quantisation noise power}}\right) \quad \text{in decibels (dB.)}$$

For a sine-wave of amplitude A, its MSV is $A^2/2$.

$$\therefore \ \text{SQNR} = 10\log_{10}\left(\frac{A^2/2}{\Delta^2/12}\right) \quad \text{dB}$$

With NB bits/sample unif quantisation & step size Δ, what is max possible value of A ? Answer: $(2^{NB}/2) \times \Delta$

$$\therefore \ \text{SQNR} = 10\times\log_{10}\left(\frac{(2^{2\times NB}/8)\times\Delta^2}{\Delta^2/12}\right) = 10\times\log_{10}(1.5\times 2^{2\times NB}) \quad \text{dB}$$

$$= 10\times\log_{10}(1.5) + 20\times NB\times\log_{10}(2) \approx 1.8 + 6\times NB \quad \text{dB}$$

= 6 dB per bit + 1.8 (Another famous result)

Lecture 2 COMP28512 25

Comments on previous result

- Applies to uniform quantisation (not A-law, mu-Law)
- Applies strictly to sine-waves, but is approximately true for other wave-shapes such as speech & music.
- With 8 bits/sample, unif quantisation & fixed Δ,
  SQNR for loudest talkers is: 6 x 8 + 1.8 = 49.8 dB
- OK for loudest talkers.
- What about a talker who is quieter by 30 dB?
- His value of A will use only 3 bits & SQNR = 19.8 dB
- Too low – you will hear quantisation noise
- 8 bits/sample not enough with uniform quantisation.
- Need A-Law or Mu Law non-uniform quantisation.

Lecture 2 COMP28512 26

CD recordings with 16 bits/sample

- Max SQNR = 6x16 + 1.8 = 97.8 dB.
- This is for LOUDest music.
- What abt quiet passages that we can just hear?
- Lower by 120 dB maybe?
- SQNR = -22.2 dB ??
  - Signal power less than quant noise power by a long way.
- Must apply DRC to CD recordings with 16 bits/sample.
- Do we really want 120 dB in our homes/cars or through ear-phones connected to mobile phones?

Lecture 2 COMP28512 27

Speech & music on mobile phones

- 64 kbits/second still too high for mobile telephony
- Need to encode speech at around 13 kb/s or lower.
- How can we do this?
- If we sample at 8 kHz, have <2 bits/sample. No good.
- If we reduce sampling rate, bandwidth will be too low.
- What can we do?
- LPC coding (See Workshop 1 - next)
- 1.411 Mbits/s CD quality too high for music,
- Need mp3 coding
- Come to Workshop 3 (next month)

Lecture 2 COMP28512 28

Summary

- Digitising analogue signals: sampling & quantisation
- Effect of aliasing
- Dynamic range required for speech & music
- Uniform quantisation noise power: $\Delta^2/12$
- SQNR = 6m + 1.8 dB (m bits)
- Non-uniform quantisation (A-Law & mu-Law).

Lecture 2 COMP28512 29

A question

- What are similarities & differences difference between companding & compression as defined in this lecture?
- Answer: Both reduce high amplitudes and/or increase small amplitudes.
- Companding does this sample by sample for the purpose of digitising the signal. The companding is reversed before we listen to the signal.
- Compression does this gradually over many samples to reduce loud sections and/or increase quiet sections. It does not work sample by sample and the compression is not reversed before we listen to it.

Lecture 2 COMP28512 30