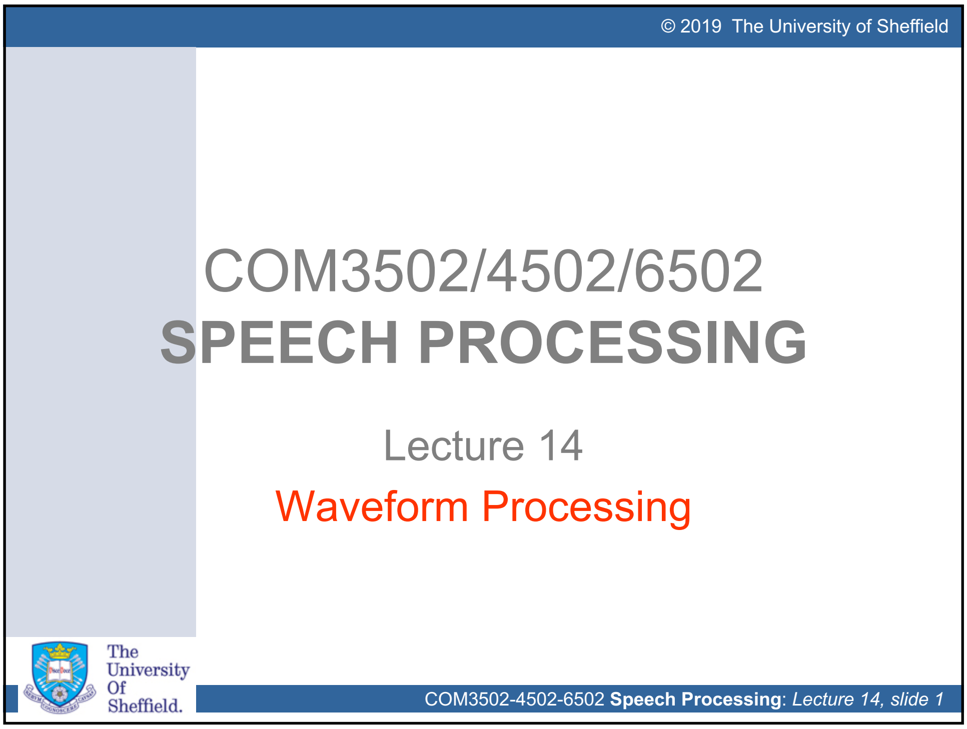


© 2019 The University of Sheffield

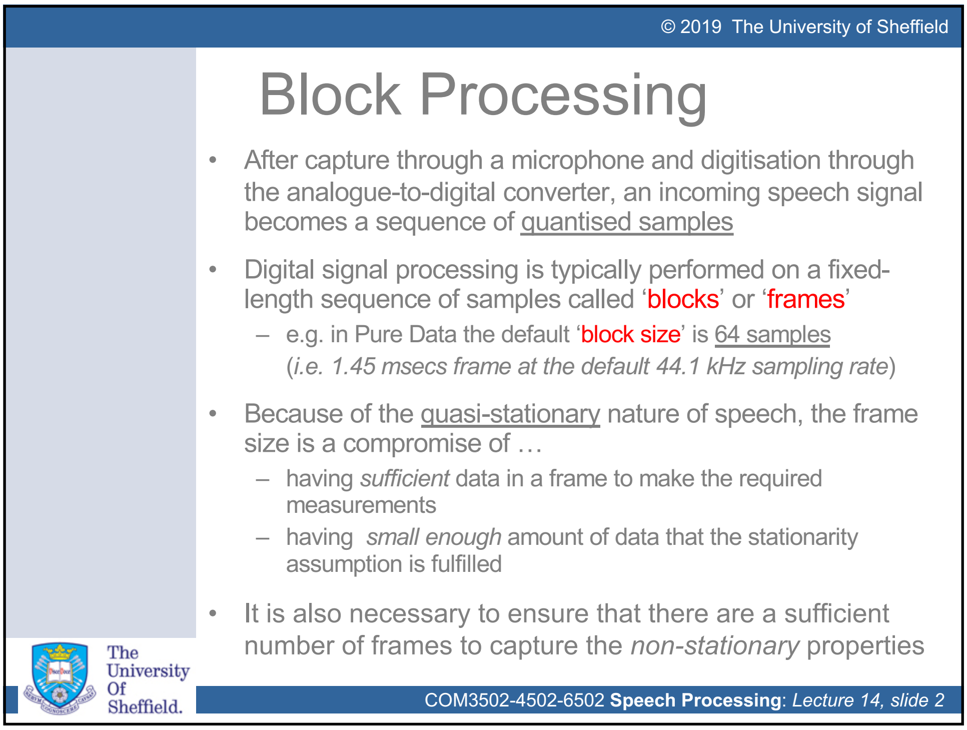
COM3502/4502/6502 SPEECH PROCESSING

Lecture 14

Waveform Processing

The University Of Sheffield.

1



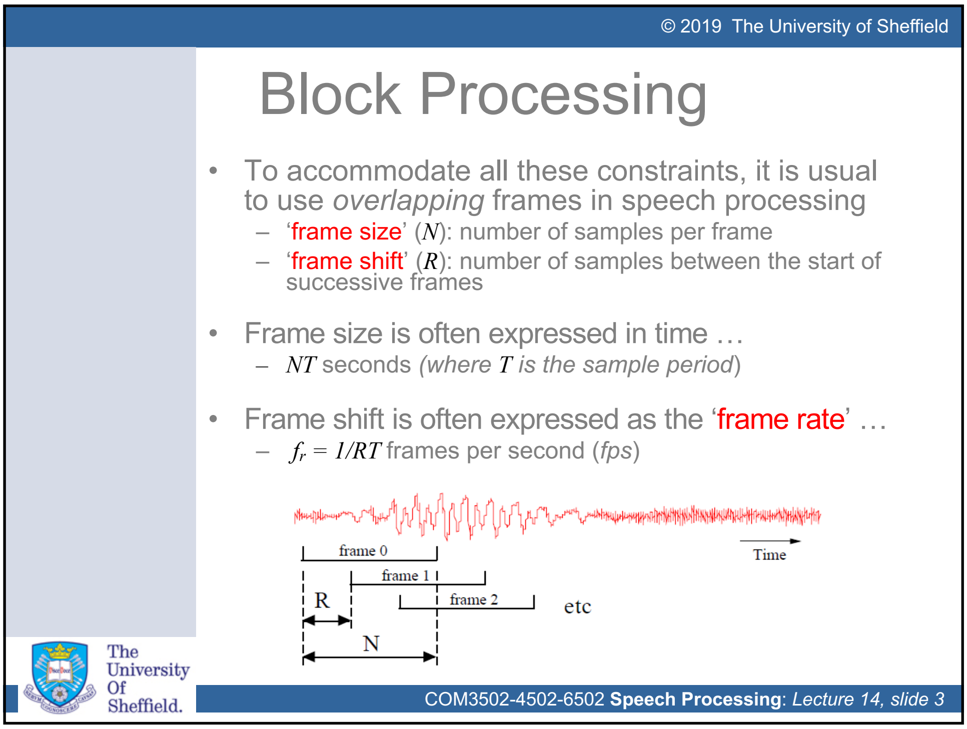
© 2019 The University of Sheffield

Block Processing

- After capture through a microphone and digitisation through the analogue-to-digital converter, an incoming speech signal becomes a sequence of quantised samples
- Digital signal processing is typically performed on a fixed-length sequence of samples called 'blocks' or 'frames'
  - e.g. in Pure Data the default 'block size' is 64 samples (*i.e. 1.45 msecs frame at the default 44.1 kHz sampling rate*)
- Because of the quasi-stationary nature of speech, the frame size is a compromise of …
  - having *sufficient* data in a frame to make the required measurements
  - having *small enough* amount of data that the stationarity assumption is fulfilled
- It is also necessary to ensure that there are a sufficient number of frames to capture the *non-stationary* properties

The University Of Sheffield.

2

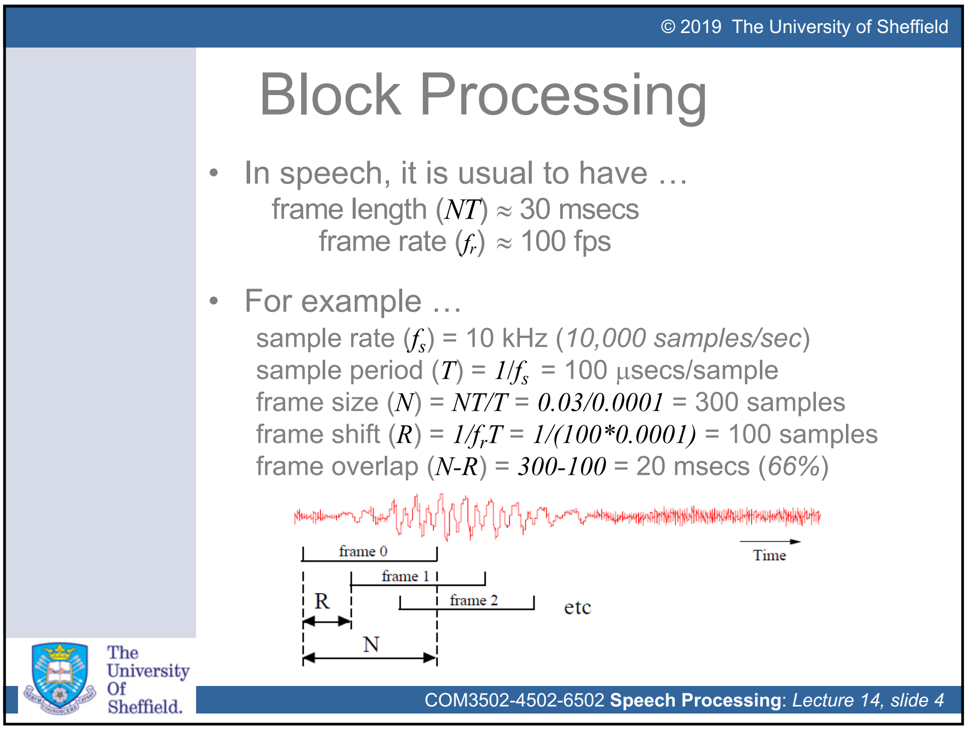


© 2019 The University of Sheffield

Block Processing

- To accommodate all these constraints, it is usual to use *overlapping* frames in speech processing
  - '<span>frame size</span>' (N): number of samples per frame
  - 'frame shift' (R): number of samples between the start of successive frames
- Frame size is often expressed in time …
  - NT seconds *(where T is the sample period*)
- Frame shift is often expressed as the 'frame rate' …
  - $f_r = 1/RT$ frames per second (*fps*)

3



© 2019 The University of Sheffield

Block Processing

- In speech, it is usual to have …

  frame length (NT) ≈ 30 msecs

  frame rate (f_r) ≈ 100 fps
- For example …

  sample rate (f_s) = 10 kHz (*10,000 samples/sec*)

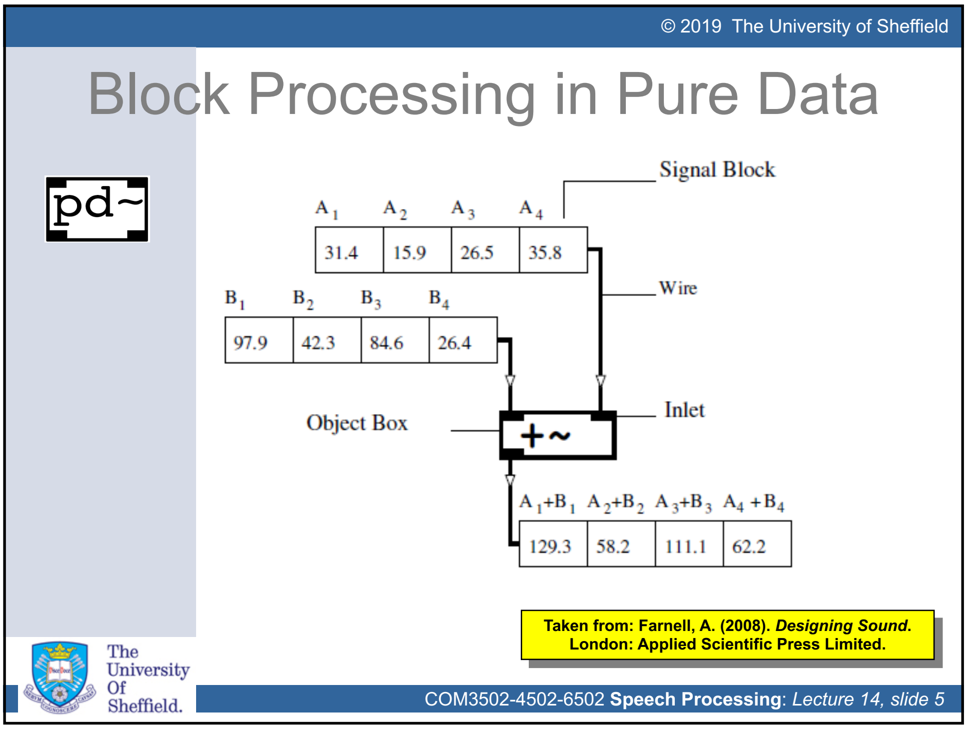
  sample period (T) = $1/f_s$ = 100 μsecs/sample

  frame size (N) = NT/T = $0.03/0.0001$ = 300 samples

  frame shift (R) = $1/f_rT$ = $1/(100*0.0001)$ = 100 samples

  frame overlap (N-R) = 300-100 = 20 msecs (*66%*)

4



© 2019 The University of Sheffield

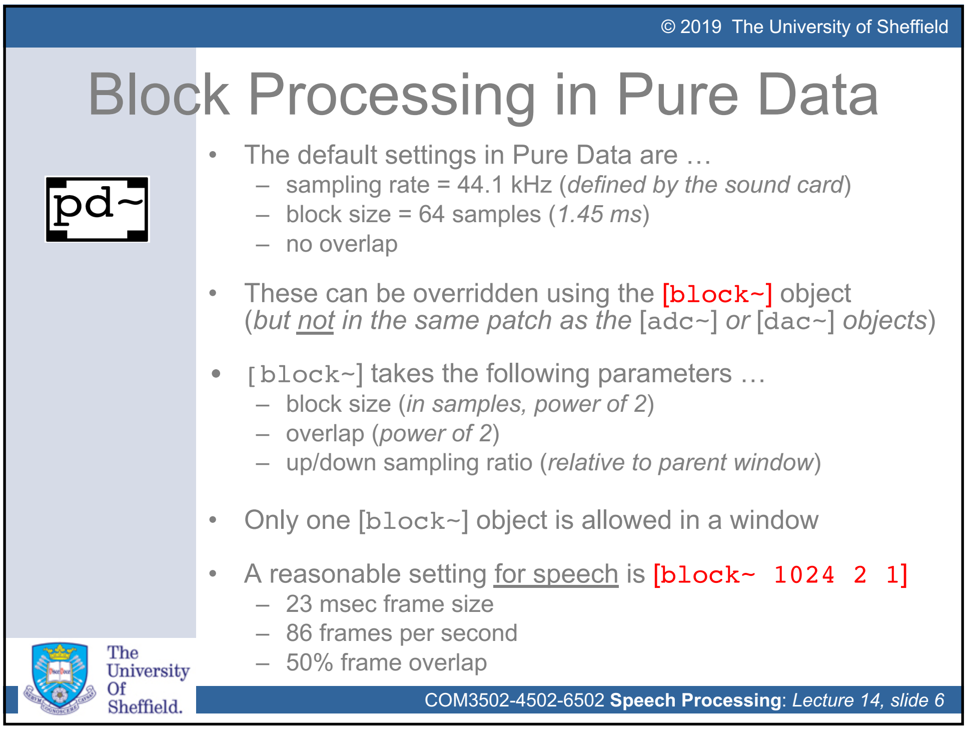
Block Processing in Pure Data

A_1	A_2	A_3	A_4
31.4	15.9	26.5	35.8

B_1	B_2	B_3	B_4
97.9	42.3	84.6	26.4

A_1+B_1	A_2+B_2	A_3+B_3	A_4+B_4
129.3	58.2	111.1	62.2

Taken from: Farnell, A. (2008). ***Designing Sound.***
London: Applied Scientific Press Limited.



© 2019 The University of Sheffield

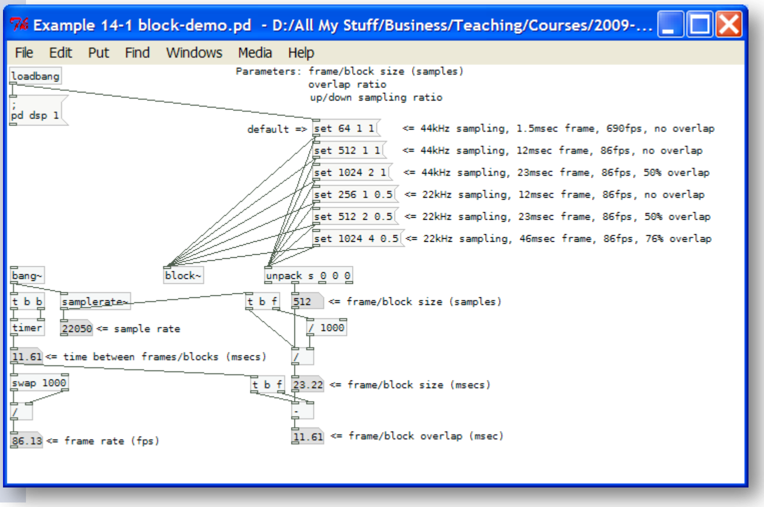
Block Processing in Pure Data

- The default settings in Pure Data are …
  - sampling rate = 44.1 kHz (*defined by the sound card*)
  - block size = 64 samples (*1.45 ms*)
  - no overlap
- These can be overridden using the `[block~]` object (*but not in the same patch as the* `[adc~]` *or* `[dac~]` *objects*)
- `[block~]` takes the following parameters …
  - block size (*in samples, power of 2*)
  - overlap (*power of 2*)
  - up/down sampling ratio (*relative to parent window*)
- Only one `[block~]` object is allowed in a window
- A reasonable setting for speech is `[block~ 1024 2 1]`
  - 23 msec frame size
  - 86 frames per second
  - 50% frame overlap

© 2019 The University of Sheffield

Block Processing in Pure Data

pd~



Example 14-1 block-demo.pd - D:/All My Stuff/Business/Teaching/Courses/2009-...

```
Parameters: frame/block size (samples)
            overlap ratio
            up/down sampling ratio

default => set 64 1 1       <= 44kHz sampling, 1.5msec frame, 690fps, no overlap
           set 512 1 1      <= 44kHz sampling, 12msec frame, 86fps, no overlap
           set 1024 2 1     <= 44kHz sampling, 23msec frame, 86fps, 50% overlap
           set 256 1 0.5    <= 22kHz sampling, 12msec frame, 86fps, no overlap
           set 512 2 0.5    <= 22kHz sampling, 23msec frame, 86fps, 50% overlap
           set 1024 4 0.5   <= 22kHz sampling, 46msec frame, 86fps, 76% overlap

512 <= frame/block size (samples)
22050 <= sample rate
11.61 <= time between frames/blocks (msecs)
23.22 <= frame/block size (msecs)
86.13 <= frame rate (fps)
11.61 <= frame/block overlap (msec)
```

The University Of Sheffield.

© 2019 The University of Sheffield

Sample-by-Sample Processing in Pd

pd~

- Sometimes it is necessary to process data one sample at a time
- Pd provides an object for this: [fexpr~]
- [fexpr~] takes the following arguments …
  - $i#: integer input variable on inlet #
  - $f#: float input variable on inlet #
- Expressions in [fexpr~] are constructed using …
  - $x#[n]: the sample from inlet # indexed by n
  - $y[n]: the output value indexed by n
    (*$x# is shorthand for the current input*)
    (*$y is shorthand for the previous output: $y[-1]*)
- E.g. [fexpr~ $x1+$y] is a simple accumulator

The University Of Sheffield.

© 2019 The University of Sheffield

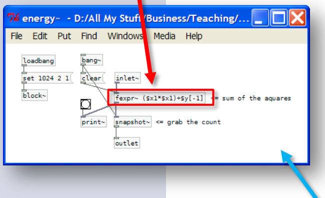
Short-Time Energy

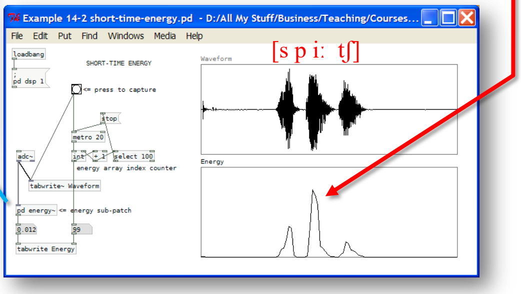
'Short-time energy' = sum of the squares of the samples in one frame

$$E = \sum_{i=0}^{N-1} s_i^2$$

Sample by sample computation

Energy is large in voiced speech

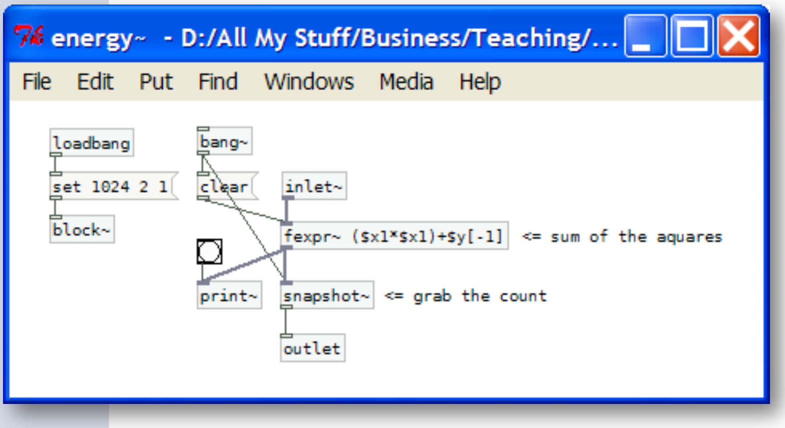




9

© 2019 The University of Sheffield

Short-Time Energy


energy~ - D:/All My Stuff/Business/Teaching/...
File Edit Put Find Windows Media Help

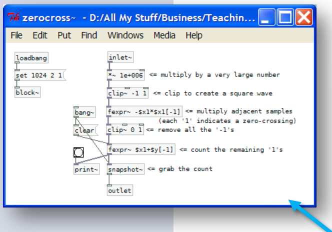
```
loadbang          bang~
set 1024 2 1      clear     inlet~
block~
                            fexpr~ ($x1*$x1)+$y[-1]   <= sum of the aquares

                  print~    snapshot~   <= grab the count

                            outlet
```

10

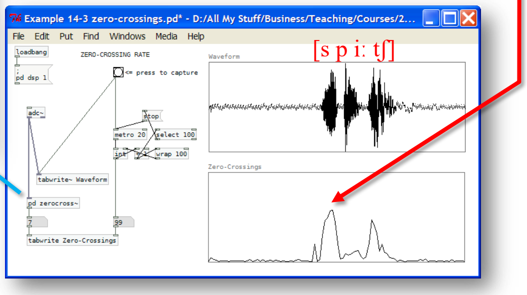
© 2019 The University of Sheffield

Zero-Crossing Rate

'ZCR' = number of times the zero axis is crossed in <u>one</u> frame


zerocross~ - D:/All My Stuff/Business/Teachin...
File Edit Put Find Windows Media Help

```
loadbang          inlet~
set 1024 2 1      *~ 1e+006  <= multiply by a very large number
block~            clip~ -1 1  <= clip to create a square wave
        bang~     fexpr~ -$x1*$x1[-1]  <= multiply adjacent samples
                                (each '1' indicates a zero-crossing)
        clear     clip~ 0 1  <= remove all the '-1's
                  fexpr~ $x1+$y[-1]  <= count the remaining '1's
        print~    snapshot~  <= grab the count
                  outlet
```

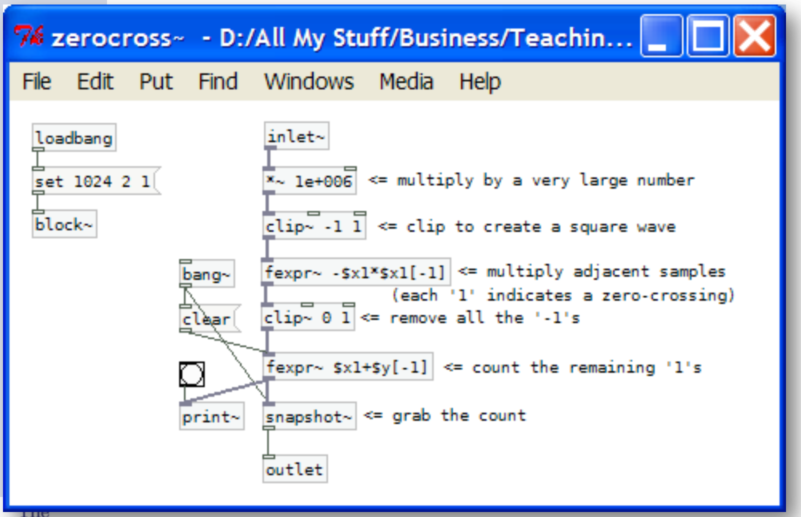

Example 14-3 zero-crossings.pd* - D:/All My Stuff/Business/Teaching/Courses/2...
File Edit Put Find Windows Media Help
Waveform
[s p i: tʃ]
Zero-Crossings

```
loadbang          ZERO-CROSSING RATE
;
pd dsp 1                         <= press to capture
adc~                             stop
                          metro 20  select 100
                          int  + 1  wrap 100
tabwrite~ Waveform
pd zerocross~
7                         99
tabwrite Zero-Crossings
```

ZCR is large in <u>unvoiced</u> speech

© 2019 The University of Sheffield

Zero-Crossing Rate


zerocross~ - D:/All My Stuff/Business/Teachin...
File Edit Put Find Windows Media Help

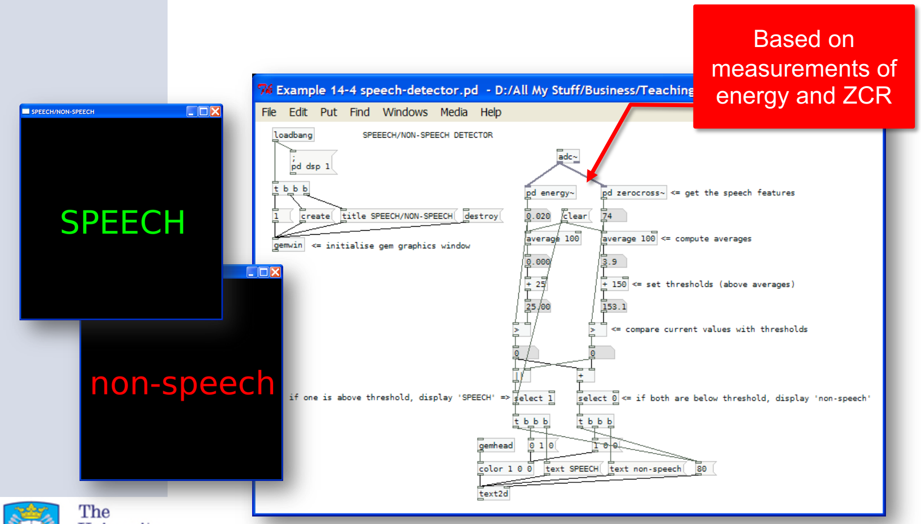
```
loadbang          inlet~
set 1024 2 1      *~ 1e+006  <= multiply by a very large number
block~            clip~ -1 1  <= clip to create a square wave
        bang~     fexpr~ -$x1*$x1[-1]  <= multiply adjacent samples
                                (each '1' indicates a zero-crossing)
        clear     clip~ 0 1  <= remove all the '-1's
                  fexpr~ $x1+$y[-1]  <= count the remaining '1's
        print~    snapshot~  <= grab the count
                  outlet
```

Speech/Non-Speech Detection

- In speech processing it is often useful to be able to detect when someone is speaking
- Accurate speech ‘end-point detection’ is very difficult
- A simple ‘speech/non-speech detector’ can be constructed using short-time energy *and* zero-crossing rate
  - energy is high in voiced speech
  - ZCR is high in unvoiced speech

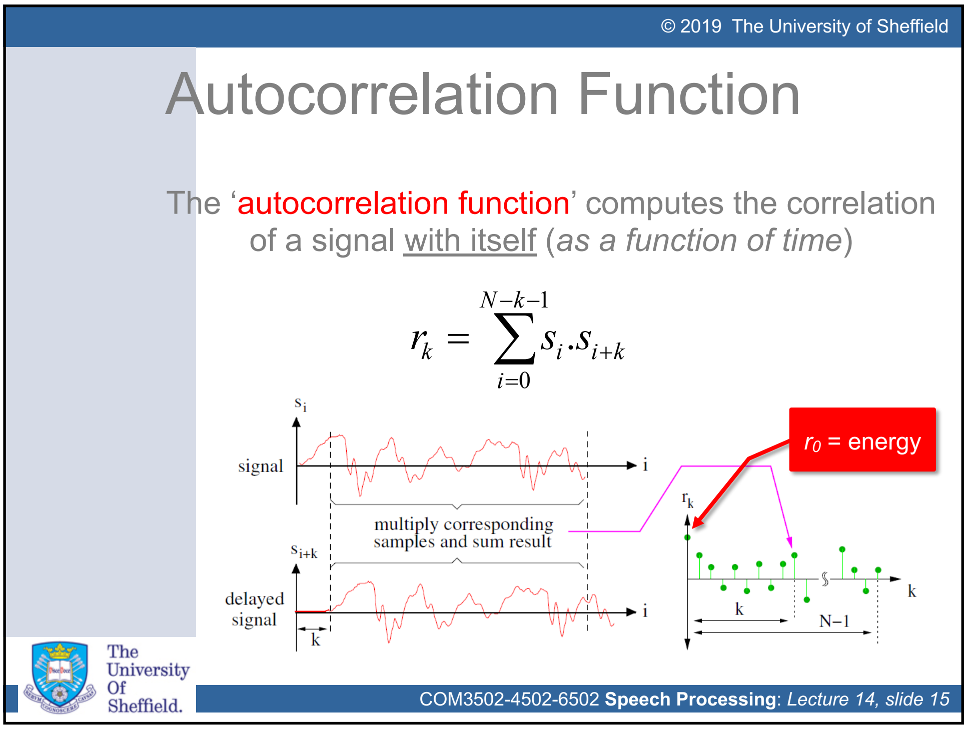
13

Speech/Non-Speech Detection



Based on measurements of energy and ZCR

14



© 2019 The University of Sheffield

Autocorrelation Function

The ‘autocorrelation function’ computes the correlation of a signal with itself (*as a function of time*)

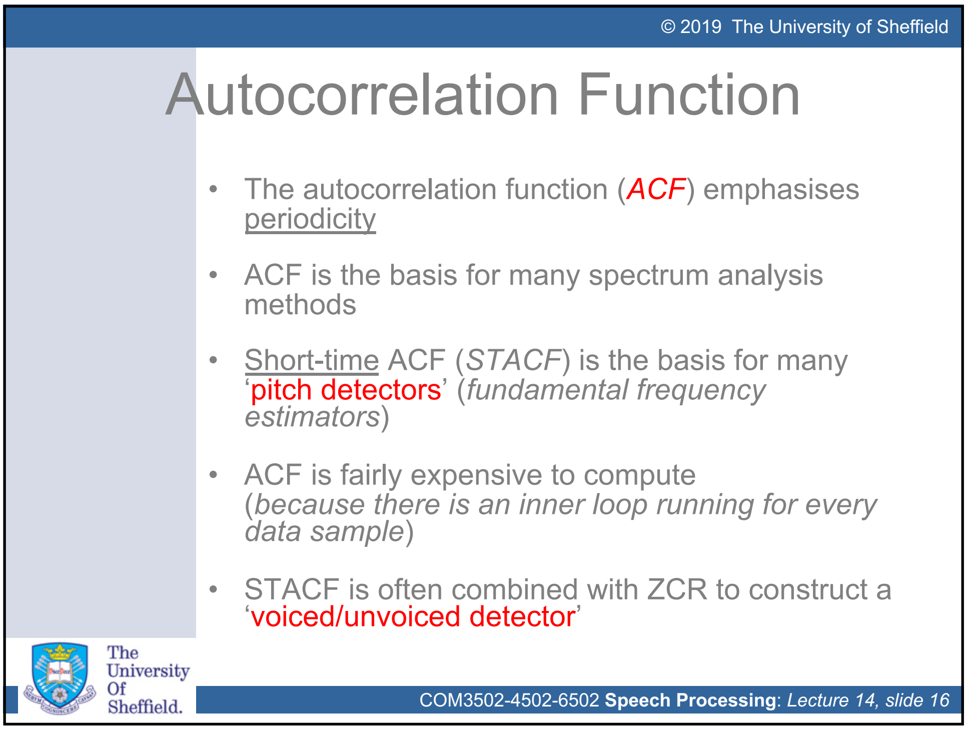
$$r_k = \sum_{i=0}^{N-k-1} s_i . s_{i+k}$$

r_0 = energy

The University Of Sheffield.

COM3502-4502-6502 **Speech Processing**: *Lecture 14, slide 15*

15



© 2019 The University of Sheffield

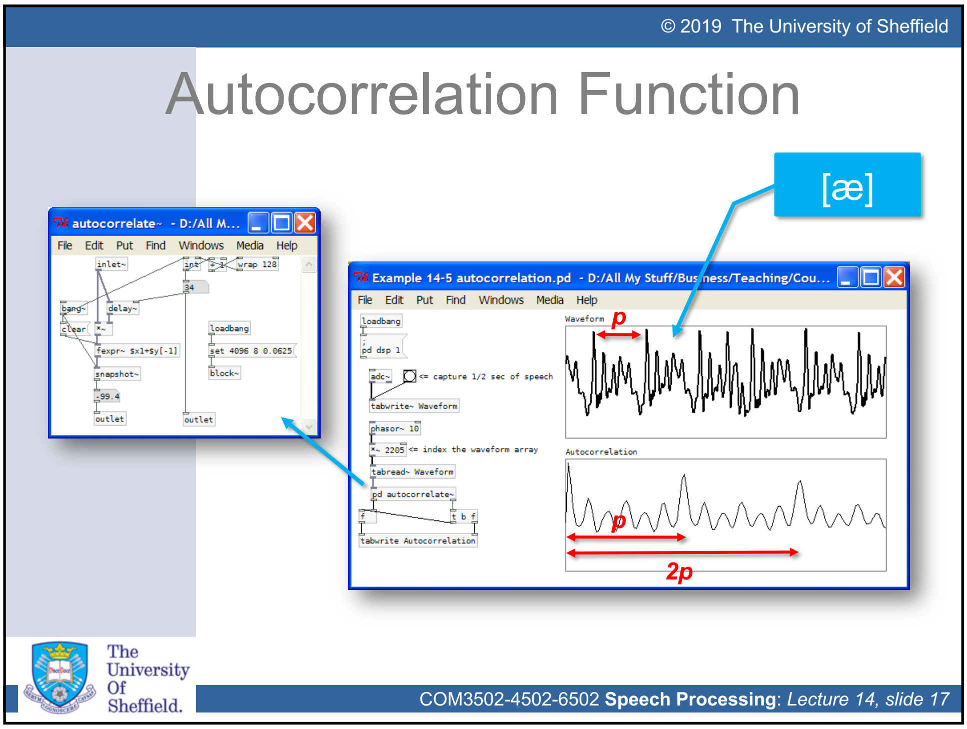
Autocorrelation Function

- The autocorrelation function (*ACF*) emphasises periodicity
- ACF is the basis for many spectrum analysis methods
- Short-time ACF (*STACF*) is the basis for many ‘pitch detectors’ (*fundamental frequency estimators*)
- ACF is fairly expensive to compute (*because there is an inner loop running for every data sample*)
- STACF is often combined with ZCR to construct a ‘voiced/unvoiced detector’

The University Of Sheffield.

COM3502-4502-6502 **Speech Processing**: *Lecture 14, slide 16*

16



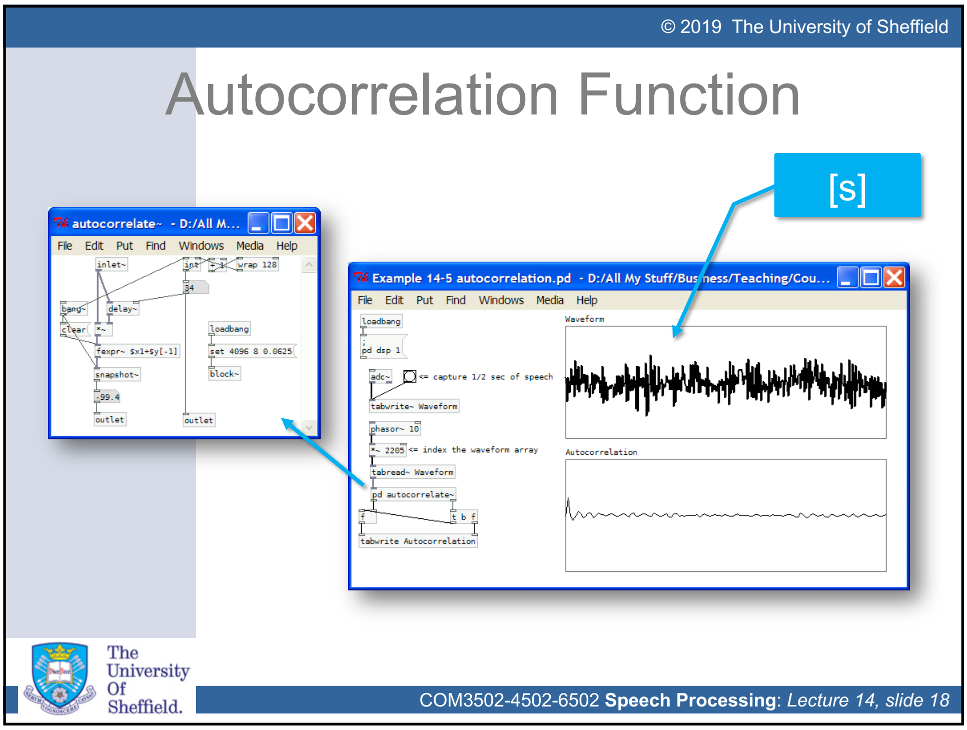
© 2019 The University of Sheffield

Autocorrelation Function

[æ]

The University Of Sheffield.

17



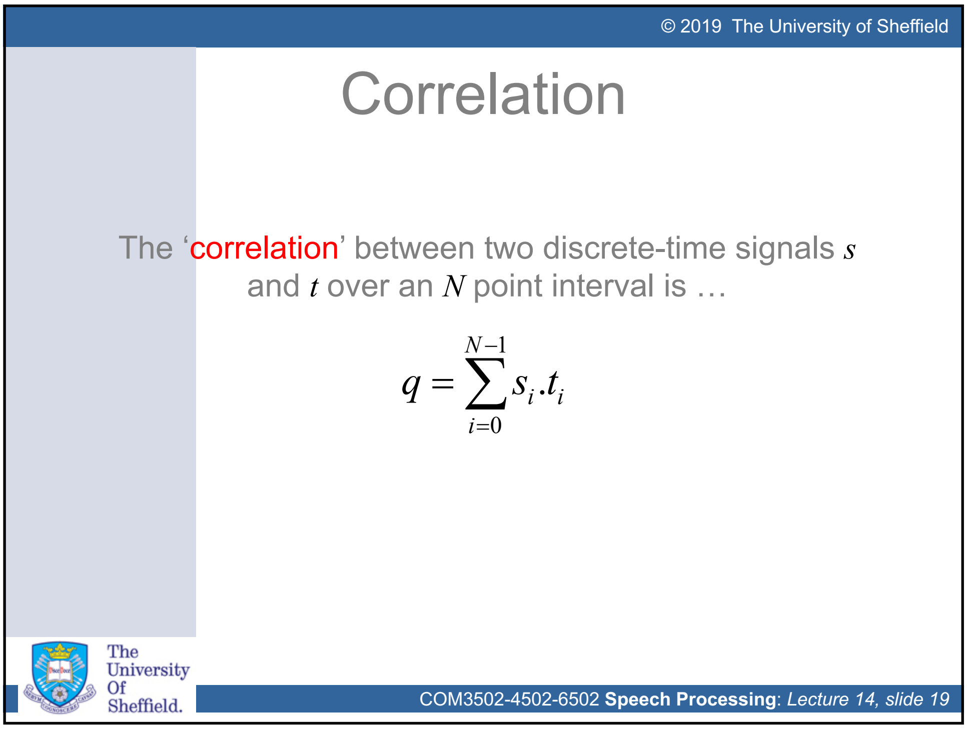
© 2019 The University of Sheffield

Autocorrelation Function

[s]

The University Of Sheffield.

18



© 2019 The University of Sheffield

Correlation

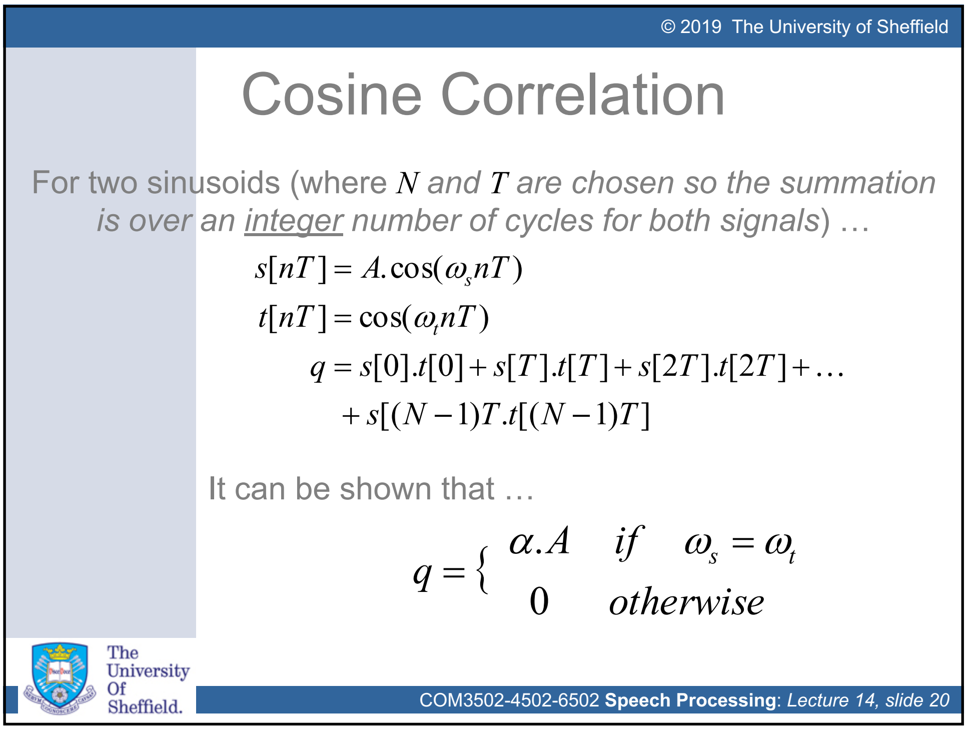
The 'correlation' between two discrete-time signals s and t over an N point interval is …

$$q = \sum_{i=0}^{N-1} s_i . t_i$$

The University Of Sheffield.

COM3502-4502-6502 **Speech Processing**: *Lecture 14, slide 19*

19



© 2019 The University of Sheffield

Cosine Correlation

For two sinusoids (where N *and* T *are chosen so the summation is over an* *integer* *number of cycles for both signals*) …

$$s[nT] = A.\cos(\omega_s nT)$$

$$t[nT] = \cos(\omega_t nT)$$

$$q = s[0].t[0] + s[T].t[T] + s[2T].t[2T] + \ldots$$
$$+ s[(N-1)T.t[(N-1)T]$$

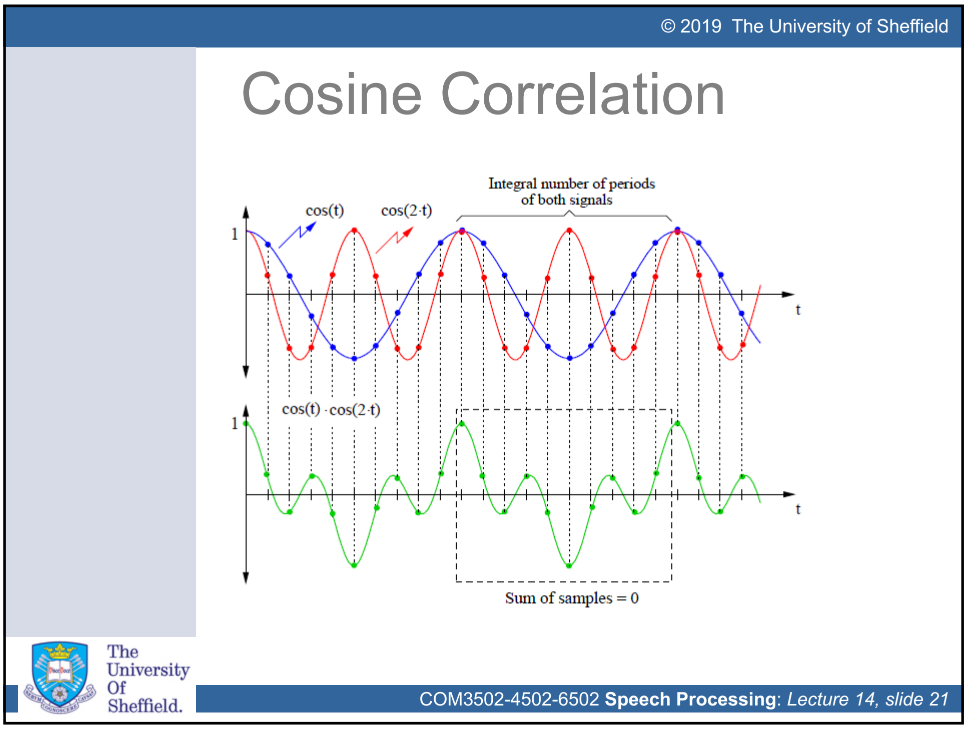
It can be shown that …

$$q = \{ \begin{matrix} \alpha.A & if \quad \omega_s = \omega_t \\ 0 & otherwise \end{matrix}$$

The University Of Sheffield.

COM3502-4502-6502 **Speech Processing**: *Lecture 14, slide 20*

20

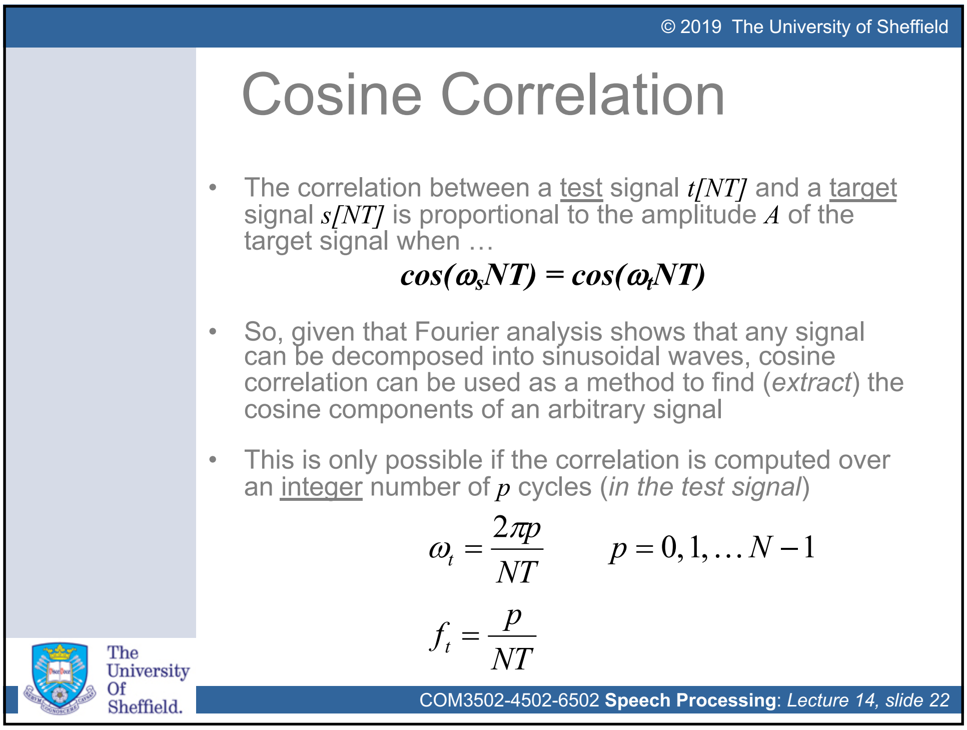


© 2019 The University of Sheffield

Cosine Correlation

The University Of Sheffield.

21



© 2019 The University of Sheffield

Cosine Correlation

- The correlation between a test signal $t[NT]$ and a target signal $s[NT]$ is proportional to the amplitude A of the target signal when …

$$\boldsymbol{cos(\omega_s NT) = cos(\omega_t NT)}$$

- So, given that Fourier analysis shows that any signal can be decomposed into sinusoidal waves, cosine correlation can be used as a method to find (*extract*) the cosine components of an arbitrary signal

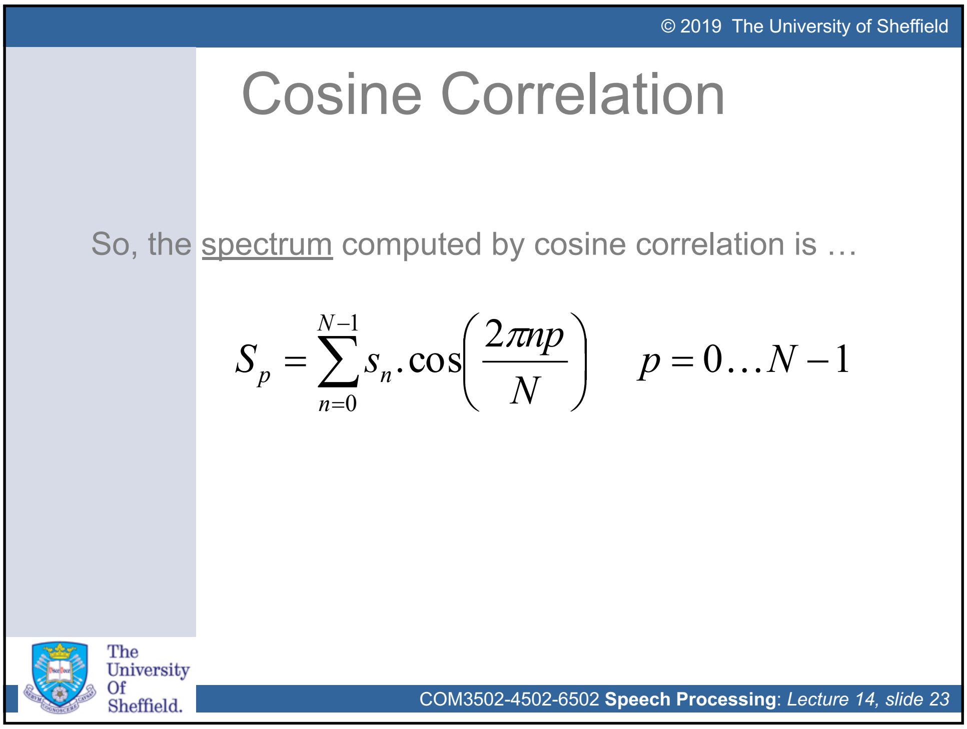
- This is only possible if the correlation is computed over an integer number of p cycles (*in the test signal*)

$$\omega_t = \frac{2\pi p}{NT} \qquad p = 0, 1, \ldots N-1$$

$$f_t = \frac{p}{NT}$$

The University Of Sheffield.

22



© 2019 The University of Sheffield

Cosine Correlation

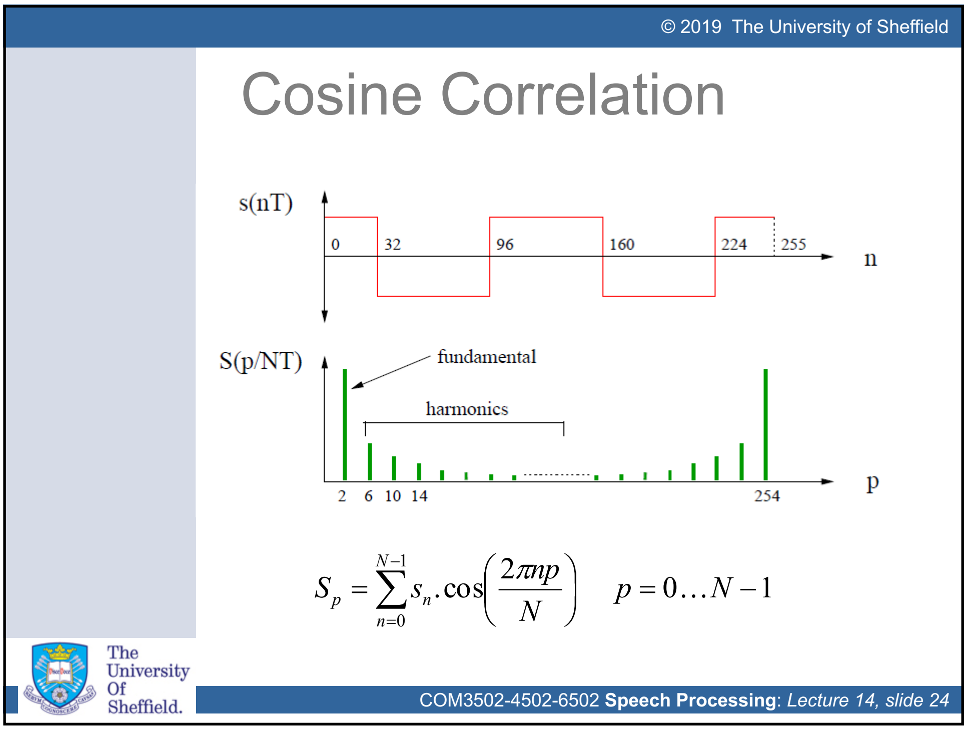
So, the spectrum computed by cosine correlation is …

$$S_p = \sum_{n=0}^{N-1} s_n . \cos\left(\frac{2\pi np}{N}\right) \qquad p = 0 \ldots N-1$$

The University Of Sheffield.

COM3502-4502-6502 **Speech Processing**: *Lecture 14, slide 23*

23



© 2019 The University of Sheffield

Cosine Correlation

$$S_p = \sum_{n=0}^{N-1} s_n . \cos\left(\frac{2\pi np}{N}\right) \qquad p = 0 \ldots N-1$$

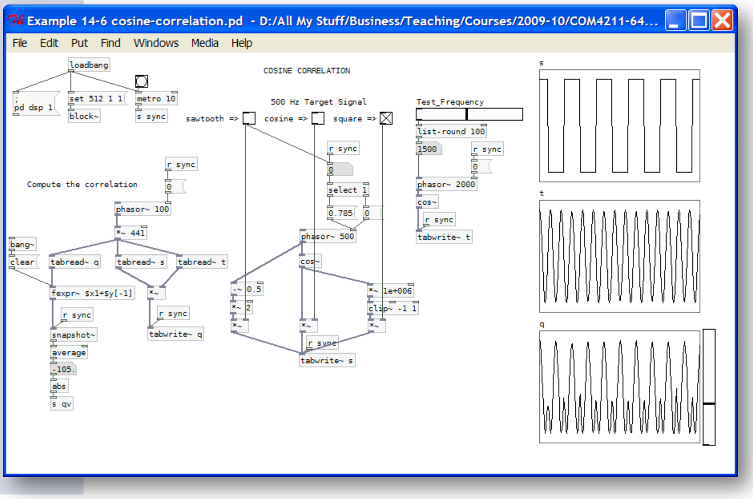
The University Of Sheffield.

COM3502-4502-6502 **Speech Processing**: *Lecture 14, slide 24*

24

© 2019 The University of Sheffield

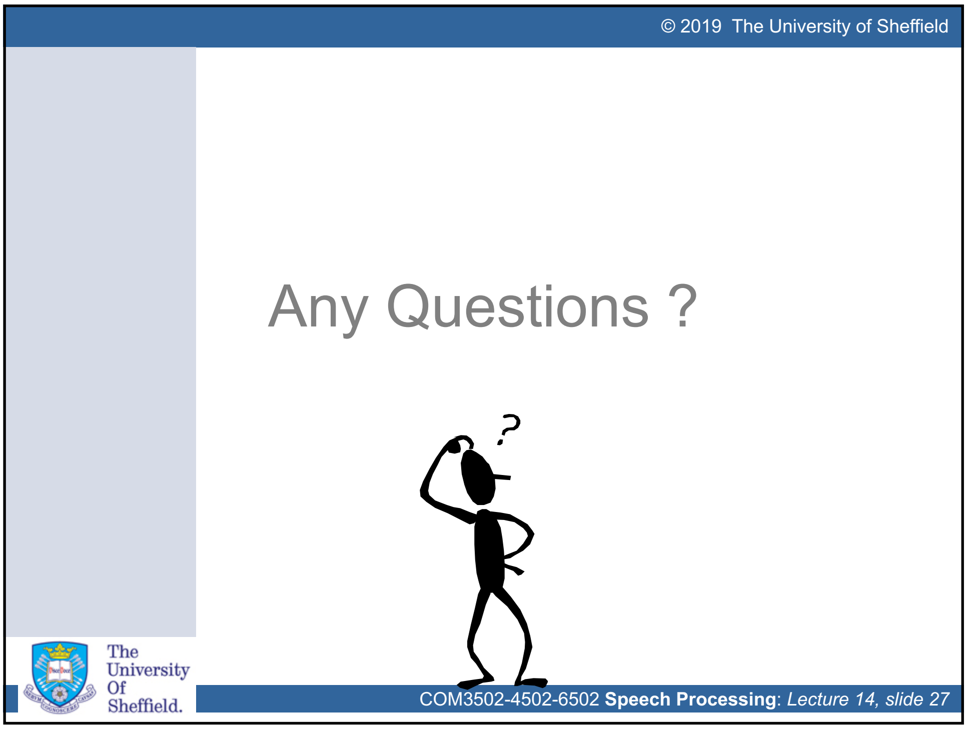
Cosine Correlation



© 2019 The University of Sheffield

This lecture has covered …

- Processing blocks/frames
- Processing samples
- Short-time energy
- Zero crossing rate
- Speech/non-speech detection
- Autocorrelation
- Cosine correlation



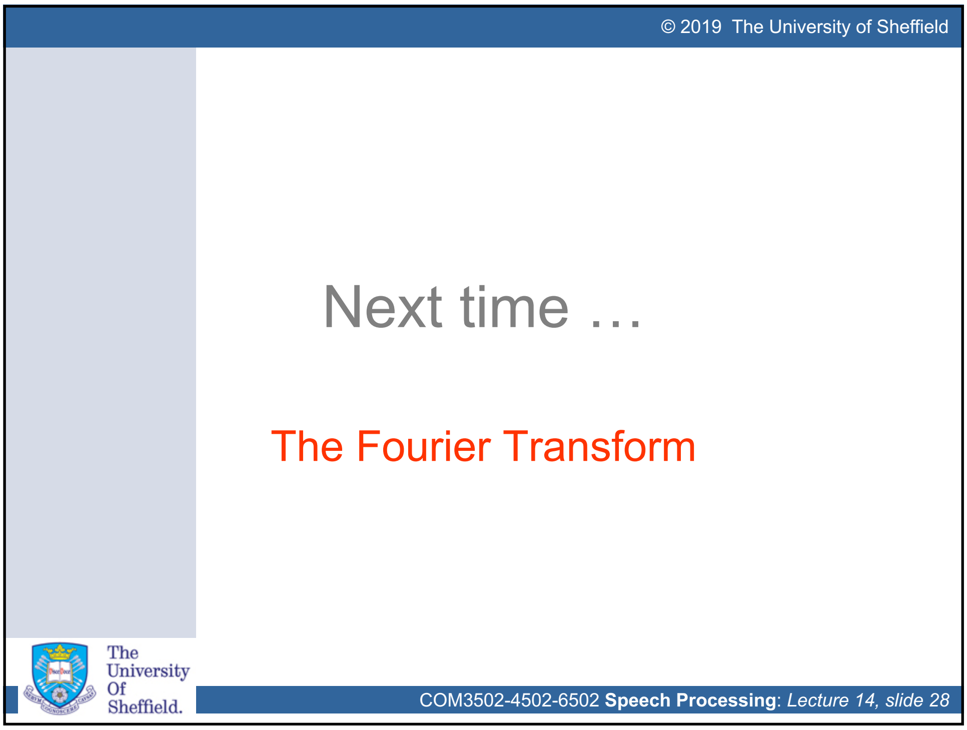
© 2019 The University of Sheffield

Any Questions ?

The University Of Sheffield.

COM3502-4502-6502 **Speech Processing**: *Lecture 14, slide 27*

27



© 2019 The University of Sheffield

Next time …

The Fourier Transform

The University Of Sheffield.

COM3502-4502-6502 **Speech Processing**: *Lecture 14, slide 28*

28