

CONTENTS

<i>Preface</i>	<i>vii</i>
<i>List of Symbols and Abbreviations</i>	<i>xxi</i>

CHAPTER 1: Introduction to Digital Signal Processing	1.1–1.8
---	----------------

- 1.1 Introduction 1.1
- 1.2 Signal 1.2
- 1.3 Discrete Time System 1.2
- 1.4 Analysis of Discrete Time System 1.3
- 1.5 Filters 1.5
- 1.6 Finite Word Length Effects 1.5
- 1.7 Multirate DSP 1.6
- 1.8 Energy and Power Spectrum 1.6
- 1.9 Digital Signal Processors 1.6
- 1.10 Importance of Digital Signal Processing 1.7
- 1.11 Use of MATLAB in Digital Signal Processing 1.8

CHAPTER 2: Discrete Time Signals and Systems	2.1–2.182
---	------------------

- 2.1 Discrete Time Signals 2.1
 - 2.1.1 Generation of Discrete Time Signals 2.1
 - 2.1.2 Digital Signal 2.2
 - 2.1.3 Representation of Discrete Time Signals 2.4
 - 2.1.4 Standard Discrete Time Signals 2.6
 - 2.1.5 Representation of a Discrete Time Signal as Summation of Impulses 2.9
 - 2.1.6 Mathematical Operations on Discrete Time Signals 2.10
- 2.2 Baseband Sampling 2.17
 - 2.2.1 Sampling and Aliasing 2.18
- 2.3 Classification of Discrete Time Signals 2.22
 - 2.3.1 Deterministic and Nondeterministic (Random) Signals 2.22
 - 2.3.2 Periodic and Nonperiodic Signals 2.23
 - 2.3.3 Symmetric (Even) and Antisymmetric (Odd) Signals 2.28
 - 2.3.4 Energy and Power Signals 2.31
 - 2.3.5 Causal, Noncausal and Anticausal Signals 2.34
 - 2.3.6 Absolutely Summable and Nonsummable Signals 2.34

- 2.4 Discrete Time System 2.35
 - 2.4.1 Linear Time Invariant (LTI) System 2.35
 - 2.4.2 Response of Discrete Time System 2.35
 - 2.4.3 Impulse and Step Response 2.36
 - 2.4.4 Difference Equation Governing Discrete Time System (Mathematical Representation of Discrete Time System) 2.36
- 2.5 Block Diagram and Signal Flow Graph Representation of Discrete Time System 2.37
- 2.6 Response of LTI Discrete Time System in Time Domain 2.40
 - 2.6.1 Zero-Input Response or Homogeneous Solution 2.40
 - 2.6.2 Particular Solution 2.41
 - 2.6.3 Zero-State Response 2.42
 - 2.6.4 Total Response 2.42
- 2.7 Classification of Discrete Time Systems 2.46
 - 2.7.1 Static and Dynamic Systems 2.47
 - 2.7.2 Time Invariant and Time Variant Systems (Shift Invariant and Variant Systems) 2.47
 - 2.7.3 Linear and Nonlinear Systems 2.51
 - 2.7.4 Causal and Noncausal Systems (Realizable and nonrealizable system) 2.58
 - 2.7.5 Stable and Unstable Systems 2.60
 - 2.7.6 FIR and IIR Systems 2.64
 - 2.7.7 Recursive and Nonrecursive Systems 2.65
 - 2.7.8 Invertible and Noninvertible Systems 2.65
- 2.8 Convolution Sum 2.81
 - 2.8.1 Discrete or Linear Convolution 2.81
 - 2.8.2 Representation of Discrete Time Signal as Summation of Impulses 2.83
 - 2.8.3 Response of LTI Discrete Time System Using Discrete Convolution 2.84
 - 2.8.4 Properties of Linear Convolution 2.85
 - 2.8.5 Methods of Performing Linear Convolution 2.86
 - 2.8.6 Convolution of Impulse Signal 2.88
 - 2.8.7 Unit Step Response Using Convolution 2.88
- 2.9 Interconnections of Discrete Time Systems 2.100
 - 2.9.1 Cascade Connected Discrete Time Systems 2.100
 - 2.9.2 Parallel Connected Discrete Time Systems 2.100
- 2.10 Circular Convolution 2.107
 - 2.10.1 Circular Representation and Circular Shift of Discrete Time Signal 2.107
 - 2.10.2 Circular Symmetries of Discrete Time Signal 2.109
 - 2.10.3 Definition of Circular Convolution 2.110
 - 2.10.4 Procedure for Evaluating Circular Convolution 2.111
 - 2.10.5 Linear Convolution via Circular Convolution 2.111
 - 2.10.6 Methods of Computing Circular Convolution 2.112

- 2.11 Sectioned Convolution 2.121
 - 2.11.1 Overlap Add Method 2.122
 - 2.11.2 Overlap Save Method 2.122
- 2.12 Inverse System and Deconvolution 2.138
 - 2.12.1 Inverse System 2.138
 - 2.12.2 Deconvolution 2.139
- 2.13 Correlation, Crosscorrelation and Autocorrelation 2.141
 - 2.13.1 Procedure for Evaluating Correlation 2.142
- 2.14 Circular Correlation 2.151
 - 2.14.1 Procedure for Evaluating Circular Correlation 2.151
 - 2.14.2 Methods of Computing Circular Correlation 2.152
- 2.15 Summary of Important Concepts 2.156
- 2.16 Short-Answer Questions 2.157
- 2.17 MATLAB Programs 2.167
- 2.18 Exercises 2.172

CHAPTER 3: z -Transform**3.1–3.164**

- 3.1 Introduction 3.1
 - 3.1.1 z -Plane 3.1
 - 3.1.2 Definition of z -Transform 3.1
 - 3.1.3 Inverse z -Transform 3.3
 - 3.1.4 Poles and Zeros of Rational Function of z . 3.4
- 3.2 Region of Convergence of z -Transform 3.6
 - 3.2.1 ROC of Discrete Time Signals 3.6
 - 3.2.2 ROC of Rational Function of z (ROC Using Poles) 3.11
 - 3.2.3 Properties of ROC 3.12
- 3.3 Properties of z -Transform 3.17
- 3.4 Various Methods of Computing Inverse z -Transform 3.31
 - 3.4.1 Inverse z -Transform by Contour Integration or Residue Method 3.31
 - 3.4.2 Inverse z -Transform by Partial Fraction Expansion Method 3.33
 - 3.4.3 Inverse z -Transform by Power Series Expansion Method (Long Division Method) 3.35
- 3.5 Analysis of LTI Discrete Time System Using z -Transform 3.54
 - 3.5.1 Transfer Function of LTI Discrete Time Systems 3.54
 - 3.5.2 Impulse Response and Transfer Function 3.55
 - 3.5.3 Response of LTI Discrete Time System Using z -Transform 3.55
 - 3.5.4 Convolution and Deconvolution Using z -Transform 3.57
 - 3.5.5 Stability in z -Domain 3.57
- 3.6 Relation between Laplace Transform and z -Transform 3.63
 - 3.6.1 Impulse Train Sampling of Continuous Time Signals 3.63
 - 3.6.2 Transformation from Laplace Transform to z -Transform 3.63
 - 3.6.3 Relation between s -Plane and z -Plane 3.64

- 3.7 Structures for Realization of LTI Discrete Time Systems in z-Domain 3.87
 - 3.7.1 Discrete Time IIR and FIR Systems 3.87
 - 3.7.2 Discrete Time IIR Systems 3.87
 - 3.7.3 Discrete Time FIR Systems 3.88
- 3.8 Structure for Realization of IIR Systems 3.89
 - 3.8.1 Direct Form-I Structure of IIR Systems 3.90
 - 3.8.2 Direct Form-II Structure of IIR Systems 3.91
 - 3.8.3 Cascade Form Realization of IIR Systems 3.94
 - 3.8.4 Parallel Form Realization of IIR Systems 3.94
- 3.9 Structures for Realization of FIR Systems 3.122
 - 3.9.1 Direct Form Realization of FIR Systems 3.122
 - 3.9.2 Cascade Form Realization of FIR Systems 3.123
 - 3.9.3 Linear Phase Realization of FIR Systems (Realization with Minimum Number of Multipliers) 3.124
- 3.10 Summary of Important Concepts 3.131
- 3.11 Short-Answer Questions 3.133
- 3.12 MATLAB programs 3.145
- 3.13 Exercises 3.150

CHAPTER 4: Fourier Series and Fourier Transform of Discrete Time Signals 4.1–4.100

- 4.1 Introduction 4.1
- 4.2 Fourier Series of Discrete Time Signals (Discrete Time Fourier Series) 4.1
 - 4.2.1 Frequency Spectrum of Periodic Discrete Time Signals 4.3
 - 4.2.2 Properties of Discrete Time Fourier Series 4.4
- 4.3 Fourier Transform of Discrete Time Signals (Discrete Time Fourier Transform) 4.16
 - 4.3.1 Development of Discrete Time Fourier Transform from Discrete Time Fourier Series 4.16
 - 4.3.2 Definition of Discrete Time Fourier Transform 4.17
 - 4.3.3 Frequency Spectrum of Discrete Time Signal 4.18
 - 4.3.4 Inverse Discrete Time Fourier Transform 4.19
 - 4.3.5 Comparison of Fourier Transform of Discrete and Continuous Time Signals 4.20
 - 4.3.6 Properties of Discrete Time Fourier Transform 4.20
- 4.4 Discrete Time Fourier Transform of Periodic Discrete Time Signals 4.29
- 4.5 Analysis of LTI Discrete Time Systems in Frequency Domain 4.30
 - 4.5.1 Transfer Function of LTI Discrete Time Systems in Frequency Domain 4.30
 - 4.5.2 Impulse Response and Transfer Function 4.30
 - 4.5.3 Response of LTI Discrete Time System for Nonperiodic Inputs 4.31
 - 4.5.4 Response of LTI Discrete Time Systems for Periodic Inputs 4.31
 - 4.5.5 Impulse Response and Frequency Response of LTI Discrete Time Systems 4.32
 - 4.5.6 Properties of Frequency Response 4.33
 - 4.5.7 Frequency Response of First-Order Discrete Time Systems 4.34
 - 4.5.8 Frequency Response of Second-Order Discrete Time Systems 4.40

- 4.6 Relation between Continuous Time and Discrete Time Systems 4.45
 - 4.6.1 Spectra of Sampled Signal 4.45
 - 4.6.2 Aliasing in Frequency Spectrum Due to Sampling 4.47
 - 4.6.3 Antialiasing Filter 4.47
 - 4.6.4 Sampling of Bandpass Signal 4.47
 - 4.6.5 Signal Reconstruction (Recovery of Continuous Time Signal) 4.48
 - 4.6.6 Hold Circuits 4.49
- 4.7 Relation between z -Transform and Discrete Time Fourier Transform 4.50
- 4.8 Summary of Important Concepts 4.82
- 4.9 Short-Answer Questions 4.83
- 4.10 MATLAB Programs 4.88
- 4.11 Exercises 4.93

CHAPTER 5: Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT) 5.1–5.70

- 5.1 Introduction 5.1
- 5.2 Discrete Fourier Transform (DFT) of Discrete Time Signals 5.1
 - 5.2.1 Development of DFT from DTFT 5.1
 - 5.2.2 Definition of Discrete Fourier Transform (DFT) 5.2
 - 5.2.3 Frequency Spectrum Using DFT 5.3
 - 5.2.4 Inverse DFT 5.3
- 5.3 Properties of DFT 5.4
- 5.4 Relation between DFT and z -Transform 5.9
- 5.5 Analysis of LTI Discrete Time Systems Using DFT 5.10
- 5.6 Fast Fourier Transform (FFT) 5.21
- 5.7 Decimation in Time (DIT) Radix-2 FFT 5.24
 - 5.7.1 8-Point DFT Using Radix-2 DIT FFT 5.26
 - 5.7.2 Flow Graph for 8-Point DFT Using Radix-2 DIT FFT 5.30
- 5.8 Decimation in Frequency (DIF) Radix-2 FFT 5.32
 - 5.8.1 8-point DFT Using Radix-2 DIF FFT 5.35
 - 5.8.2 Flow Graph For 8-point DFT Using Radix-2 DIF FFT 5.37
 - 5.8.3 Comparison of DIT and DIF Radix-2 FFT 5.40
- 5.9 Computation of Inverse DFT Using FFT 5.40
- 5.10 Summary of Important Concepts 5.55
- 5.11 Short-Answer Questions 5.56
- 5.12 MATLAB Programs 5.59
- 5.13 Exercises 5.65

CHAPTER 6: FIR Filters

6.1–6.120

- 6.1 Introduction 6.1
- 6.2 LTI System as Frequency Selective Filters 6.2

- 6.3 Ideal Frequency Response of Linear Phase FIR Filters 6.4
- 6.4 Characteristics of FIR Filters with Linear Phase 6.6
- 6.5 Frequency Response of Linear Phase FIR Filter 6.8
- 6.6 Design Techniques for Linear Phase FIR Filters 6.20
- 6.7 Fourier Series Method of FIR Filter Design 6.22
- 6.8 Windows 6.40
 - 6.8.1 Rectangular Window 6.41
 - 6.8.2 Bartlett or Triangular Window 6.43
 - 6.8.3 Raised Cosine Windows 6.44
 - 6.8.4 Hanning Window 6.46
 - 6.8.5 Hamming Window 6.47
 - 6.8.6 Blackman Window 6.49
 - 6.8.7 Kaiser Window 6.50
 - 6.8.8 Summary of Various Features of Windows 6.54
- 6.9 FIR Filter Design Using Windows 6.54
- 6.10 Design of FIR Filters by Frequency Sampling Technique 6.79
- 6.11 Summary of Important Concepts 6.92
- 6.12 Short-Answer Questions 6.93
- 6.13 MATLAB Programs 6.102
- 6.14 Exercises 6.113

CHAPTER 7: IIR Filters

7.1–7.142

- 7.1 Introduction 7.1
- 7.2 Frequency Response of Analog and Digital IIR Filters 7.3
- 7.3 Impulse Invariant Transformation 7.6
 - 7.3.1 Relation between Analog and Digital Filter Poles in Impulse Invariant Transformation 7.7
 - 7.3.2 Relation between Analog and Digital Frequency in Impulse Invariant Transformation 7.8
 - 7.3.3 Useful Impulse Invariant Transformation 7.9
- 7.4 Bilinear Transformation 7.15
 - 7.4.1 Relation between Analog and Digital Filter Poles in Bilinear Transformation 7.16
 - 7.4.2 Relation between Analog and Digital Frequency in Bilinear Transformation 7.17
- 7.5 Specifications of Digital IIR Lowpass Filter 7.23
- 7.6 Design of Lowpass Digital Butterworth Filter 7.27
 - 7.6.1 Analog Butterworth Filter 7.27
 - 7.6.2 Poles of Butterworth Lowpass Filter 7.28
 - 7.6.3 Transfer Function of Analog Butterworth Lowpass Filter 7.34
 - 7.6.4 Frequency Response of Analog Lowpass Butterworth Filter 7.35
 - 7.6.5 Order of the Lowpass Butterworth Filter 7.35

- 7.6.6 Cutoff Frequency of Lowpass Butterworth Filter 7.36
- 7.6.7 Design Procedure for Lowpass Digital Butterworth IIR Filter 7.37
- 7.7 Design of Lowpass Digital Chebyshev Filter 7.39
 - 7.7.1 Transfer Function of Analog Chebyshev Lowpass Filter 7.40
 - 7.7.2 Order of Analog Lowpass Chebyshev Filter 7.41
 - 7.7.3 Cutoff Frequency of Analog Lowpass Chebyshev Filter 7.42
 - 7.7.4 Frequency Response of Analog Chebyshev Lowpass Filter 7.42
 - 7.7.5 Design Procedure for Lowpass Digital Chebyshev IIR Filter 7.43
- 7.8 Frequency Transformation 7.45
 - 7.8.1 Analog Frequency Transformation 7.45
 - 7.8.2 Digital Frequency Transformation 7.46
- 7.9 Summary of Important Concepts 7.103
- 7.10 Short-Answer Questions 7.105
- 7.11 MATLAB Programs 7.114
- 7.12 Exercises 7.134

CHAPTER 8: Finite Word Length Effects in Digital Filters

8.1–8.90

- 8.1 Introduction 8.1
- 8.2 Representation of Numbers in Digital Systems 8.2
 - 8.2.1 Binary Codes 8.2
 - 8.2.2 Radix Number System 8.3
 - 8.2.3 Fixed Point Representation 8.4
 - 8.2.4 Floating Point Representation 8.8
- 8.3 Types of Arithmetic in Digital Systems 8.12
 - 8.3.1 One's Complement Addition 8.12
 - 8.3.2 Two's Complement Addition 8.13
 - 8.3.3 Floating Point Addition 8.14
 - 8.3.4 Floating Point Multiplication 8.15
 - 8.3.5 Comparison of Fixed Point and Floating Point Arithmetic 8.16
- 8.4 Quantization by Truncation and Rounding 8.16
 - 8.4.1 Quantization Steps 8.17
 - 8.4.2 Truncation 8.18
 - 8.4.3 Rounding 8.22
- 8.5 Quantization of Input Data 8.23
- 8.6 Quantization of Filter Coefficients 8.29
- 8.7 Product Quantization Error 8.37
- 8.8 Limit Cycles in Recursive Systems 8.52
 - 8.8.1 Zero Input Limit Cycle 8.52
 - 8.8.2 Overflow Limit Cycle 8.60
 - 8.8.3 Scaling To Prevent Overflow 8.61

- 8.9 Summary of Important Concepts 8.72
- 8.10 Short-Answers Questions 8.74
- 8.11 Exercises 8.80

CHAPTER 9: Multirate DSP**9.1–9.66**

- 9.1 Introduction 9.1
- 9.2 Downsampling (or Decimation) 9.2
 - 9.2.1 Spectrum of Downsampler 9.4
 - 9.2.2 Anti-aliasing Filter 9.6
- 9.3 Upsampling (or Interpolation) 9.15
 - 9.3.1 Spectrum of Upsampler 9.18
 - 9.3.2 Anti-imaging Filter 9.19
- 9.4 Sampling Rate Conversion by a Rational Factor $1/D$ 9.23
 - 9.4.1 Spectrum of Sampling Rate Converter by a Rational Factor $1/D$ 9.24
- 9.5 Multistage Implementation of Sampling Rate Conversion 9.25
- 9.6 Identities in Multistage Digital Signal Processing 9.26
- 9.7 Implementation of Sampling Rate Conversion in FIR Filters 9.30
 - 9.7.1 Implementation of Sampling Rate Conversion Using Decimator in FIR Filters 9.30
 - 9.7.2 Implementation of Sampling Rate Conversion Using Interpolator in FIR Filters 9.32
- 9.8 Polyphase Decomposition 9.33
 - 9.8.1 Polyphase Decomposition of FIR Filters 9.33
 - 9.8.2 Polyphase Structure of Decimator 9.36
 - 9.8.3 Polyphase Structure of Interpolator 9.37
 - 9.8.4 Polyphase Decomposition of IIR Filters 9.39
- 9.9 Applications of Multirate DSP 9.44
 - 9.9.1 Digital Filter Banks 9.44
 - 9.9.2 Sub-Band Coding of Speech Signals 9.45
 - 9.9.3 Quadrature Mirror Filter (QMF) Bank 9.45
- 9.10 Summary of Important Concepts 9.46
- 9.11 Short-Answers Questions 9.47
- 9.12 MATLAB Program 9.51
- 9.13 Exercises 9.58

CHAPTER 10: Energy and Power Spectrum Estimation**10.1–10.44**

- 10.1 Introduction 10.1
- 10.2 Energy Spectrum of Discrete Time Signal 10.1
- 10.3 Random Signal and Random Process 10.3
- 10.4 Power Spectrum of Random Process 10.4
- 10.5 Periodogram 10.5
- 10.6 Use of DFT/FFT in Power Spectrum Estimation 10.6

- 10.7 Nonparametric Methods of Power Spectrum Estimation 10.6
 - 10.7.1 Bartlett Method of Power Spectrum Estimation 10.7
 - 10.7.2 Welch Method of Power Spectrum Estimation 10.8
 - 10.7.3 Blackman-Tukey Method of Power Spectrum Estimation 10.9
- 10.8 Performance Characteristics of Nonparametric Methods of Power Spectrum Estimation 10.22
 - 10.8.1 Performance Characteristics of Periodogram Power Spectrum Estimation 10.22
 - 10.8.2 Performance Characteristics of Bartlett Power Spectrum Estimation 10.24
 - 10.8.3 Performance Characteristics of Welch Power Spectrum Estimation 10.26
 - 10.8.4 Performance Characteristics of Blackman-Tukey Power Spectrum Estimation 10.27
 - 10.8.5 Summary of Performance Characteristics of Power Spectrum Estimation 10.29
- 10.9 Summary of Important Concepts 10.31
- 10.10 Short-Answers Questions 10.32
- 10.11 MATLAB Programs 10.34
- 10.12 Exercises 10.41

CHAPTER 11: Digital Signal Processors**11.1–11.86**

- 11.1 Introduction 11.1
- 11.2 Special Features of Digital Signal Processors 11.3
 - 11.2.1 Fast Data Access 11.3
 - 11.2.2 Fast Computation 11.5
 - 11.2.3 Numerical Fidelity 11.7
 - 11.2.4 Fast Execution Control 11.8
- 11.3 TMS320C5x Family of Digital Signal Processors 11.8
 - 11.3.1 Pin Diagram of TMS320C5x Processors 11.10
 - 11.3.2 Architecture of TMS320C5x Processors 11.14
 - 11.3.3 Functional Units in CPU of TMS320C5x Processors 11.15
 - 11.3.4 On-Chip Memory in TMS320C5x Processors 11.19
 - 11.3.5 On-Chip Peripherals of TMS320C5x Processors 11.19
 - 11.3.6 Addressing Modes of TMS320C5x Processors 11.21
 - 11.3.7 Instruction Pipelining in TMS320C5x Processors 11.23
 - 11.3.8 Instructions of TMS320C5x Processors 11.24
 - 11.3.9 Assembly Language Programs in TMS320C5x Processors 11.35
- 11.4 TMS320C54x Family of Digital Signal Processors 11.44
 - 11.4.1 Pin Diagram of TMS320C54x Processors 11.47
 - 11.4.2 Architecture of TMS320C54x Processors 11.50
 - 11.4.3 Functional Units in CPU of TMS320C54x Processors 11.51
 - 11.4.4 On-Chip Memory in TMS320C54x Processors 11.56
 - 11.4.5 On-Chip Peripherals of TMS320C54x Processors 11.57
 - 11.4.6 Addressing Modes of TMS320C54x Processors 11.60
 - 11.4.7 Instruction Pipelining in TMS320C54x Processors 11.63

- 11.4.8 Instruction set of TMS320C54x Processors 11.63
- 11.4.9 Assembly Language Programs in TMS320C54x Processors 11.72
- 11.5 Summary of Important Concepts 11.76
- 11.6 Short-Answers Questions 11.78
- 11.7 Exercises 11.81

CHAPTER 12: Applications of DSP**12.1–12.12**

- 12.1 Introduction 12.1
- 12.2 Speech Processing 12.1
 - 12.2.1 Speech Coding and Decoding 12.2
 - 12.2.2 Speech Recognition 12.4
 - 12.2.3 Speech Synthesis 12.5
 - 12.2.4 Digital Vocoder 12.6
- 12.3 Musical Sound Processing 12.7
 - 12.3.1 Digital Music Synthesis 12.7
 - 12.3.2 Musical Sound Processing for Recording 12.7
- 12.4 Digital Radio 12.8
- 12.5 Digital Television 12.9
- 12.6 DTMF in Telephone Dialing 12.9
- 12.7 Radar 12.10
- 12.8 Biomedical Signal Processing 12.11

APPENDICES**A.1–A.22**

- Appendix 1: Important Mathematical Relations A. 1
- Appendix 2: MATLAB Commands and Functions A. 4
- Appendix 3: Summary of Various Standard Transform Pairs A. 9
- Appendix 4: Summary of Properties of Various Transforms A. 12
- Appendix 5: Summary of Important Equations for FIR Filter Design A. 17
- Appendix 6: Summary of Important Equations for IIR Filter Design A. 21
- Appendix 7: Summary of Properties of Power Spectrum Estimator A. 22

INDEX**I.1–I.8**

CHAPTER 1

INTRODUCTION TO DIGITAL SIGNAL PROCESSING

1.1 Introduction

Digital Signal Processing (DSP) refers to processing of signals by digital systems like Personal Computers (PC) and systems designed using digital Integrated Circuits (ICs), microprocessors and microcontrollers. DSP gained popularity in the 1960s. Earlier, DSP systems were limited to general purpose non-real-time scientific and business applications. The rapid advancement in computers and IC fabrication technology leads to complete domination of DSP systems in both real-time and non-real-time applications in all fields of engineering and technology.

The basic components of a DSP system are shown in Fig 1.1. The **DSP system** involves conversion of analog signal to digital signal, then processing of the digital signal by a digital system and then conversion of the processed digital signal back to analog signal.

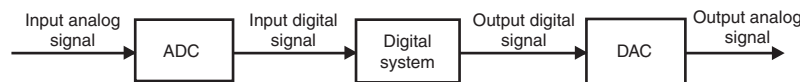


Fig 1.1: Basic components of a DSP system.

The real-world signals are analog, and only for processing by digital systems, the signals are converted to digital. For conversion of signals from analog to digital, an ADC (Analog to Digital Converter) is employed. The various steps in analog to digital conversion process are sampling and quantization of analog signals, and then converting the quantized samples to suitable binary codes. The digital signals in the form of binary codes are fed to digital system for processing, and after processing, it generates an output digital signal in the form of binary codes. The output analog signal is constructed from the output binary codes using a DAC (Digital to Analog Converter).

The processing of signals are basically spectrum analysis to determine the various frequency components of a signal and filtering the signal to extract the required frequency component of the signal.

The **digital system** can be a specially designed programmable hardware for DSP or an algorithm/software running on a general purpose digital system like Personal Computer (PC).

Advantages of Digital Signal Processing

Some of the advantages of digital processing of signals are,

1. The digital hardware are compact, reliable, less expensive and programmable.
2. Since the DSP systems are programmable, the performance of the system can be easily upgraded/modified.

3. By employing high speed, sophisticated digital hardware higher precision can be achieved in processing of signals.
4. The digital signals can be permanently stored in magnetic media so that they are transportable and can be processed in non-real-time or off-line.

1.2 Signal

Any physical phenomenon that conveys or carries some information can be called a **signal**. The music, speech, motion pictures, still photos, heart beat, etc., are examples of signals that we normally encounter in day-to-day life.

When a signal is defined continuously for any value of an independent variable, it is called an **analog** or **continuous signal**. Most of the signals encountered in science and engineering are analog in nature. When the dependent variable of an analog signal is time, it is called a **continuous time signal** and it is denoted as " $x(t)$ ".

When a signal is defined for discrete intervals of an independent variable, it is called a **discrete signal**. When the dependent variable of a discrete signal is time, it is called **discrete time signal** and it is denoted by " $x(n)$ ". Most of the discrete signals are either sampled versions of analog signals for processing by digital systems or output of digital systems.

The quantized and coded version of the discrete time signals are called **digital signals**. In digital signals the value of the signal for every discrete time " n " is represented in binary codes. The process of conversion of a discrete time signal to digital signal involves quantization and coding.

Normally, for binary representation, a standard size of binary is chosen. In m -bit binary representation, we can have 2^m binary codes. The possible range of values of the discrete time signals are usually divided into 2^m steps called **quantization levels**, and a binary code is attached to each quantization level. The values of the discrete time signals are approximated by rounding or truncation in order to match the nearest quantization level.

1.3 Discrete Time System

Any process that exhibits cause and effect relation can be called a **system**. A system will have an input signal and an output signal. The output signal will be a processed version of the input signal. A system is either interconnection of hardware devices or software/algorithm.

A system which can process a discrete time signal is called a **discrete time system**, and so the input and output signals of a discrete time system are discrete time signals.

A discrete time system is denoted by the letter $\mathcal{H}$. The input of discrete time system is denoted as " $x(n)$ " and the output of discrete time system is denoted as " $y(n)$ ". The diagrammatic representation of a discrete time system is shown in Fig 1.2.

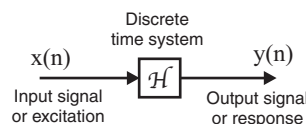


Fig 1.2: Representation of discrete time system.

The operation performed by a discrete time system on input to produce output or response can be expressed as,

$$\text{Response, } y(n) = \mathcal{H}\{x(n)\}$$

where, $\mathcal{H}$ denotes the system operation (also called system operator).

When a discrete time system satisfies the properties of linearity and time invariance then it is called **LTI (Linear Time Invariant) discrete time system**.

The input-output relation of an LTI discrete time system is represented by constant coefficient difference equation shown below:

$$y(n) = - \sum_{m=1}^N a_m y(n-m) + \sum_{m=0}^M b_m x(n-m)$$

where, N = Order of the system, and $M \leq N$.

The solution of the above difference equation is the response $y(n)$ of the discrete time system, for the input $x(n)$.

1.4 Analysis of Discrete Time System

Mostly, the discrete time systems are designed for analysis of discrete time signals. Physically, the discrete time systems are realized in time domain. In time domain, the discrete time systems are governed by difference equations.

The analysis of discrete time signals and systems in time domain involves solution of difference equations. The solution of difference equations are difficult due to assumption of a solution and then solving the constants using initial conditions.

In order to simplify the task of analysis, the discrete time signals can be transformed to some other domain, where the analysis may be easier. One such transform exists for discrete time signals is $\mathcal{Z}$ -transform.

The $\mathcal{Z}$ -transform, will transform a function of discrete time "n" into a function of complex variable "z", where $z = re^{j\omega}$. Therefore, **$\mathcal{Z}$ -transform** of a discrete time signal will transform the time domain signal into z-domain signal.

On taking $\mathcal{Z}$ -transform of the difference equation governing the discrete time system, it becomes algebraic equation in "z" and the solution of algebraic equation will give the response of the system as a function of "z" and it is called z-domain response.

The inverse $\mathcal{Z}$ -transform of the z-domain response, will give the time domain response of the discrete time system. Also, the stability analysis of the discrete systems are much easier in z-domain.

The ratio of $\mathcal{Z}$ -transform of output and input is called **transfer function** of the discrete time system. The inverse $\mathcal{Z}$ -transform of the system gives the **impulse response** of the system, which is used to study the characteristics of a system.

Another important characteristic of any signal is frequency, and for most of the applications the frequency content of the signal is an important criteria. The frequency range of some of the signals are listed in Tables 1.1 and 1.2.

Table 1.1: Frequency Range of some Electromagnetic Signals

Type of signal	Wavelength (m)	Frequency range (Hz)
Radio broadcast	10^4 to 10^2	3×10^4 to 3×10^6
Shortwave radio signals	10^2 to 10^{-2}	3×10^6 to 3×10^{10}
Radar/Space communications	1 to 10^{-2}	3×10^8 to 3×10^{10}
Common-carrier microwave	1 to 10^{-2}	3×10^8 to 3×10^{10}
Infrared	10^{-3} to 10^{-6}	3×10^{11} to 3×10^{14}
Visible light	3.9×10^{-7} to 8.1×10^{-7}	3.7×10^{14} to 7.7×10^{14}
Ultraviolet	10^{-7} to 10^{-8}	3×10^{15} to 3×10^{16}
Gamma rays and x-rays	10^{-9} to 10^{-10}	3×10^{17} to 3×10^{18}

Table 1.2: Frequency Range of some Biological and Seismic Signals

Type of Signal	Frequency Range (Hz)
Electroretinogram	0 to 20
Electronystagmogram	0 to 20
Pneumogram	0 to 40
Electrocardiogram (ECG)	0 to 100
Electroencephalogram (EEG)	0 to 100
Electromyogram	10 to 200
Sphygmomanogram	0 to 200
Speech	100 to 4000
Wind noise	100 to 1000
Seismic exploration signals	10 to 100
Earthquake and nuclear explosion signals	0.01 to 10
Seismic noise	0.1 to 1

The frequency contents of a discrete time signal can be studied by taking Fourier transform of the discrete time signal. The Fourier transform of discrete time signal is a particular class of $\mathbf{Z}$ -transform in which $z = e^{j\omega}$, where " ω " is the frequency of the discrete time signals.

The **Fourier transform**, will transform a function of discrete time " n " into a function of frequency " ω ". Therefore, Fourier transform of a discrete time signal will transform the discrete time signal into frequency domain signal. The Fourier transform of the discrete time signal, is also called **frequency spectrum** of the discrete time signal.

The Fourier transform of the impulse response of a system is called **frequency response** of the system. The frequency spectrum is a complex function of " ω " and so can be expressed as magnitude spectrum and phase spectrum. The magnitude spectrum is used to study the various frequency components of the discrete time signal.

The frequency spectrum obtained via Fourier transform will be a continuous spectrum and so cannot be computed by digital systems. Therefore, the samples of Fourier transform can be computed at sufficient number of points by digital systems. The samples of Fourier transform can also be directly computed using DFT (**D**iscrete **F**ourier **T**ransform).

The computation of DFT involves a large number of calculations. In order to reduce the computational task of DFT, a number of methods/algorithms are developed which are collectively called **FFT** (**F**ast **F**ourier **T**ransform). The DFT of discrete time signal will give the *discrete frequency spectrum* of the signal.

1.5 Filters

The filters are frequency selective devices. The two major types of digital filters are FIR (**F**inite **I**mpulse **R**esponse) and IIR (**I**nfinite **I**mpulse **R**esponse) filters.

Generally, the filter specification will be a desired frequency response. The inverse Fourier transform of the frequency response will be the impulse response of the filter, and it will be an infinite duration signal. The digital filters designed by choosing finite samples of impulse response are called **FIR filters**, and the filters designed by considering all the infinite samples are called **IIR filters**.

Since, an FIR filter is designed from the finite samples of impulse response, the direct design of FIR filter is possible in which the transfer function of the filter is obtained by taking $\mathcal{Z}$ -transform of impulse response.

Note: Mathematically, the filter design is design of transfer function of the filter.

Since, an IIR filter is designed by considering/preserving the infinite samples of impulse response, the direct design of IIR filter is not possible. Therefore, the IIR filter is designed via analog filter.

For designing IIR filter, first the specifications of IIR filter is transformed to specifications of analog filter using bilinear or impulse-invariant transformation, then an analog filter transfer function is designed using Butterworth or Chebychev approximation. Finally the analog filter transfer function is transferred to digital filter transfer function using the transformation chosen for transforming the specifications.

1.6 Finite Word Length Effects

In digital representation the signals are represented as an array of binary numbers, and the digital system employ a fixed size of binary called "word size or word length" for number representation. This finite word size for number representation leads to errors in input signals, intermediate signals in computations and in the final output signals.

In general, the various effects due to finite precision representation of numbers in digital systems are called *finite word length effects*.

Some of the finite word length effects in digital systems are given below.

- Errors due to quantization of input data.
 - Errors due to quantization of filter coefficients.
 - Errors due to rounding the product in multiplication.
 - Errors due to overflow in addition.
 - Limit cycles in recursive computations.
-

1.7 Multirate DSP

In many communication systems, the sampling rate conversion is a vital requirement. Some of the systems that employ sampling rate conversion are video receivers that receive both NTSC and PAL signals, audio systems that can play CDs recorded in different standards, etc.

The processing of discrete time signals at different sampling rates in different parts of a system is called ***multirate DSP***. In digital systems, the sampling rate conversion is achieved by either decimation or interpolation. In decimation, the sampling rate is reduced, whereas in interpolation the sampling rate is increased. The multirate DSP systems leads to reduction in computations, memory requirement and errors due to finite word length effects.

1.8 Energy and Power Spectrum

There are many situations where the signals are corrupted by noise like sonar signals corrupted by ambient ocean noise, speech signal from cockpit of an airplane corrupted by engine noise, etc. When the signals are corrupted by noise, then the energy or power spectrum will be useful to identify the signal from noise.

The ***energy spectrum*** can be computed for deterministic signals, and it is given by square of magnitude of Fourier transform of the signal. Alternatively, the energy spectrum is given by Fourier transform of the autocorrelation sequence of the signal.

The power spectrum can be estimated for nondeterministic signals or random process/signals. The power spectrum estimation methods can be broadly classified into two groups, namely, nonparametric methods and parametric methods.

In ***nonparametric methods***, first an estimate of autocorrelation of the random process is determined which represents the average behaviour of the signal, then the Fourier transform of estimated autocorrelation is determined, which is the power spectrum estimate of the random process.

In ***parametric methods***, first an appropriate model is selected for the given random process, then the parameters of the model are computed using the available data of the random process. Finally, the power spectrum is estimated from the constructed model.

1.9 Digital Signal Processors

The ***digital signal processors*** are specially designed microprocessors/microcontrollers for DSP applications.

The importance of special purpose processors for signal processing applications were realised in 1980s, and many companies started releasing special processors for DSP applications. The pioneers among them are Texas Instruments and Analog Devices. The Texas Instruments has released a large variety of processors in the family name TMS320Cxx and Analog Devices has released processors in the family name ADSPxx.

Some of the special features of digital signal processors are given below.

- Modified Harvard architecture with two or more internal buses for simultaneous access of code and one or two data.
- Specialized addressing modes like circular addressing and bit reversed addressing suitable for computations like convolution, correlation and FFT.

- MAC unit for performing multiply-accumulate computations involved in convolution, correlation and FFT in single clock cycle.
- Larger size ALU and accumulators with guard bits to accommodate the overflow in computation.
- Pipelining of instructions to execute different phases of four or six instructions in parallel.
- VLIW architecture to fetch and execute multiple instructions in parallel.
- Multiprocessor architecture by integrating multiple processors on a single piece of silicon for parallel processing.

1.10 Importance of Digital Signal Processing

The technology advancement in programmable digital signal processors, helps to implement more and more real time applications in digital systems.

The digital processing of signal plays a vital role in almost every field of Science and Engineering. Some of the applications of digital processing of signals in various field of Science and Engineering are listed here.

1. Biomedical

- ECG is used to predict heart diseases.
- EEG is used to study normal and abnormal behaviour of the brain.
- EMG is used to study the condition of muscles.
- X-ray images are used to predict the bone fractures and tuberculosis.
- Ultrasonic scan images of kidney and gall bladder is used to predict stones.
- Ultrasonic scan images of foetus is used to predict abnormalities in a baby.
- MRI scan is used to study minute inner details of any part of the human body.

2. Speech Processing

- Speech compression and decompression to reduce memory requirement of storage systems.
- Speech compression and decompression for effective use of transmission channels.
- Speech recognition for voice operated systems and voice based security systems.
- Speech recognition for conversion of voice to text.
- Speech synthesis for various voice based warnings or announcements.

3. Audio and Video Equipments

- The analysis of audio signals will be useful to design systems for special effects in audio systems like stereo, woofer, karaoke, equalizer, attenuator, etc.
- Music synthesis and composing using music keyboards.
- Audio and video compression for storage in DVDs.

4. Communication

- The spectrum analysis of modulated signals helps to identify the information bearing frequency component that can be used for transmission.
- The analysis of signals received from radars are used to detect flying objects and thier velocity.
- Generation and detection of DTMF signals in telephones.
- Echo and noise cancellation in transmission channels.

5. Power electronics

- The spectrum analysis of the output of converters and inverters will reveal the harmonics present in the output, which in turn helps to design suitable filter to eliminate the harmonics.
- The analysis of switching currents and voltages in power devices will help to reduce losses.

6. Image processing

- Image compression and decompression to reduce memory requirement of storage systems.
- Image compression and decompression for effective use of transmission channels.
- Image recognition for security systems.
- Filtering operations on images to extract the features or hidden information.

7. Geology

- The seismic signals are used to determine the magnitude of earthquakes and volcanic eruptions.
- The seismic signals are also used to predict nuclear explosions.
- The seismic noises are also used to predict the movement of earth layers (tectonic plates).

8. Astronomy

- The analysis of light received from a star is used to determine the condition of the star.
 - The analysis of images of various celestial bodies gives vital information about them.
-

1.11 Use of MATLAB in Digital Signal Processing

MATLAB (**MATrix LABoratory**) is a software developed by The MathWork Inc, USA, which can run on any windows platform in a PC (**P**ersonal **C**omputer). This software has a number of tools for the study of various engineering subjects. It includes various tools for digital signal processing also. Using these tools, a wide variety of studies can be made on discrete time signals and systems. Some of the analysis that is relevant to this particular textbook are given below.

- Sketch or plot of discrete time signals as a function of independent variable.
 - Spectrum analysis of discrete time signals.
 - Solution of LTI discrete time systems.
 - Perform convolution and deconvolution operations on discrete time signals.
 - Perform various transforms on discrete time signals like Fourier transform, **Z**-transform, **F**ast **F**ourier **T**ransform (FFT), etc.
 - Design and frequency response analysis of FIR and IIR filters.
-