BIOM9660 Assignment

Software Design and more….

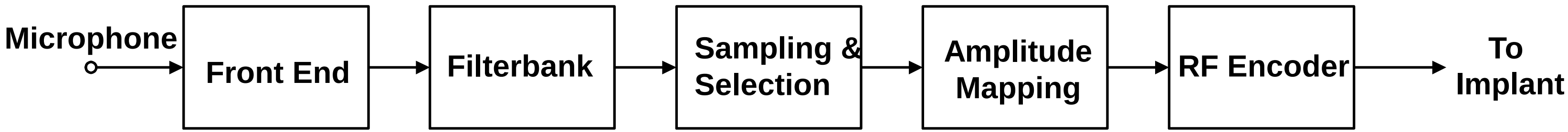
Never Stand Still

Faculty of Engineering

Graduate School of Biomedical Engineering

Greg Watkins – August 2016

- **Software Design & Submission**
- Frequency/Time Matrix
- FFT as a band pass filter
- Amplitude Mapping



Dr B.Swanson – Cochlear Implant Sound Processing model

Software – Files Provided

- Three MatLab files will be provided
  1. **cochlearProc.m** - runs your Cochlear Implant simulation
  2. **classCochlear.m** - skeleton of functions you need to write
  3. **classCochlearSupport.m** - “helper” functions/data

- These files should be copied to a single folder on your PC

- <u>The only file you may modify is #2 – more later</u>

classCochlear.m – *Insert your code (only) here!*

- This file provides the declarations for 6 functions that you must write
  - The declarations (names and parameters) must not be changed
  - Comments provide a brief description of what the function should do. The details are up to you.
- For example
  - function result = getWav(obj, name)

    % read the wav file <name> and resample if required so

    % that the sampling frequency is fSample +/- fTolerance

getWav

getFTM

process

applyDR

plotSignal

plotElectrodogram

Software –Testing

- To test your simulation in MatLab type….

cochlearProc(soundFileName, processingType)

Where

soundFileName is the name of a sound file eg ‘hello.wav’,
processingType is 1 or 2 or 3 (see assignment spec)

- This function will call the functions you have written and save the results to disk. Eg output files for “Speak” are:-
  - Speak.jpg (electrodogram)
  - Speak_speech.jpg (time domain speech)
  - Speak.csv (stimulation file)
  - Speak.wav (vocoded sound file)

Other data and functions

- classCochlearSupport.m provides helper functions and data
- For example
  - `fSample      = 16000; % required sampling frequency`
  - `fTolerance   = 0.1;   % allowed variation from fSample`
  - `tSample      = 0.002; % sample time for cochlear implant processing`
  - `frameOverlap = 3/4;   % required overlap when using Hann window`
  - `numElectrodes= 16;    % number of electrodes in cochlear implant`
  - `numFormants  = 2;     % formants used by the F0F1F2 strategy`
  - `dynamicRange = 10;    % dB`
  - `maxOutput    = 1024;`
  - `function result = procName(obj, procType)`

    `% return string containing the name of the process type <procType>`
- Access data using *obj.<name>* eg obj.numElectrodes

Software Submission

- Only your file **classCochlear.m** is submitted via Moodle

- When you submit classCochlear.m you must rename it to 'class<znumber>' *(eg classz1234567)*
  - Moodle will ask for two versions 'class<znumber>**.m'** and 'class<znumber>**.txt'** *(eg classz1234567.m and classz1234567.txt)*
  - <u>If your files have other names they will not be marked.</u>

- cochlearProc.m and classCochlearSupport.m are <u>not</u> submitted.
  - <u>Any changes will not be used when your code is marked.</u>

- Software Design & Submission
- **Frequency/Time Matrix**
- FFT as a band pass filter
- Amplitude Mapping

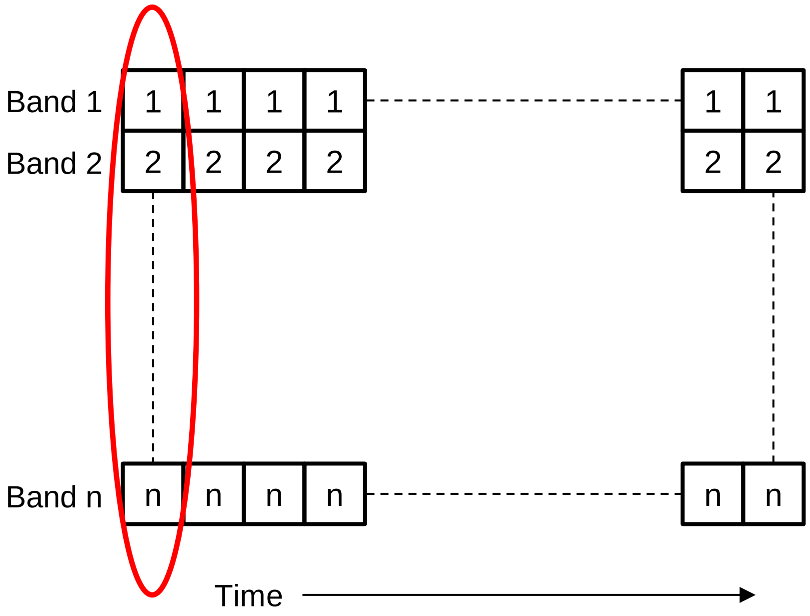
Dr B.Swanson – Cochlear Implant Sound Processing model

Frequency/Time matrix (FTM)

- Every 2mS your simulation will take a sample (frame or epoch) of the input signal and pass it through n parallel band pass filters (BPF)

- This provides a matrix as shown at right called the “Frequency/ time matrix” or FTM

- The FTM is the key data structure in your simulation

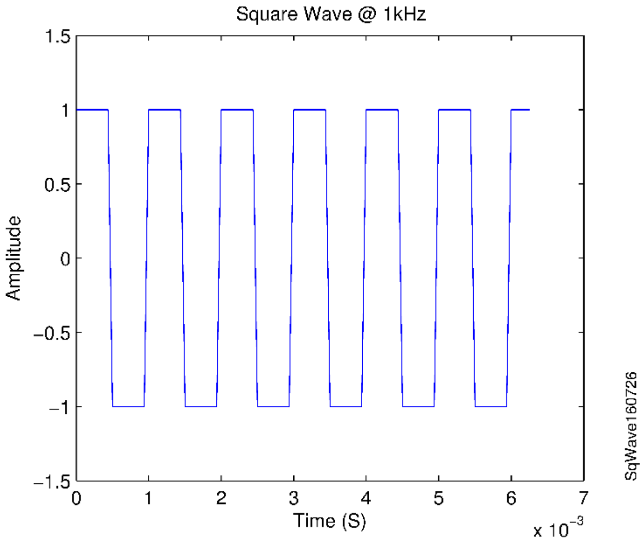


- Software Design & Submission
- Frequency/Time Matrix
- **FFT as a band pass filter**
- Amplitude Mapping

Dr B.Swanson – Cochlear Implant Sound Processing model

FFT – A possible BPF implementation

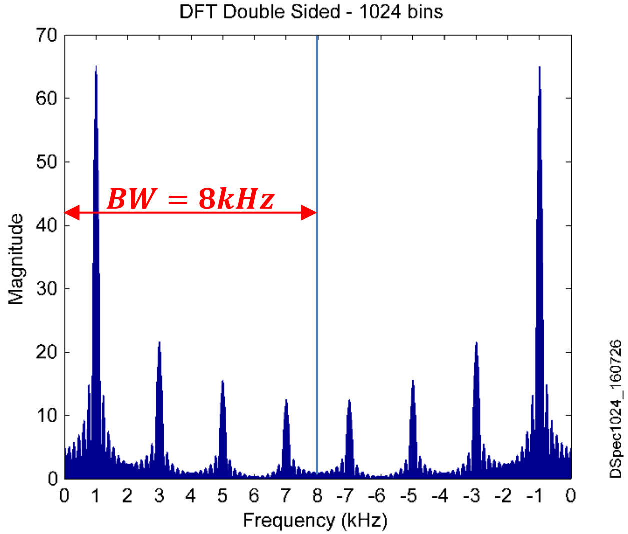
- As an example… take a sample of a 1kHz Square wave
- Nyquist theorem: $\mathrm{F_s} \geq 2\mathrm{BW}$. For $\mathrm{F_s}$=16kHz max bandwidth is 8kHz.



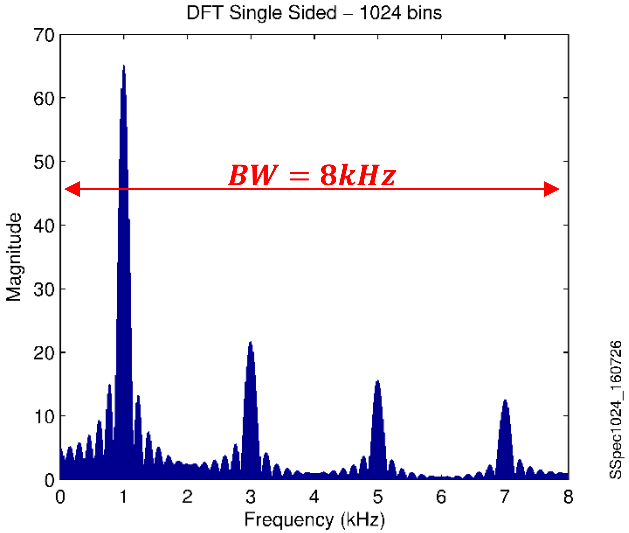
Sampled waveform split into “frames” and FFT applied.

MatLab FFT

Use abs() function to obtain magnitude of bin data



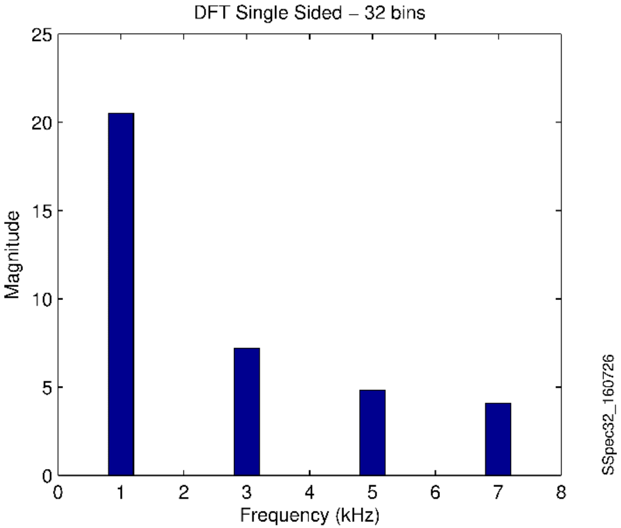
BPF/FFT



In cochlear implants FFT used to implement n x bandpass filters

Need to choose number of bins appropriately

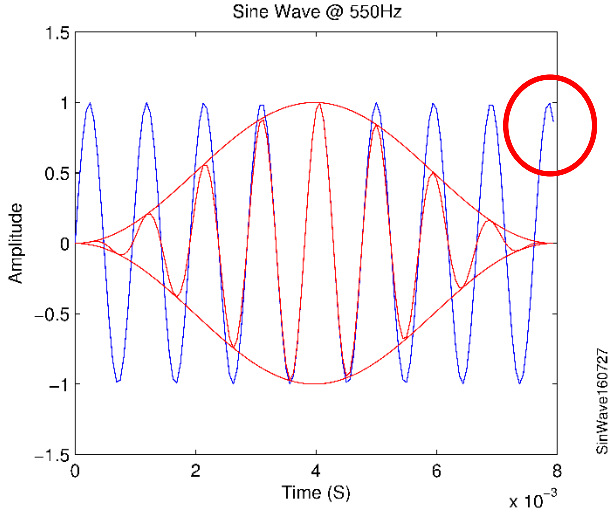
The FFT is a simple way to implement the required Band-pass filters



- The example shown has a wave form that fits perfectly in the sample frame. Very neat plots result.
- In real-life and in particular with speech this is not always true….

Hann Window

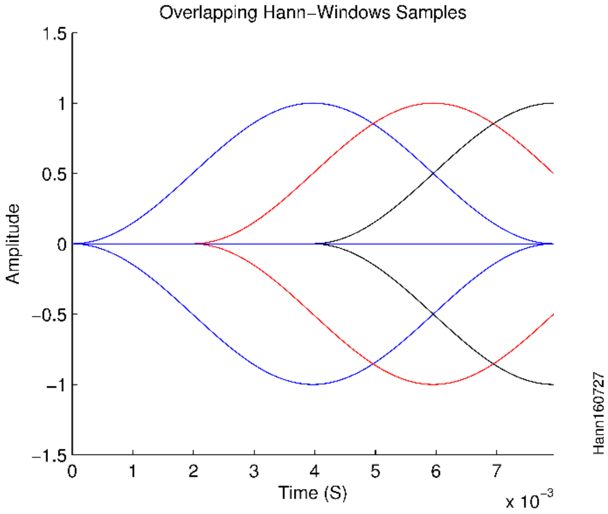
- Real-life signals, especially speech, don’t fit neatly in sample window.
- Windowing and overlapped sampling are applied



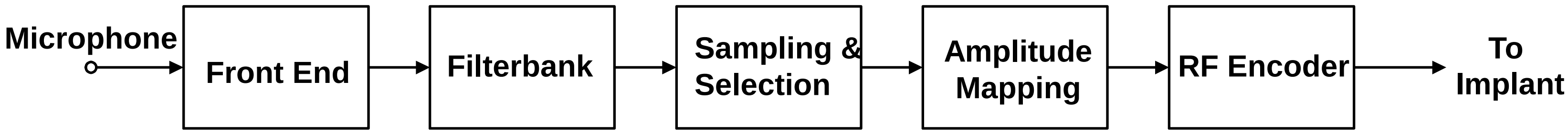
Plot at left shows a wave form that does not fit neatly in the sample window.

Hann window (red) is applied to remove transients

Plot at right shows n x overlapping, samples with hann window applied as required in the assignment



- Software Design & Submission
- Frequency/Time Matrix
- FFT as a band pass filter
- **Amplitude Mapping**



Dr B.Swanson – Cochlear Implant Sound Processing model

Amplitude Mapping

- Referring to Dr Swanson's processing model the "Amplitude Mapping" block contains a number of functions including:-
    1. Ensuring that levels are limited within specific bounds (eg 0 to 1)
    2. Mapping the levels to an individuals T&C levels
- The assignment "stimulus amplitude range" requirement is similar to 1) + 2) above.

Amplitude Mapping – T & C levels

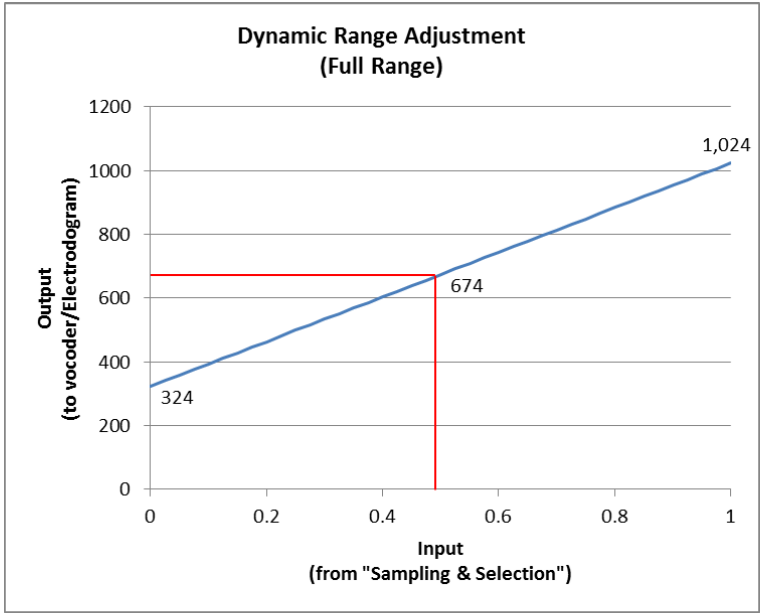
- The assignment specifies
  - maximum or "C" level of 1024 and
  - dynamic range (difference between C and T levels) of 10dB.
- Rearranging the equation from the assignment instructions gives

  $$\mathrm{A_{min}} = \frac{\mathrm{A_{max}}}{\sqrt{10}} = 324 = "T"$$

  where A_{max} is the "C" level and A_{min} is the "T" level
- This tells us that the amplitude information presented to the Vocoder or plotted in the electrodogram must always be in the range 324 to 1024.

Amplitude Mapping – Scaling

- Let's assume that the amplitude of the information output from the processing strategy in the "Sampling & Selection" block (eg ACE) is in the range 0 to 1.
- In the simplest approach
  - An input amplitude of 0 is mapped to the "T" level
  - An input amplitude of 1 is mapped to the "C" level
  - Amplitudes in between are scaled between the "T" and "C" levels



Amplitude Mapping - Scaling

- As part of your assignment you might investigate whether more sophisticated mappings are more intelligible.
- The mapping at right maps any low amplitude input (eg < 0.2) to 0 and scales inputs > 0.2 between the T and C levels
- This approach is used in Cochlear Implants. You might like to consider the advantages of this approach