COMP28512: Mobile Systems
Lab_Support Lecture 1

Barry Cheetham

barry@man.ac.uk

www.cs.man.ac.uk/~barry/mydocs/COMP28512

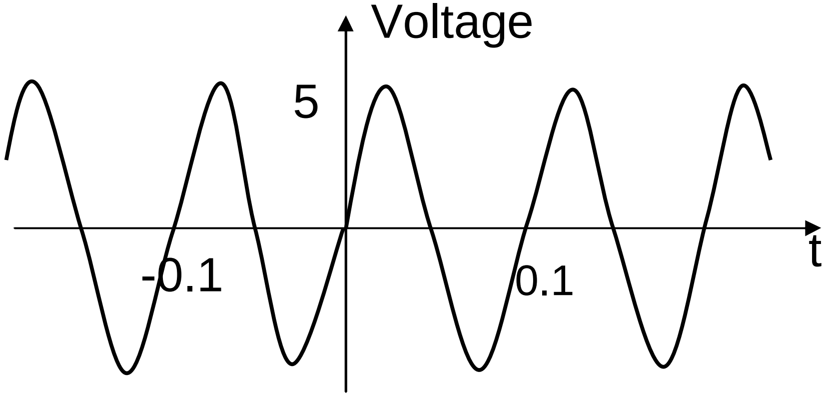
Analogue signals

- Continuous variations of voltage.
- Exist for all values of t in range -∞ to +∞.
- Example:

  $v(t) = 5\times\sin(2\pi\times 10 \times t)$:

  sine-wave of amplitude 5

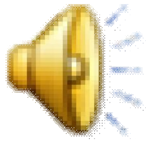
  and frequency 10 cycles per second (Hz)
- Graph of voltage against time gives continuous 'waveform':

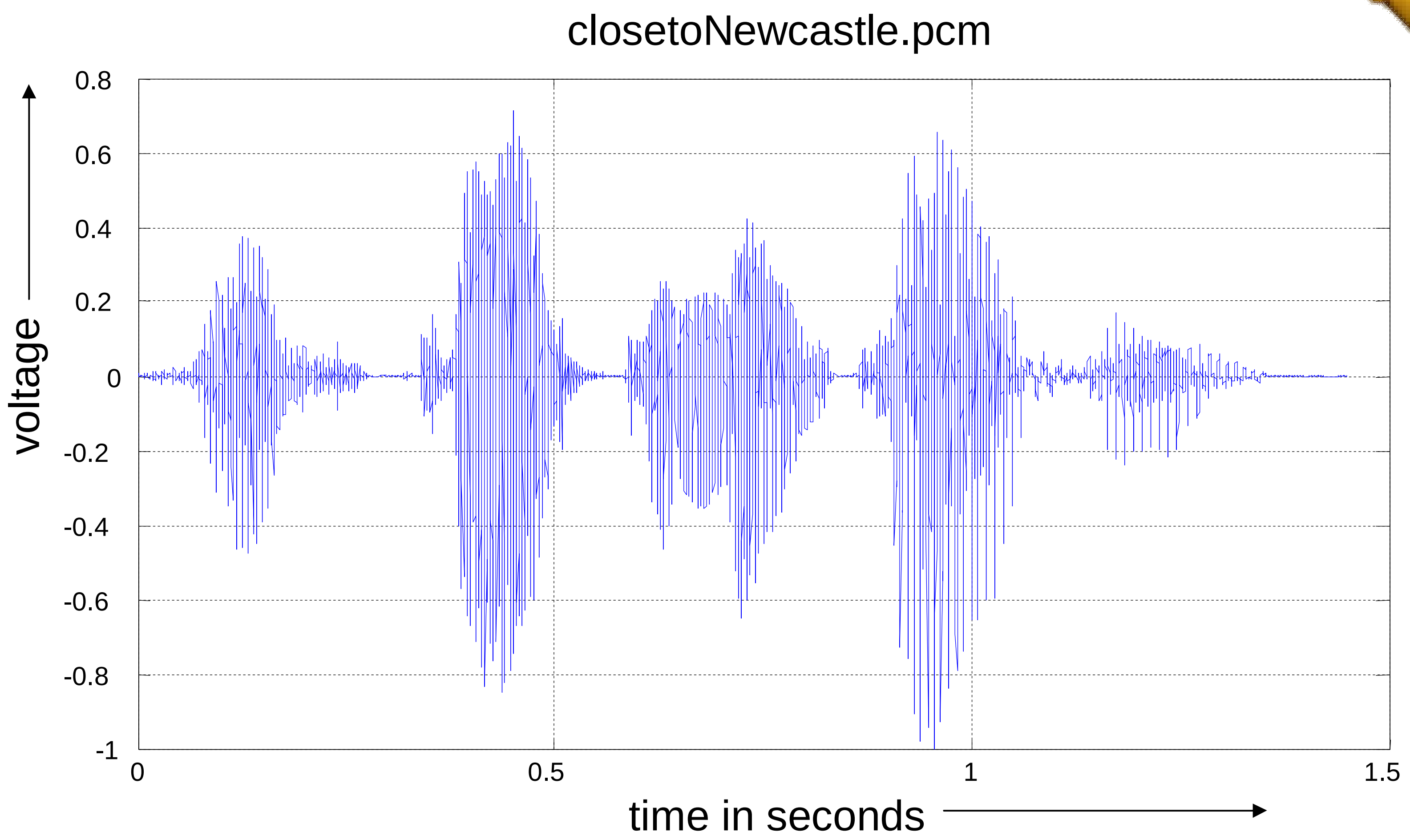


‘Sound’

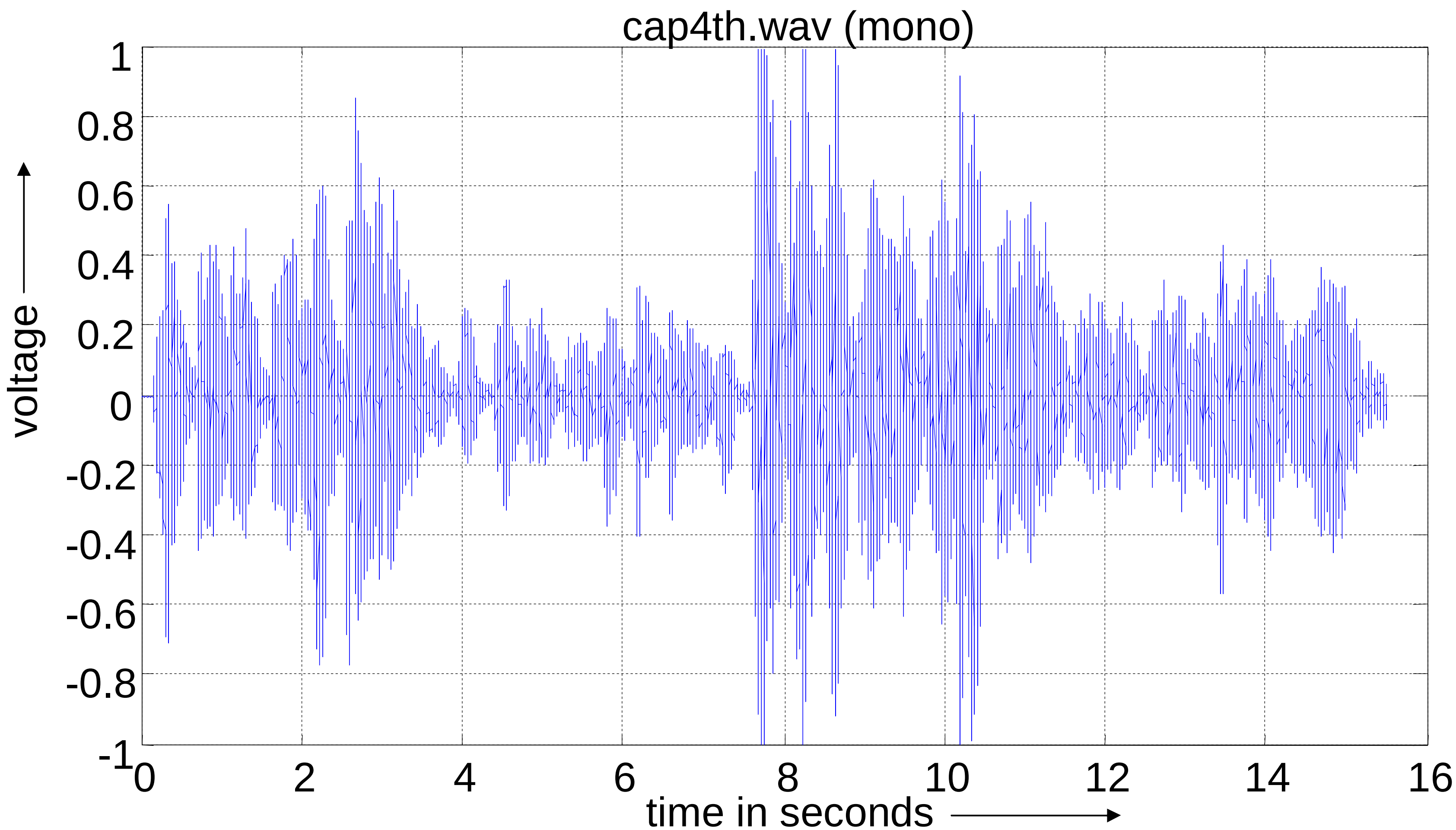
- Variation in air pressure
- May be generated by musical instruments or human voice.
- E.g. by a piano string or your vocal cords vibrating
- Local pressure variation travels thro’ the air to your ear or a microphone.
- There it causes a ‘diaphragm’ to vibrate.
- Microphone produces a continuous voltage proportional to the variation in air pressure.
- Graph of voltage against time is ‘sound waveform’.

Segment of telephone speech: 'Its close to Newcastle'

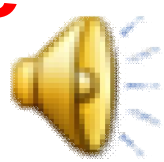


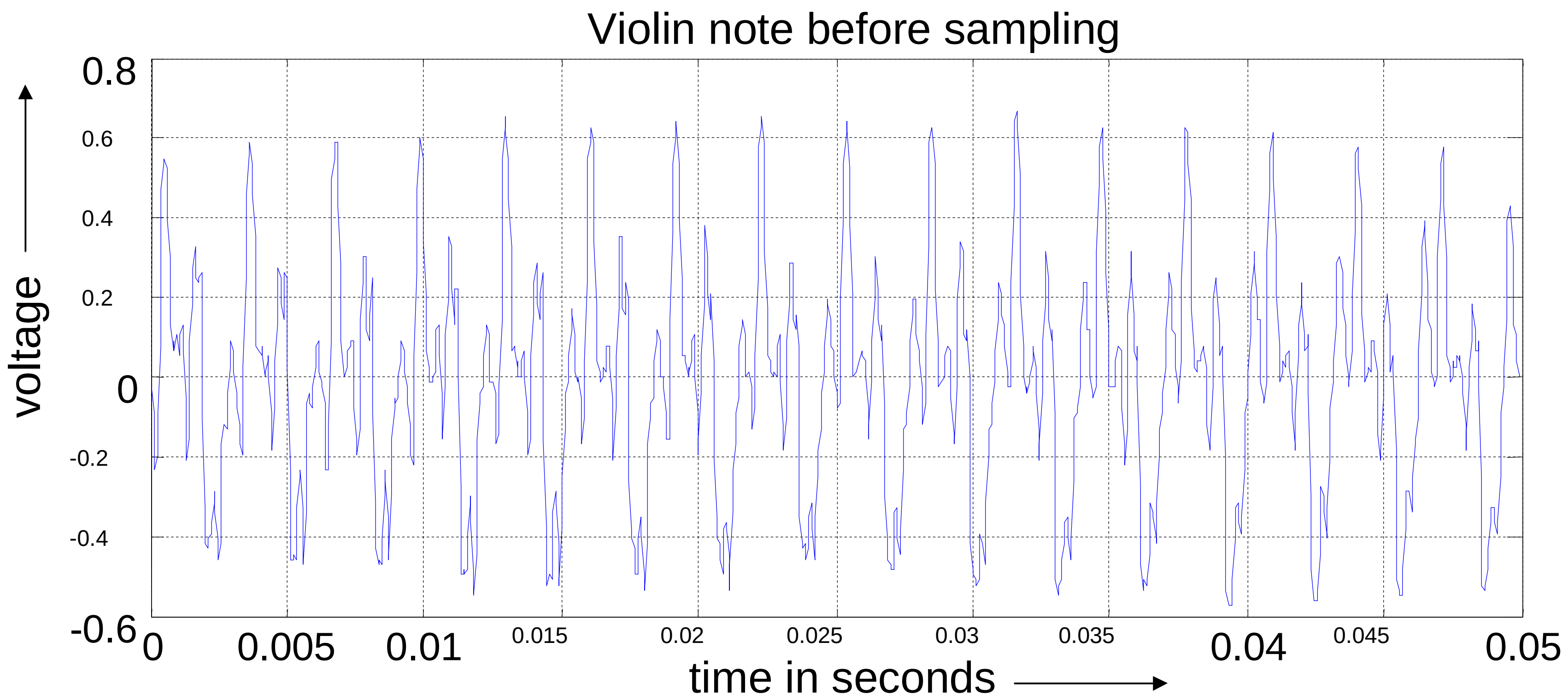


16 s segment of violin music



50ms segment of music: violin note





What is frequency of note being played? Ans ≈16 cycles in 0.05 s ≈ 320 Hz

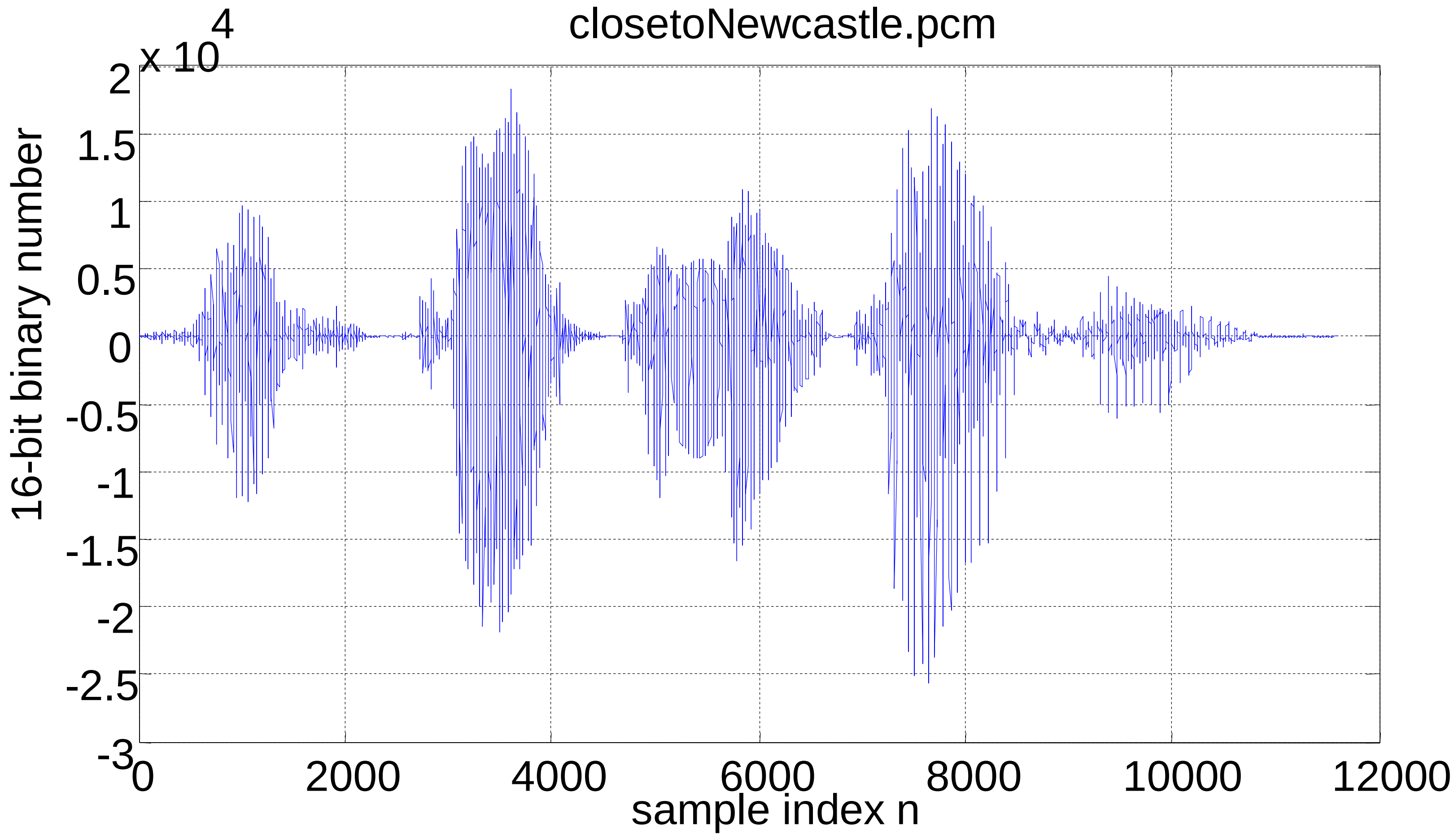
Discrete-time signal

- Exists only at discrete points in time.
- Often obtained by **sampling** an analogue signal,

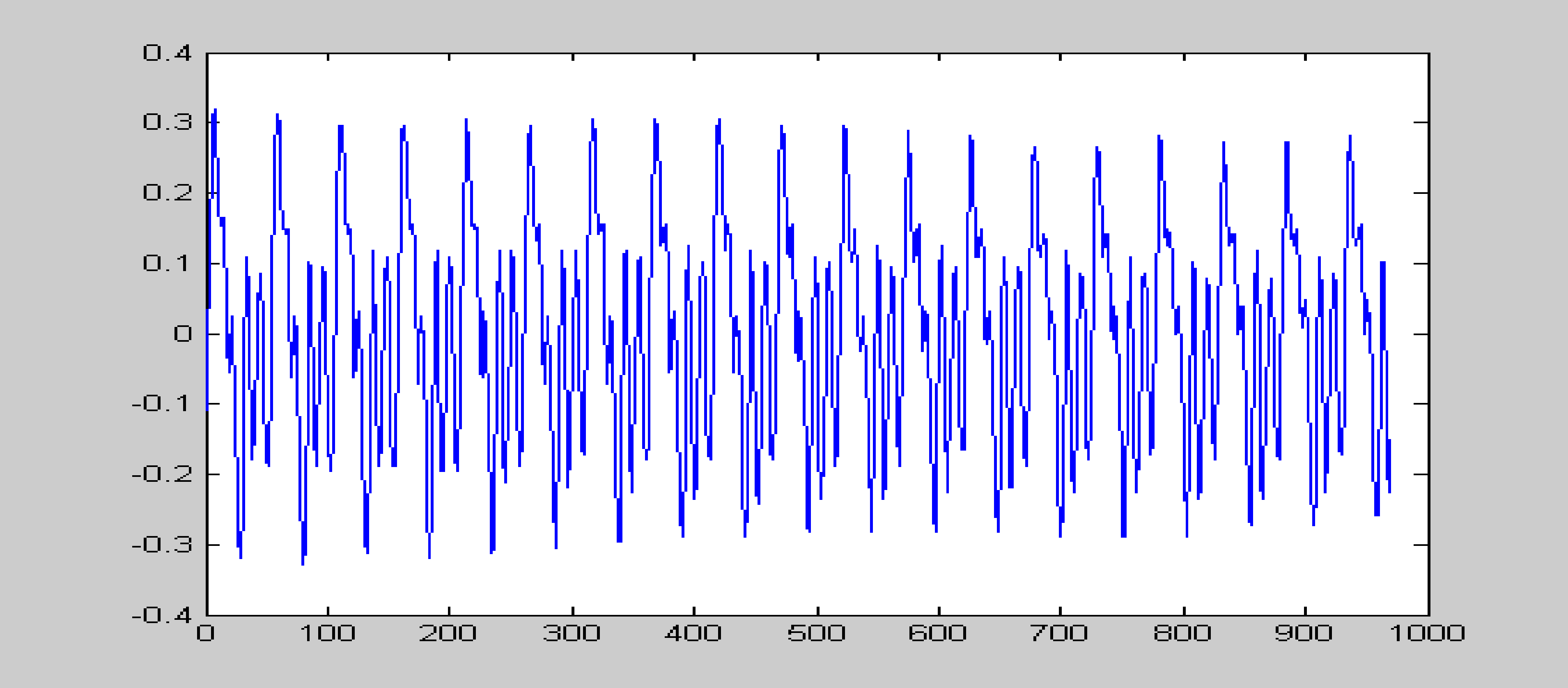
  i.e. measuring its value at discrete points in time.
- Sampling points separated by equal intervals of T seconds.
- Given analogue signal **x(t)**, **x[n]** = value of x(t) when t = nT.
- Sampling process produces a sequence of numbers:

  { ..., x[-2], x[-1], x[0], x[1], x[2], ... }
- Referred to as {x[n]} or ' the sequence x[n] '.
- Sequence exists for all integer n in the range $-\infty$ to ∞.

Speech sampled at 8kHz with 16 bits per sample



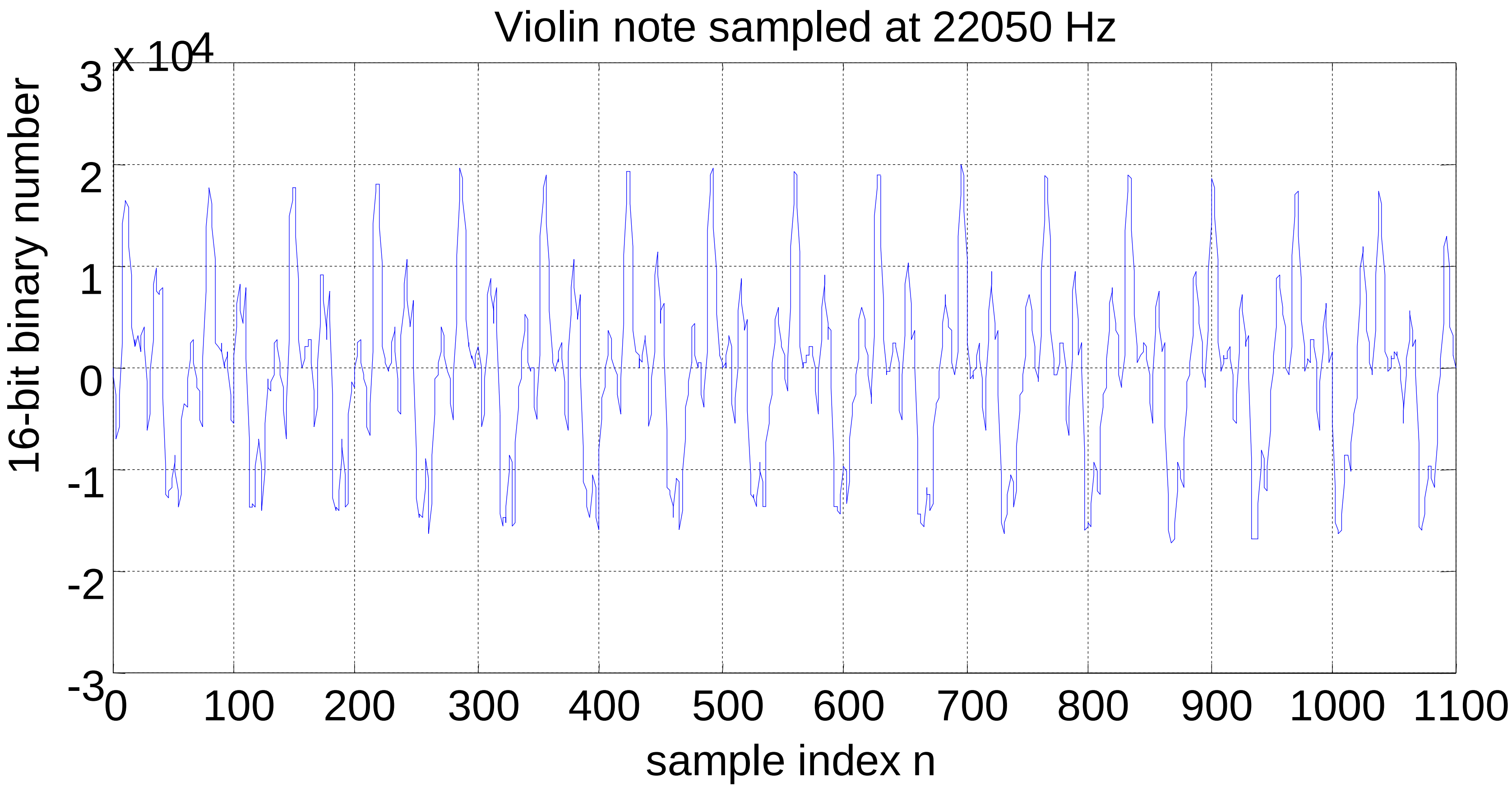
45 ms segment of music sampled at 22.05 kHz



Frequency of note ≈19/1000 = 0.019 cycles/sample

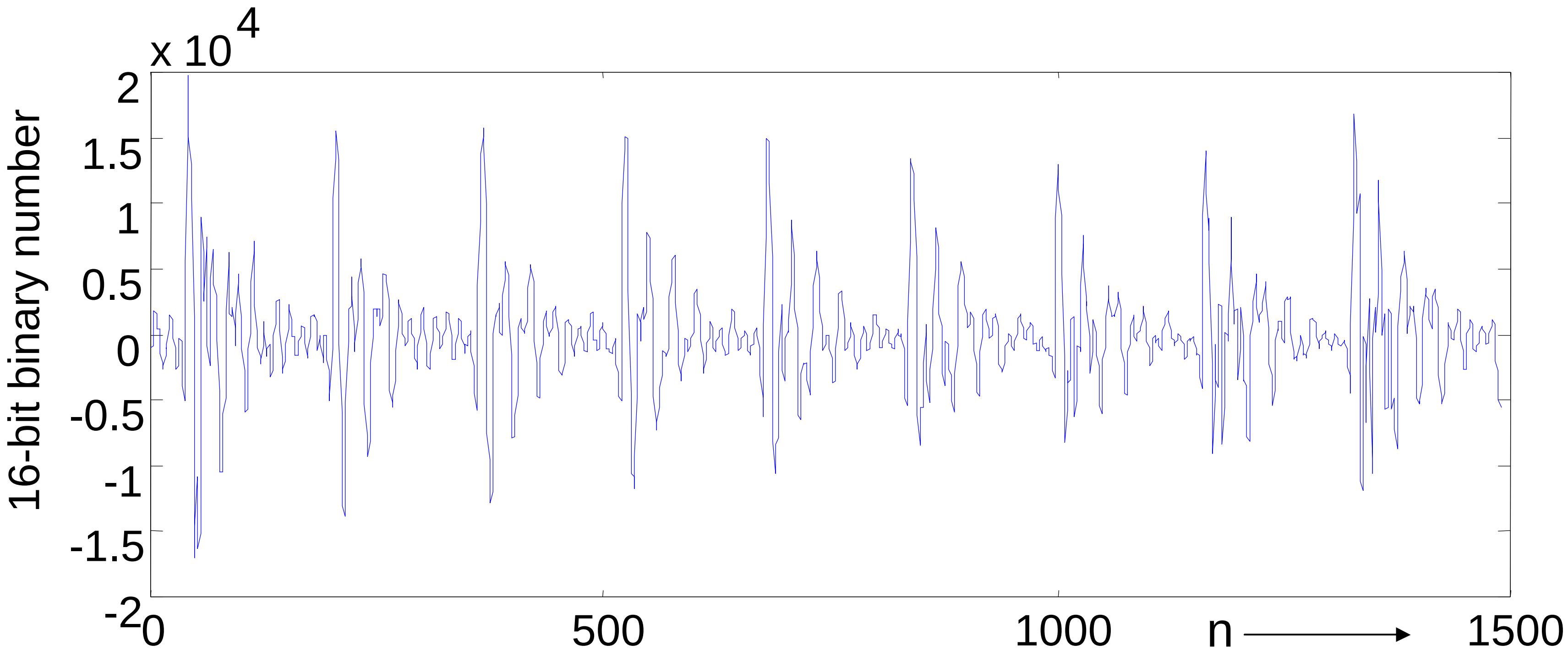
= 0.019 *22050 = 419 Hz

50 ms segment of music sampled at 22.05 kHz



What’s the frequency now, folks?

Voiced telephone speech sampled at 16000 Hz



Short (≈ 100ms) segment of a vowel. [1ms = 1/1000 second]

Vowels are approximately periodic.

What is the pitch of the voice here?

Ans: ≈ 9 cycles in 1/10 s i.e. ≈ 90 Hz - probably male speech

Use of Python + Numpy, Scipy etc

- Mobile phones do a lot of real time processing on signals obtained directly from the ADC
- They also generate output signals for DAC in real time.
- Signals are ‘buffered’ for processing in blocks
- More on this later
- To test & understand algorithms, use Python with input signals read from files & output signals written to files.
- Need the extra libraries (Numpy, etc.) for dealing with arrays & providing useful functions
- (e.g. for reading in audio files, plotting waveforms, etc)

‘resample’

- yr = resample(y, (y.size)*P)
- Changes sampling rate of y by factor P
- Reduces it if P <1 (decimation)
- Increases it if P >1 (up-sampling)
- Filtering is carried out by this function.
- If P = 0.5, y is filtered before it is resampled.

Storing signals in files

- Sound files are commonly stored in a ‘.wav’ format.
- Cannot be read as text because the info is compactly represented as direct binary numbers.
- 1000, -2000 would not be represented by:

  ‘1’, ‘0’, ‘0’, ‘0’, ‘-‘, ‘2’, ‘0’, ‘0’, ‘0’ .
- Instead just four 8-bit binary numbers would be stored: 11101000 00000111 00110000 11111000.
- First representation is ‘text’, ‘ascii’ or ‘formatted’
- Second representation is ‘binary’ or ‘unformatted’.
- Term ‘binary’ is misleading as all representations use binary numbers.
- Audacity can read wav files easily.

Functions for reading/writing *.wav

```
from scipy.io import wavfile
(y, Fs) = wavfile.read(“cap4th.wav”);
wavfile.write(“outfile.wav”, Fs,y);

from Ipython.display import Audio
audio(y,rate=Fs,);            # listen directly
```

Sound files provided

See

www.cs.man.ac.uk/~barry/mydocs/COMP28512/MS15_Labs

for examples of speech & music files all in wav format.

Frequency spectrum & sampling

- Speech & music need of lots of sine-waves added together.
- Harmonically related, i.e. *f, 2f, 3f, 4f, 5f,* ...
- Other sounds approximated as sum of lots of very small sine-waves.
- Any type of signal has a ‘frequency spectrum’.
- Most energy in **speech** is within 300 to 3400 Hz.
- Energy in **music** that we can hear is within 50 to 20,000 Hz.

‘Sampling Theorem’

- If a signal has all its spectral energy below B Hz,
- & is sampled at Fs Hz where Fs ≥ 2B, it can be reconstructed exactly from the samples. Nothing is lost.

- Speech below 4000 Hz can be sampled at 8000 Hz.
- Music below 20 kHz can be sampled at 40 kHz
- In practice speech is filtered to 50-3400 Hz
- And music is sampled at 44.1 kHz.
- When we process a signal whose sampling rate is Fs Hz, we can only observe frequencies in range 0 to Fs/2.

‘Aliasing’

- What happens if you sampled a speech or music signal at **Fs** & it has some spectral energy above **Fs/2** ?
- Leads to a form of distortion known as ‘aliasing’.
- Lab experiment on this.
- Must filter off all spectral energy at & above **Fs/2** Hz before sampling at **Fs**.

Fixed & floating point arithmetic

- Real time DSP often uses 'fixed point' operations since they consume less power than 'floating point' ones.
- Fixed point operations deal with all numbers as integers.
- Numbers often restricted to a word-length of only 16 bits.
- When we use NUMPY for DSP, we have floating point operations available with word-lengths much larger than 16 bits.
- This makes the task much easier.
- When simulating real time DSP in Python, we can restrict programs to integer arithmetic, & this is useful for testing.

Reading List

Core Text

Title: Computer networks (5th edition)

Author: Tanenbaum, Andrew S. and David Wetherall

ISBN: 9780132126953

Publisher: Prentice Hall

Edition: 5th

Year: 2010