



UNIVERSITY OF  
NEWCASTLE UPON TYNE

SCHOOL OF ELECTRICAL, ELECTRONIC & COMPUTER  
ENGINEERING

**Digital Signal Processing**  
**EEE305 & EEE801("Part A")**

# Chapter 1

## Introduction

### 1.1 Syllabus

#### **Chapter 1 Introduction**

- 1.1 Syllabus
- 1.2 Recommended Books
- 1.3 Course Aims
- 1.4 What is Matlab/Simulink?
- 1.5 Organisation of Classes

#### **Chapter 2 The Fourier Transform**

- 2.1 Introduction
- 2.2 Derivation of the DFT
- 2.3 Matrix Interpretation of the DFT
- 2.4 The Fast Fourier Transform
- 2.5 Derivation of the FFT
- 2.6 Radix-2 FFT
- 2.7 Windowing
- 2.8 Spectral Leakage

#### **Chapter 3 Introduction to Digital Filters**

- 3.1 Introduction
- 3.2 Finite Impulse Response
- 3.3 Infinite Impulse Response
- 3.4 Low-Pass Filter Specifications
- 3.5 Realisation of Digital Filters
- 3.6 FIR Digital Filter Realisation
- 3.7 IIR Digital Filter Realisation

#### **Chapter 4 Design of FIR Filters**

- 4.1 Introduction
- 4.2 Fourier Transform Relationship
- 4.3 FIR Filter Design by Impulse Response Truncation (IRT)
- 4.4 Summary of FIR Filter Design Using The IRT Method
- 4.5 Optimality of the IRT Method
- 4.6 Gibb's Phenomenon
- 4.7 Filter Specification Requirements
- 4.8 FIR Filter Design Using Windows
- 4.9 Summary of FIR Filter Design Using The Window Method
- 4.10 Finite Word Length Effects in FIR Filters

#### **Chapter 5 Design of IIR Filters**

- 5.1 Introduction
- 5.2 IIR Filter Basics
- 5.3 Analogue Low-pass Filters
- 5.4 The Bilinear z-transform
- 5.5 Frequency Transformations
- 5.6 Z-Plane Poles and Zeros
- 5.7 Summary of IIR Filter Design Using the Bilinear z-transform
- 5.8 Finite Word Length Effects in IIR Filters
- 5.9 Finite Word Length Effects in Pole – Zero Placement

**Chapter 6 Describing Random Sequences**

- 6.1 Introduction
- 6.2 Mean, Mean-Square and Variance
- 6.3 Ensemble Averages and Time Averages
- 6.4 Autocorrelation
- 6.5 Power Spectrum
- 6.6 Wiener-Khintchine Power Theorem
- 6.7 Cross-Correlation
- 6.8 Cross-Correlation Coefficient
- 6.9 Cross Spectrum

**Chapter 7 Adaptive Filtering**

- 7.1 Introduction
- 7.2 Wiener Filter Theory
- 7.3 Performance Surface
- 7.4 The Steepest Descent Algorithm
- 7.5 The LMS Algorithm
- 7.6 Applications of Closed-Loop Adaptation

**Chapter 8 Sampling, Aliasing and Data Conversion**

- 8.1 Introduction
- 8.2 Nyquist Sampling Theorem
- 8.3 Aliasing
- 8.4 Anti-alias Filter Requirements
- 8.5 Modelling of Quantisation Noise
- 8.6 Derivation of the Mean Square Error of the Quantisation Process
- 8.7 Dynamic Ranges

**Chapter 9 Multirate Digital Signal Processing**

- 9.1 Introduction
- 9.2 Decimation
- 9.3 Interpolation
- 9.4 Frequency Transforms of Decimated and Expanded Sequences
- 9.5 Sampling-rate Conversion
- 9.6 Multistage Approach
- 9.7 Polyphase Filters
- 9.8 Applications of Multirate DSP

**1.2 Recommended Books**

Course Handouts

“A Course in Digital Signal Processing”. Boaz Porat, Wiley, 1997, ISBN: 0-471-14961-6

“Digital Signal Processing”, A Practical Approach”, E.C. Ifeachor, B.W. Jervis, Addison-Wesley, 1993, ISBN: 0-201-54413-X

“Digital Signal Processing”, J.G. Proakis, D.G. Manolakis, Prentice Hall, 1996, ISBN: 0-13-373762-4

### 1.3 Course Outcomes (2002/3)

This section gives a statement of the main outcomes that you should have achieved by the end of the course. Since the content is ambitious, particular aspects will be focussed upon closer to the examination.

#### Matlab/Simulink

1. You should become familiar with the use of Matlab/Simulink to simulate DSP in practice.
2. You should be aware of how Matlab can be used in the design chain from DSP algorithm, through simulation and test, to implementation in ‘C’ and/or assembler, and thence to object code in a special purpose DSP sub-system.

---

#### QUESTION 4

##### Chapter 2 – Fourier Transform & FFT

3. You should understand the derivation of a DFT, and be able to write down the matrix representation.
4. You should be able to explain how the DFT is analogous to a baseband frequency analyser.
5. You should know how to interpret the DFT output as a complex spectrum, how to calculate the frequency corresponding to each bin, and determine the magnitude and phase of each bin from the complex DFT output.
6. You should be able to explain spectral leakage due to windowing in terms of convolution in the spectral domain, and be able to calculate the magnitude of leakage for a rectangular window.
7. You should understand how different windows (Hamming, etc) reduce leakage effects, and understand the criteria for selecting the best window for a particular application.
8. You should understand the method of zero padding to interpolate the DFT output, and to understand how it can be used to improve signal detection.
9. You should be able to derive the DIT FFT algorithm, and trace through the evaluation of a 4 or 8 length FFT by hand.
10. You should understand the difference between “in-place” and “constant geometry” implementations, be able to estimate FFT computational complexity, and be able to describe a possible hardware architecture using constant geometry.

---

#### QUESTION 1

##### Chapter 3 – Introduction to Digital Filters

11. You should know the difference between FIR and IIR digital filters and their relative advantages and disadvantages.
12. You should understand the principle of fast convolution using the overlap-add method.
13. You should be able to draw the realisation (in the form of a signal flow diagram) of an IIR digital filter in its direct, canonic, series and parallel forms.

##### Chapter 4 – Design of FIR Filters

14. You should know how to derive FIR filter coefficients using the Fourier relationship between frequency response and impulse response.
15. You should understand why “symmetrical” FIR coefficients results in a linear phase filter.
16. You should be able to explain why truncating an “infinite” FIR filter with a rectangular window leads to Gibbs phenomenon (i.e. a sinc function distortion at the edge of the transition band) due to the convolution of the ideal filter response with the rectangular window frequency response.
17. You should know how to select other window functions to achieve specified transition band width and ripple magnitude.

##### Chapter 5 – Design of IIR Filters

18. You should be able to derive and understand the design of IIR filters using the bilinear z-transform method. You should be able to use this methods to design simple IIR filters given an analogue prototype.
19. You should be able to explain why finite wordlengths in representing IIR filter coefficients can lead to incorrect characteristics, and even to instability.

**QUESTION 3****Chapter 6 – Describing Random Sequences**

20. You should know what is meant by autocorrelation function, autocorrelation matrix, cross-correlation function, variance, autocovariance, and the Wiener-Khinchine Theorem (i.e. the DFT relationship between power spectrum and autocorrelation function).
21. You should know in detail the form of the autocorrelation function for the following waveforms: DC, sine wave, white noise, lowpass noise.
22. You should be able to derive the cross-correlation of two uncorrelated waveforms (such as a signal and noise).

**Chapter 7 – Adaptive Filtering**

23. You should understand the significance of the Wiener optimal estimation filter, and be able to derive the Wiener-Hopf equation for the filter (including the autocorrelation matrix).
24. You should understand and be able to derive the steepest descent algorithm for solving for the Wiener filter.
25. You should be able to derive the LMS adaptive filter algorithm, and explain the effect of varying the convergence constant  $\mu$ .
26. You should know the three main application types for the adaptive filter: the direct, inverse, and linear predictive models.
27. You should know how to apply the adaptive filter in various applications such as: adaptive noise cancellation for car-phones, adaptive mains interference cancellation in ECGs, adaptive echo-cancellation in telephony, adaptive equalisation in digital receivers.

**QUESTION 2****Chapter 8 – Sampling, Aliasing and Data Conversion**

28. You should know the advantages and disadvantages of digital filters as compared with analogue.
29. You should be familiar with the spectrum of sampled signals (i.e. the repeated spectrum), and explain this in terms of convolution between the signal spectrum and the spectrum of the delta function sampling sequence.
30. You should be able to explain aliasing, and to be able to determine the required order of an anti-alias filter for a given passband ripple, worst-case signal-to-aliasing ratio, and sampling frequency.
31. You should be familiar with the terminology and definition of sample and hold devices.
32. You should understand, and be able to derive, the signal-to-quantisation-noise (SQNR) formula for a sine wave, and for any signals where the MSV is known in terms of the peak-to-peak amplitude (i.e. the use of  $q^2/12$ , etc).
33. You should be able to determine the number of bits required for an ADC, taking account of the required SQNR and dynamic range of the signal.
34. You should understand the effect of nonlinearities in an ADC (i.e. inter-harmonic distortion).

**Chapter 9 – Multirate Digital Signal Processing**

35. In the context of multirate DSP, you should be able to explain decimation and interpolation in the time and frequency domains.
36. You should know how to achieve a non-integer sampling rate change (such as for CD to DAT sampling rates), by using a combination of interpolation and decimation.
37. You should be able to explain how the computational complexity of the combined interpolation/decimation filter in the point above can be reduced by exploiting zero values and reduced rate on the output.
38. You should understand the reduction in quantisation SNR due to oversampling, and be able to use this in an ADC to determine the reduction in the number of ADC bits for a given SNR [i.e. the  $(\log_2 R)/2$  formula, where  $R$  is the sampling rate change].

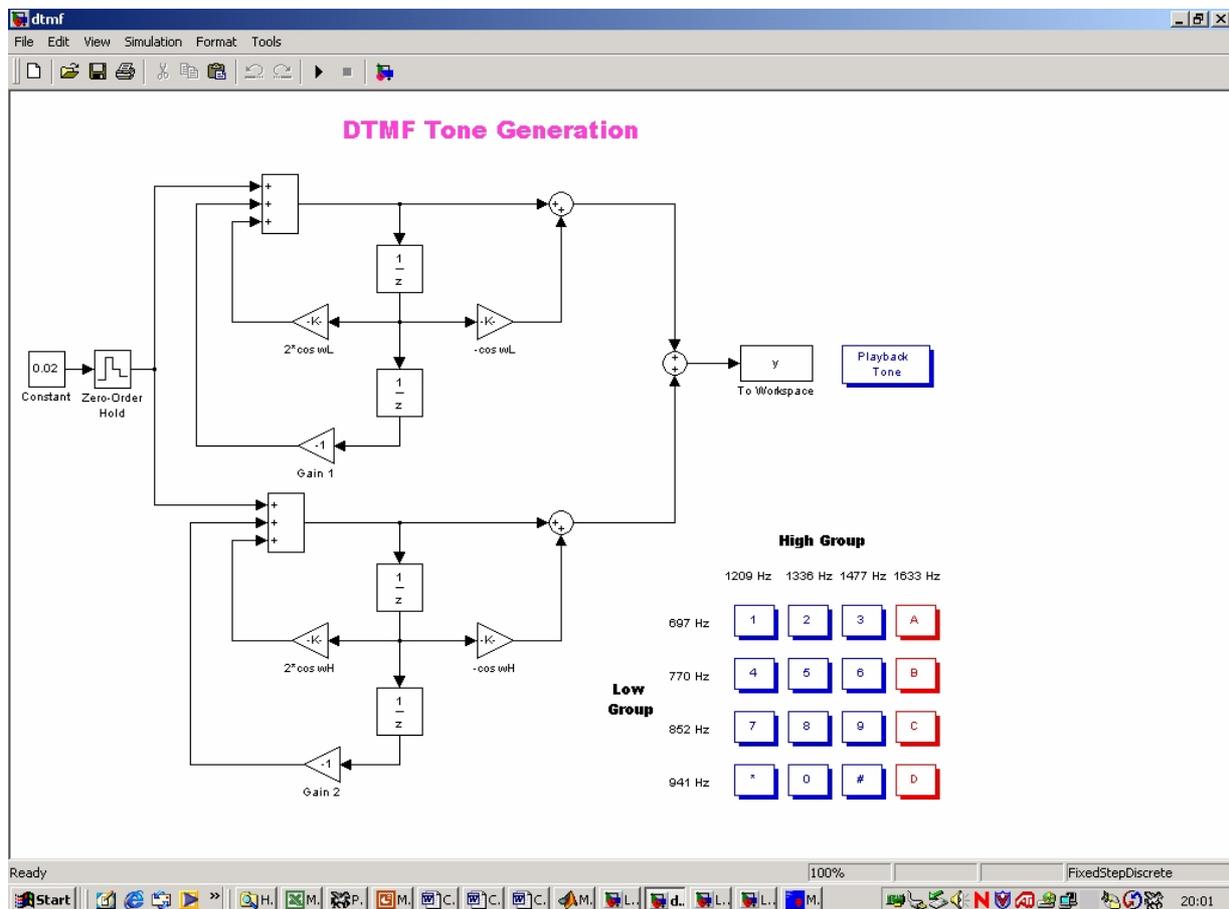
## 1.4 What is Matlab/Simulink?

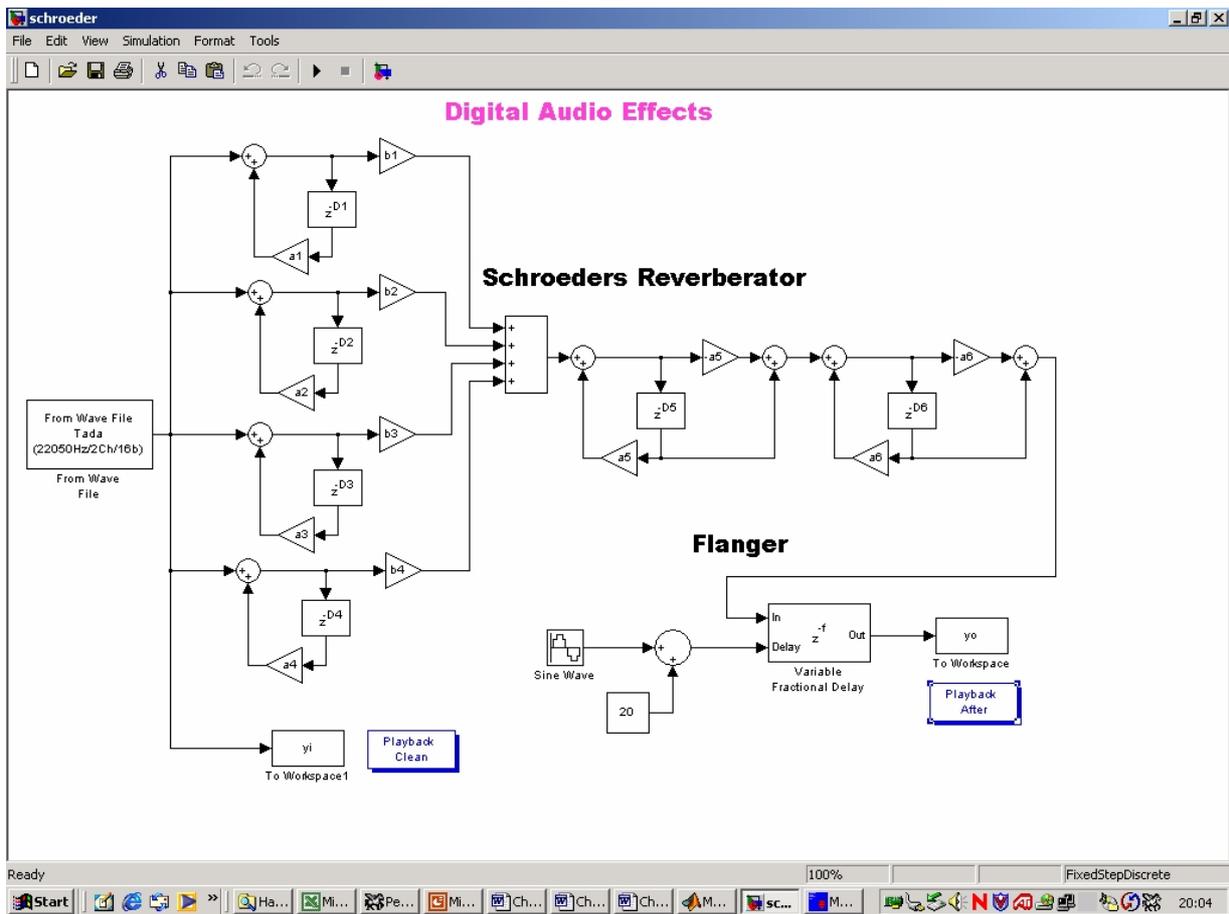
Matlab and Simulink were designed to make the programming of DSP and signal flow graphs easy. It is now a very mature programming language/system that is widely used in industry. The School has a flexible licence for all Versions of the software, including almost all of the so-called Toolboxes. Initially, we will be using Version 6.0 because later versions still have difficulties in running within the Windows 2000 Common Desktop. The licence covers the installation of this software on students’ own PCs, providing it is in connection with education within the School. A CD is available from EECE Reception for overnight loan for this purpose.

There are two main modes of using the system: “batch processing” vector/matrix language (Matlab), and “real-time signal flow graph processing (Simulink). When developing a DSP system, a typical set of design steps would be:

- (a) Identify the algorithm required and use Matlab to design the appropriate parameters.
- (b) Simulate the DSP system, using either Simulink or Matlab with real data acquired through the PC using Matlab, and analyse the results to check for correctness, etc.
- (c) Translate the Matlab code into ‘C’ or Assembler, for cross compilation onto the target specialist DSP device, such as the TMS320C60 or SHARC processor. This translation can either be done by hand, or by using one of the increasingly available Matlab compilers, depending on the efficiency of code required.

### 1.4.1 Simulink Examples





## 1.5 Organisation of Classes & Resources

### 1.4.1 Lectures

EEE305 & EEE801: Tuesday, 11.00-12.00 in Claremont Tower CLT.L120  
Tuesday, 12.00-13.00 in Merz L101  
Lecturer: Dr Lok Woo

EEE801 only: Friday, 16.00-17.00 in Bedson L.G.38  
Lecturer: Dr Harris Tsimenidis

Detailed course handouts will be provided. Lectures will explain the main concepts covered in the notes, with the help of powerpoint material.

### 1.4.2 Matlab/Simulink Tutorials

In the first week or two of the DSP course, you will learn how to use this language by attending taught sessions in one of the University PC cluster rooms, where you will be given a Tutorial booklet. Undergraduate and MSc students will be split into groups, each group has been scheduled according to a timetable that you will be given. MSc students will attend two 2 hour sessions; undergraduates will have one 2 hour session. After these sessions, you are expected to work through most of the tutorial book in your own time, so you have a good working knowledge of Matlab/Simulink.

During the 10 weeks following, there will be further sessions in the Cluster Rooms at which various DSP assignments should be undertaken. These assignments have been designed to help understand the theory presented in the lectures and notes.

The cluster rooms are available outside of the timetabled hours for those who wish to spend longer.

### 1.4.3 Examinations

EEE305: Duration 2 hours; answer 3 questions out of 4.

EEE801: Duration 3 hours; two sections, each of 3 questions each; answer 2 questions from each section. Section A on Tuesday lecture material, section B on Friday afternoon lecture material.

For both exams, at least one of the questions on the Tuesday lecture material will be heavily based on the Matlab Tutorials.

### 1.4.4 Web Site

Lecture notes, copies of the simulink programs, and various other materials can be found on the web site:

<http://www.staff.ncl.ac.uk/oliver.hinton/>