## Contents

**2.2 System Concepts**
- **2.2.1 LTI Systems**
- **2.2.2 The z-transform**
- **2.2.3 Transfer Functions**
- **2.2.4 Poles and Zeros**
- **2.2.5 Frequency Responses**
- **2.2.6 Discrete Fourier Transform**

**2.3 Introduction to Random Variables**
- **2.3.1 Review of Random Variables**
- **2.3.2 Operations of Random Variables**

**2.4 Fixed-Point Representations and Quantization Effects**
- **2.4.1 Fixed-Point Formats**
- **2.4.2 Quantization Errors**
- **2.4.3 Signal Quantization**
- **2.4.4 Coefficient Quantization**
- **2.4.5 Roundoff Noise**
- **2.4.6 Fixed-Point Toolbox**

**2.5 Overflow and Solutions**
- **2.5.1 Saturation Arithmetic**
- **2.5.2 Overflow Handling**
- **2.5.3 Scaling of Signals**
- **2.5.4 Guard Bits**

**2.6 Experiments and Program Examples**
- **2.6.1 Overflow and Saturation Arithmetic**
- **2.6.2 Function Approximations**
- **2.6.3 Real-Time Signal Generation Using eZdsp**

**Exercises**

**References**

---

**3 Design and Implementation of FIR Filters**

**3.1 Introduction to FIR Filters**
- **3.1.1 Filter Characteristics**
- **3.1.2 Filter Types**
- **3.1.3 Filter Specifications**
- **3.1.4 Linear Phase FIR Filters**
- **3.1.5 Realization of FIR Filters**

**3.2 Design of FIR Filters**
- **3.2.1 Fourier Series Method**
- **3.2.2 Gibbs Phenomenon**
- **3.2.3 Window Functions**
- **3.2.4 Design of FIR Filters Using MATLAB®**
- **3.2.5 Design of FIR Filters Using the FDATool**

**3.3 Implementation Considerations**
- **3.3.1 Quantization Effects in FIR Filters**
- **3.3.2 MATLAB® Implementations**
- **3.3.3 Floating-Point C Implementations**
- **3.3.4 Fixed-Point C Implementations**
3.4 Applications: Interpolation and Decimation Filters 130
3.4.1 Interpolation 130
3.4.2 Decimation 131
3.4.3 Sampling Rate Conversion 133
3.4.4 MATLAB® Implementations 134
3.5 Experiments and Program Examples 135
3.5.1 FIR Filtering Using Fixed-Point C 135
3.5.2 FIR Filtering Using C55xx Assembly Program 136
3.5.3 Symmetric FIR Filtering Using C55xx Assembly Program 137
3.5.4 Optimization Using Dual-MAC Architecture 138
3.5.5 Real-Time FIR Filtering 140
3.5.6 Decimation Using C and Assembly Programs 141
3.5.7 Interpolation Using Fixed-Point C 142
3.5.8 Sampling Rate Conversion 142
3.5.9 Real-Time Sampling Rate Conversion 143
Exercises 144
References 147

4 Design and Implementation of IIR Filters 148
4.1 Introduction 148
4.1.1 Analog Systems 148
4.1.2 Mapping Properties 150
4.1.3 Characteristics of Analog Filters 151
4.1.4 Frequency Transforms 153
4.2 Design of IIR Filters 154
4.2.1 Bilinear Transform 155
4.2.2 Filter Design Using the Bilinear Transform 156
4.3 Realization of IIR Filters 158
4.3.1 Direct Forms 158
4.3.2 Cascade Realizations 160
4.3.3 Parallel Realizations 161
4.3.4 Realization of IIR Filters Using MATLAB® 162
4.4 Design of IIR Filters Using MATLAB® 164
4.4.1 Filter Design Using MATLAB® 164
4.4.2 Frequency Transforms Using MATLAB® 166
4.4.3 Filter Design and Realization Using the FDATool 166
4.5 Implementation Considerations 168
4.5.1 Stability 168
4.5.2 Finite-Precision Effects and Solutions 170
4.5.3 MATLAB® Implementations of IIR Filters 172
4.6 Practical Applications 174
4.6.1 Recursive Resonators 174
4.6.2 Recursive Quadrature Oscillators 177
4.6.3 Parametric Equalizers 179
4.7 Experiments and Program Examples 179
4.7.1 Direct-Form I IIR Filter Using Floating-Point C 179
4.7.2 Direct-Form I IIR Filter Using Fixed-Point C 181
4.7.3 Cascade IIR Filter Using Fixed-Point C 182
4.7.4 Cascade IIR Filter Using Intrinsics 185
4.7.5 Cascade IIR Filter Using Assembly Program 188
4.7.6 Real-Time IIR Filtering 189
4.7.7 Parametric Equalizer Using Fixed-Point C 190
4.7.8 Real-Time Parametric Equalizer 190

Exercises 191

References 194

## 5 Frequency Analysis and the Discrete Fourier Transform 195

5.1 Fourier Series and Fourier Transform 195
   5.1.1 Fourier Series 195
   5.1.2 Fourier Transform 197

5.2 Discrete Fourier Transform 198
   5.2.1 Discrete-Time Fourier Transform 198
   5.2.2 Discrete Fourier Transform 200
   5.2.3 Important Properties 202

5.3 Fast Fourier Transforms 205
   5.3.1 Decimation-in-Time 206
   5.3.2 Decimation-in-Frequency 208
   5.3.3 Inverse Fast Fourier Transform 209

5.4 Implementation Considerations 210
   5.4.1 Computational Issues 210
   5.4.2 Finite-Precision Effects 210
   5.4.3 MATLAB® Implementations 211
   5.4.4 Fixed-Point Implementation Using MATLAB® 212

5.5 Practical Applications 214
   5.5.1 Spectral Analysis 214
   5.5.2 Spectral Leakage and Resolution 215
   5.5.3 Power Spectral Density 219
   5.5.4 Convolution 222

5.6 Experiments and Program Examples 224
   5.6.1 DFT Using Floating-Point C 224
   5.6.2 DFT Using the C55xx Assembly Program 226
   5.6.3 FFT Using Floating-Point C 227
   5.6.4 FFT Using Fixed-Point C with Intrinsics 227
   5.6.5 Experiment with the FFT and IFFT 231
   5.6.6 FFT Using the C55xx Hardware Accelerator 231
   5.6.7 Real-Time FFT Using the C55xx Hardware Accelerator 233
   5.6.8 Fast Convolution Using the Overlap-Add Technique 234
   5.6.9 Real-Time Fast Convolution 235

Exercises 236

References 238
6 Adaptive Filtering

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Introduction to Random Processes</td>
<td>239</td>
</tr>
<tr>
<td>6.2 Adaptive Filters</td>
<td>243</td>
</tr>
<tr>
<td>6.2.1 Introduction to Adaptive Filtering</td>
<td>243</td>
</tr>
<tr>
<td>6.2.2 Performance Function</td>
<td>244</td>
</tr>
<tr>
<td>6.2.3 Method of Steepest Descent</td>
<td>248</td>
</tr>
<tr>
<td>6.2.4 LMS Algorithm</td>
<td>249</td>
</tr>
<tr>
<td>6.2.5 Modified LMS Algorithms</td>
<td>251</td>
</tr>
<tr>
<td>6.3 Performance Analysis</td>
<td>252</td>
</tr>
<tr>
<td>6.3.1 Stability Constraint</td>
<td>252</td>
</tr>
<tr>
<td>6.3.2 Convergence Speed</td>
<td>253</td>
</tr>
<tr>
<td>6.3.3 Excess Mean-Square Error</td>
<td>254</td>
</tr>
<tr>
<td>6.3.4 Normalized LMS Algorithm</td>
<td>254</td>
</tr>
<tr>
<td>6.4 Implementation Considerations</td>
<td>255</td>
</tr>
<tr>
<td>6.4.1 Computational Issues</td>
<td>255</td>
</tr>
<tr>
<td>6.4.2 Finite-Precision Effects</td>
<td>256</td>
</tr>
<tr>
<td>6.4.3 MATLAB® Implementations</td>
<td>257</td>
</tr>
<tr>
<td>6.5 Practical Applications</td>
<td>259</td>
</tr>
<tr>
<td>6.5.1 Adaptive System Identification</td>
<td>259</td>
</tr>
<tr>
<td>6.5.2 Adaptive Prediction</td>
<td>262</td>
</tr>
<tr>
<td>6.5.3 Adaptive Noise Cancellation</td>
<td>264</td>
</tr>
<tr>
<td>6.5.4 Adaptive Inverse Modeling</td>
<td>267</td>
</tr>
<tr>
<td>6.6 Experiments and Program Examples</td>
<td>268</td>
</tr>
<tr>
<td>6.6.1 LMS Algorithm Using Floating-Point C</td>
<td>268</td>
</tr>
<tr>
<td>6.6.2 Leaky LMS Algorithm Using Fixed-Point C</td>
<td>270</td>
</tr>
<tr>
<td>6.6.3 Normalized LMS Algorithm Using Fixed-Point C and Intrinsics</td>
<td>270</td>
</tr>
<tr>
<td>6.6.4 Delayed LMS Algorithm Using Assembly Program</td>
<td>274</td>
</tr>
<tr>
<td>6.6.5 Experiment of Adaptive System Identification</td>
<td>275</td>
</tr>
<tr>
<td>6.6.6 Experiment of Adaptive Predictor</td>
<td>276</td>
</tr>
<tr>
<td>6.6.7 Experiment of Adaptive Channel Equalizer</td>
<td>277</td>
</tr>
<tr>
<td>6.6.8 Real-Time Adaptive Prediction Using eZdsp</td>
<td>279</td>
</tr>
<tr>
<td>Exercises</td>
<td>280</td>
</tr>
<tr>
<td>References</td>
<td>282</td>
</tr>
</tbody>
</table>

7 Digital Signal Generation and Detection

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1 Sine Wave Generators</td>
<td>283</td>
</tr>
<tr>
<td>7.1.1 Lookup Table Method</td>
<td>283</td>
</tr>
<tr>
<td>7.1.2 Linear Chirp Signal</td>
<td>286</td>
</tr>
<tr>
<td>7.2 Noise Generators</td>
<td>288</td>
</tr>
<tr>
<td>7.2.1 Linear Congruential Sequence Generator</td>
<td>288</td>
</tr>
<tr>
<td>7.2.2 Pseudo-random Binary Sequence Generator</td>
<td>289</td>
</tr>
<tr>
<td>7.2.3 White, Color, and Gaussian Noise</td>
<td>290</td>
</tr>
<tr>
<td>7.3 DTMF Generation and Detection</td>
<td>291</td>
</tr>
<tr>
<td>7.3.1 DTMF Generator</td>
<td>291</td>
</tr>
<tr>
<td>7.3.2 DTMF Detection</td>
<td>292</td>
</tr>
</tbody>
</table>
9.2.2 Short-Time Spectrum Estimation
9.2.3 Magnitude Spectrum Subtraction
9.3 VoIP Applications
  9.3.1 Overview of VoIP
  9.3.2 Discontinuous Transmission
  9.3.3 Packet Loss Concealment
  9.3.4 Quality Factors of Media Stream
9.4 Experiments and Program Examples
  9.4.1 LPC Filter Using Fixed-Point C with Intrinsics
  9.4.2 Perceptual Weighting Filter Using Fixed-Point C with Intrinsics
  9.4.3 VAD Using Floating-Point C
  9.4.4 VAD Using Fixed-Point C
  9.4.5 Speech Encoder with Discontinuous Transmission
  9.4.6 Speech Decoder with CNG
  9.4.7 Spectral Subtraction Using Floating-Point C
  9.4.8 G.722.2 Using Fixed-Point C
  9.4.9 G.711 Companding Using Fixed-Point C
  9.4.10 Real-Time G.711 Audio Loopback
Exercises
References

10 Audio Signal Processing
  10.1 Introduction
  10.2 Audio Coding
    10.2.1 Basic Principles
    10.2.2 Frequency-Domain Coding
    10.2.3 Lossless Audio Coding
    10.2.4 Overview of MP3
  10.3 Audio Equalizers
    10.3.1 Graphic Equalizers
    10.3.2 Parametric Equalizers
  10.4 Audio Effects
    10.4.1 Sound Reverberation
    10.4.2 Time Stretch and Pitch Shift
    10.4.3 Modulated and Mixed Sounds
    10.4.4 Spatial Sounds
  10.5 Experiments and Program Examples
    10.5.1 MDCT Using Floating-Point C
    10.5.2 MDCT Using Fixed-Point C and Intrinsics
    10.5.3 Pre-echo Effects
    10.5.4 MP3 Decoding Using Floating-Point C
    10.5.5 Real-Time Parametric Equalizer Using eZdsp
    10.5.6 Flanger Effects
    10.5.7 Real-Time Flanger Effects Using eZdsp
    10.5.8 Tremolo Effects
    10.5.9 Real-Time Tremolo Effects Using eZdsp
10.5.10 Spatial Sound Effects
10.5.11 Real-Time Spatial Effects Using eZdsp

Exercises

References

11 Introduction to Digital Image Processing

11.1 Digital Images and Systems
   11.1.1 Digital Images
   11.1.2 Digital Image Systems

11.2 Color Spaces
11.3 YCgCr Sub-sampled Color Space
11.4 Color Balance and Correction
   11.4.1 Color Balance
   11.4.2 Color Correction
   11.4.3 Gamma Correction

11.5 Histogram Equalization
11.6 Image Filtering
11.7 Fast Convolution
11.8 Practical Applications
   11.8.1 DCT and JPEG
   11.8.2 Two-Dimensional DCT
   11.8.3 Fingerprint
   11.8.4 Discrete Wavelet Transform

11.9 Experiments and Program Examples
   11.9.1 YCgCr to RGB Conversion
   11.9.2 White Balance
   11.9.3 Gamma Correction and Contrast Adjustment
   11.9.4 Image Filtering
   11.9.5 DCT and IDCT
   11.9.6 Image Processing for Fingerprints
   11.9.7 The 2-D Wavelet Transform

Exercises

References

Appendix A Some Useful Formulas and Definitions

A.1 Trigonometric Identities
A.2 Geometric Series
A.3 Complex Variables
A.4 Units of Power
References

Appendix B Software Organization and List of Experiments

Appendix C Introduction to the TMS320C55xx Digital Signal Processor

C.1 Introduction
C.2 TMS320C55xx Architecture
C.2.1 Architecture Overview 490
C.2.2 On-Chip Memories 494
C.2.3 Memory-Mapped Registers 495
C.2.4 Interrupts and Interrupt Vector 498
C.3 TMS320C55xx Addressing Modes 498
  C.3.1 Direct Addressing Modes 501
  C.3.2 Indirect Addressing Modes 502
  C.3.3 Absolute Addressing Modes 505
  C.3.4 MMR Addressing Mode 505
  C.3.5 Register Bits Addressing Mode 506
  C.3.6 Circular Addressing Mode 507
C.4 TMS320C55xx Assembly Language Programming 508
  C.4.1 Arithmetic Instructions 508
  C.4.2 Logic and Bit Manipulation Instructions 509
  C.4.3 Move Instruction 509
  C.4.4 Program Flow Control Instructions 510
  C.4.5 Parallel Execution 514
  C.4.6 Assembly Directives 516
  C.4.7 Assembly Statement Syntax 518
C.5 C Programming for TMS320C55xx 520
  C.5.1 Data Types 520
  C.5.2 Assembly Code Generation by C Compiler 520
  C.5.3 Compiler Keywords andPragma Directives 522
C.6 Mixed C and Assembly Programming 525
C.7 Experiments and Program Examples 529
  C.7.1 Examples 529
  C.7.2 Assembly Program 530
  C.7.3 Multiplication 530
  C.7.4 Loops 531
  C.7.5 Modulo Operator 532
  C.7.6 Use Mixed C and Assembly Programs 533
  C.7.7 Working with AIC3204 533
  C.7.8 Analog Input and Output 534
References 535

Index 537