





This is to certify that the

thesis entitled

INTRODUCTORY BIOMEDICAL DIGITAL SIGNAL PROCESSING: A SURVEY AND TEXT OUTLINE

presented by

Dale L. Grover

has been accepted towards fulfillment of the requirements for

<u>Masters</u> degree in <u>Electrical</u> Eng

John R. Well Major professor Date 13 May 99

O-7639

MSU is an Affirmative Action/Equal Opportunity Institution

LIBRARY Michigan State University

PLACE IN RETURN BOX to remove this checkout from your record. TO AVOID FINES return on or before date due. MAY BE RECALLED with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE

1/98 c:/CIRC/DateDue.p65-p.14

INTRODUCTORY BIOMEDICAL DIGITAL SIGNAL PROCESSING: A SURVEY AND TEXT OUTLINE

By

Dale L. Grover

A THESIS

Submitted to Michigan State University in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

ABSTRACT

INTRODUCTORY BIOMEDICAL DIGITAL SIGNAL PROCESSING: A SURVEY AND TEXT OUTLINE

By

Dale L. Grover

Digital signal processing (DSP) is a vital tool in biomedical engineering. In contrast to most electrical engineering students, undergraduate students of biomedical engineering learn DSP in diverse contexts, as revealed in a brief survey of biomedical engineering classes. Students also come to these classes with diverse backgrounds that often do not include linear signals and systems and other traditional DSP prerequisites. In addition to context and student background, the content of biomedical DSP courses vary from a traditional DSP course. This thesis argues that a course can present practical fundamental DSP concepts to biomedical engineering students without requiring traditional prerequisites, and in a manner that addresses some of the unique aspects of biomedical DSP. To illustrate one approach, an outline for a text is presented that might be helpful in accomplishing this goal.

TABLE OF CONTENTS

	LIST OF TABLES	iv
	LIST OF FIGURES	v
1.1 1.2 1.3 1.4	INTRODUCTION The Importance of DSP in Biomedical Engineering What Do Students of Biomedical Engineering Need to Know About DSP? An Earlier Approach to DSP with a Similar Audience Thesis	1 1 1 2 3
2 2.1 2.2 2.3 2.4	Overview of Biomedical DSP Courses. Purpose of Survey. Method	3 3 4 17
3 3.1 3.2	DSP for "Non-EE" Biomedical Engineering Students Biomedical DSP vs. "Traditional" DSP Useful Pedagogical Themes	17 17 20
4 4.1 4.2 4.3 4.4 4.5 4.6	Outline for a Proposed Text	23 23 24 25 28 36 37
	SUMMARY AND CONCLUSIONS	39
	APPENDICES	40
	APPENDIX A: OUTLINE FOR A TEXT	41
	APPENDIX B: EXAMPLE EXERCISES	56
	BIBLIOGRAPHY	69

LIST OF TABLES

Table 1	Summary of Courses	4
Table 2	Course Descriptions.	9
Table 3	Exercise 8	64
Table 4	Exercise 9	65
Table 5	Exercise 10	66
Table 6	Exercise 15	67

LIST OF FIGURES

v

Figure 1 Exercise 1, 2 cycles of a cosine wave.	. 56
Figure 2 Exercise 1, signals a) through d)	. 59
Figure 3 Exercise 1, signals e) through h)	. 60
Figure 4 Exercise 2, FFT of a square wave	. 61
Figure 5 Exercise 6, original signal	. 62
Figure 6 Exercise 6, signal with 60 Hz noise	. 63
Figure 7 Exercise 6, filtered and original signals	. 63

INTRODUCTION

1.1 The Importance of DSP in Biomedical Engineering

Digital Signal Processing (DSP) is a set of techniques for processing signals using mathematical algorithms implemented with digital hardware. Combined with analog-todigital and digital-to-analog converters, these techniques allow a wide variety of realworld (analog) signals to be processed in ways that are not possible using analog electronics. DSP techniques include operations such as filtering, correlation, and transformation between time and frequency domains.

DSP is an increasingly critical component of modern biomedical engineering (BME)—a large part of BME simply would not exist without these techniques. For example, computer-based restoration, enhancement, coding, and detection in medical imaging would not be possible without DSP. In other cases, DSP provides significant advantages over older technologies for processing biomedical signals; the ECG predates widespread DSP, but is now processed using techniques that would not have been practical prior to widespread availability of DSP hardware.

1.2 What Do Students of Biomedical Engineering Need to Know About DSP?

The practice of BME involves topics as varied as thermodynamics, chemistry, mechanical engineering, robotics, material science, and electrical engineering (e.g., control, signal processing, neural networks). Many biomedical engineers are not directly concerned with signal processing theory—in these cases, DSP may be no more than a tool. For others, it may be sufficient to be as acquainted with DSP techniques as an embedded systems programmer is with thermodynamics—a familiarity is desirable and the basic concepts are about all that will fit into the curriculum. At the other extreme, biomedical engineers who implement DSP need a deep understanding on par with signal processing engineers from any specialty.

How well students from BME programs are prepared for studying DSP is another issue. A number of subjects come together in the study of DSP, including:

- Signals: Linear systems and signals, control theory
- Mathematics: Algebra, complex numbers, trigonometry, calculus
- Computer Science: Programming
- Electronics: Circuit analysis, microprocessor programming, architecture.

These subjects are largely familiar to any undergraduate electrical engineering (EE) student, but some elements, particularly in the signals and computer architecture areas, are less likely to be treated in undergraduate curricula, for example, in mechanical or chemical engineering—both possible areas of emphasis within BME.

1.3 An Earlier Approach to DSP with a Similar Audience

This thesis was motivated in part by the desire to extend the experience and insights from a recently-completed writing project to the field of BME DSP. *Digital Signal Processing and the Microcontroller* [7] was co-written by the author of this thesis for embedded systems programmers who require basic DSP skills, but who typically do not have an EE background.

The starting point for this project was the hypothesis that the practical emphasis and limited prerequisites of [7] would parallel those of many BME students, and so the content and approach, with modification, would be useful in a BME context. The relevant characteristics of the intended audience of [7] and those of most BME students are not identical in type or degree; for example in the realm of background knowledge, embedded systems programmers typically have an excellent knowledge of programming and microprocessor architecture, which is not generally true of BME students. However, since the majority of [7] is about basic DSP topics rather than implementation-specific issues, it was plausible that most of the content (and often, the approaches taken in discussing this content) could be applied to creating a text for BME DSP. Chapter 4 will discuss the differences between [7] and the outline of a BME DSP text presented in this paper.

1.4 Thesis

An understanding of basic DSP techniques is important to many, if not most, BME students, not all of whom have a sufficient background to take a "traditional," EE-based DSP course. This paper makes the case that a course/text can present practical fundamental DSP concepts to BME students without requiring the conventional prerequisites of an EE-based DSP course. First, a national review of selected BME courses that involve DSP will provide a context. Next, based on this review and discussions with the instructors of these courses, a particular audience for, and characteristics of, a course will be outlined. Finally, this information, along with material and approaches from [7], will be used as the basis for an outline and supporting material for a text addressing the needs of the non-EE BME student.

2 Overview of Biomedical DSP Courses

2.1 Purpose of Survey

This section presents the results of a survey of BME courses involving DSP. The purpose of this survey is to provide a broad picture of the context, content, student background, and methods of current BME courses that include DSP topics. The main intention of this survey was to gauge the diversity of such courses and determine whether there currently exist introductory DSP courses directed at BME students with a limited EE background. A comprehensive survey of the field, while not the objective here, is worthy of future research.

2.2 Method

A list of U.S. colleges and universities that offer coursework in biomedical DSP was created from a list of schools offering BME courses [1] and searches on the Internet. Specific courses were then identified, the main criterion being that introductory DSP topics were presented (roughly encompassing sampling theory through simple digital filters) in a biomedical context. Information for each course was compiled from course web pages, course descriptions, and phone and e-mail contact with instructors. During this sur-

vey of existing courses, instructors were often willing to discuss "philosophical" issues surrounding DSP in BME, such as pedagogical strategies, value and role of DSP in a biomedical curriculum, etc. These comments have been incorporated into the discussion in Chapter 4.

2.3 Survey Results

2.3.1 Summary of Courses

Table 1 summarizes some aspects of the context, content, and methods of 18 BME courses involving DSP from 14 colleges and universities. (Table 2 provides a brief description of each class.) Each of these aspects is discussed in detail below, but a few general comments should be made. First, the information available about each course varied greatly, from detailed course syllabi to brief course descriptions. Thus, this table only indicates, for example, the positive inclusion of particular topics, not their exclusion. In other cases prerequisites, content, and/or methods are implied by descriptions; a capital I marks these cases.

	UCSD BE 122B	CWR EBME 308	Duke BME164	Duke BME171	H/MIT HST582J	U Iowa 51:180	U Miami BME 614	U Miami BME 622	U Mich. BME 458	MSOE BE 433	NW C20	U Penn. BE 301	U Penn. BE 465	U Pitt. BIOENG 1580	U Pitt. BIOENG 2510	P/IU "Biosignal Proc."	Vanderbilt BME 263	Wright 466-3
<u>G</u> rad/ <u>U</u> nder- grad/ <u>B</u> oth	U	U	U	U	G	G	G	G	U	U	U	U	U	B	G	G	U	U
Capstone			С															
Prerequisites ¹ (BME/EE)		I2				Ι		I	I	I					I		Ι	
Calculus / Dif- ferential Equa- tions	E			E	e					I	E	E	e					
Linear Algebra	E				e													
Probability																e		E

Table 1Summary of Courses

-

TT

														0	0	= ,		
	UCSD BE 122B	CWR EBME 308	Duke BME164	Duke BME171	H/MIT HST582J	U Iowa 51:180	U Miami BME 614	U Miami BME 622	U Mich. BME 458	MSOE BE 433	NW C20	U Penn. BE 301	U Penn. BE 465	U Pitt. BIOENG 158	U Pitt. BIOENG 251	P/IU "Biosignal Proc	Vanderbilt BME 263	Wright 466-3
Physics											E				1			
Biomed. (misc.)											-	B	B					
Programming							E								1	e		
Circuit analysis			В	В						Ι								
Control	В																	
Linear Signals & Systems	b				E				-	Ι			e	E		e		E
Content: Biomedical		-		-			-	-		-	1			-	-			
Biosignals			X				-		X	12	X	X	12	X				X
Electrochemis- try									X							R	1	X
Bioinstrumen- tation			X			X			X				X					
Medical Image Technology			X		X											1	2	
Safety				\top	1	1	X	-	1	-	-	-	X	1		1	-	-
Content: Electronics, Computer																		
Analog circuits Hardware			X										X				-	
A/D, D/A Hard- ware			X		X		X											
Microprocessor Hardware							X											
Real-Time Pro- gramming						X	X	X										
Telemetry		1	X									1		1			1	1
Content: Continuous- Time SP																		1
Linear Signals & Systems	X	X		X		X					X	X			X			

	UCSD BE 122B	CWR EBME 308	Duke BME164	Duke BME171	H/MIT HST582J	U Iowa 51:180	U Miami BME 614	U Miami BME 622	U Mich. BME 458	MSOE BE 433	NW C20	U Penn. BE 301	U Penn. BE 465	U Pitt. BIOENG 1580	U Pitt. BIOENG 2510	P/IU "Biosignal Proc."	Vanderbilt BME 263	Wright 466-3
Fourier Series, Transform		X	X	X							X				X	1		-
Laplace Trans- form		X		X											X			
Frequency Re- sponse Function		X									X	X			X			
Convolution				X											X			
Correlation															X			
Content: Discrete-Time SP	X	X		X					X	X	X					X	-	X
DSP (unspeci- fied)						X	X		X	X			X		X			X
DFT/FFT	Х							X		X	X	X		X	X	X	X	X
Z Transform										X					X	X		
Convolution				X														
Correlation			X							X	X			X		X	X	
Digital Filters, (general)	X	X	X		X			X		X	X	X	X	X	X	X		
Digital Filter Design		X			X					X	X				X			
Modulation											X							
Signal Averag- ing			X					X	X	X						X		
Feature Extrac- tion					X									X	X	X		
Detection/ Estimation/ Classification								X								x		
Time/Frequen- cy Analysis	X										X				X			
Stochastic Pro- cessing															X		X	X
Adaptive Filters		1	1	1			-									X		
Modeling																X	X	

	UCSD BE 122B	CWR EBME 308	Duke BME164	Duke BME171	H/MIT HST582J	U Iowa 51:180	U Miami BME 614	U Miami BME 622	U Mich. BME 458	MSOE BE 433	NW C20	U Penn. BE 301	U Penn. BE 465	U Pitt. BIOENG 1580	U Pitt. BIOENG 2510	P/IU "Biosignal Proc."	Vanderbilt BME 263	Wright 466-3
Simulation				X	-			_			-		_	-				_
Compression											-				1	X		
Image Process- ing	x				X						x			-	x			-
Nonlinear Sig- nals and Sys- tems					X										X		-	-
Control		-				_	-	-	-	-	_	-		-	X			
Homework/ Laboratories		I		-		I	I	I	-	-		I		5	1.0	1010	I	I
MATLAB	X		X	X	X	-				X	X		1	X	X	X	1	1
LabView	-					-			X				-			X		1
C/C++	-					1				X					1	197	1.00	100
Real Data	-	X	X		X					X				X	1			1
Laboratories	+	X	X		X				X	X	X	-	Ι			X	-	1
Real-Time Pro- cessing			X															
Bioinstrumen- tation			X						X									
Program/Build Hardware			X						X	X			X					
Signal Acquisi- tion			X						X									
Projects	X		X						X					X		X		-
Signals Dis- cussed/Used	I	I		I	-	I	I	-			+		+		-	+	I	I
Cardiology			X		X			X	X	X		X	X	X		X		
Neurophysiolo- gy					X			X		X		X		X	X	X		
Respiratory								X										T

12 1 17 170 #	UCSD BE 122B	CWR EBME 308	Duke BME164	Duke BME171	H/MIT HST582J	U Iowa 51:180	U Miami BME 614	U Miami BME 622	U Mich. BME 458	MSOE BE 433	NW C20	U Penn. BE 301	U Penn. BE 465	U Pitt. BIOENG 1580	U Pitt. BIOENG 2510	P/IU "Biosignal Proc."	Vanderbilt BME 263	Wright 466-3
Motor								X										
Blood Pressure								1	X					100	X	10	10	1
Ultrasound		1	1	1	1	1	-	1	X								1	1
Speech					X													11-
Imaging			X		X					1	X	1					100	

Table Notes:

 In the area of prerequisites, instructors sometimes indicated expectations with respect to students' backgrounds; these are indicated by a lower case letter, while upper case indicates an explicit prerequisite. Also recorded are whether prerequisites are within the biomedical curriculum ("B" and "b") or are typically taken in another department, thence "B" and "e").

2. Items that are inferred from course descriptions or other sources are indicated by a capital "L."

Institution	Course, WWW page ¹	Description
University of Califor- nia, San Diego	122B: Biomedical Electronics (http://www- dur.ucsd.edu/general- catalog/Bioengin/ html/ gen_cat_courses.ht m, 10/25/98)	Measurement circuits and signal analy- sis in biological systems and medicine. Nonlinear devices. Signals in continuous and discrete time systems. Modulation. Digital signal processing. Sampling. Noise. Digital filters. Computer design and use for biomedical instrumentation. A term project and oral presentation are required.
Case Western Reserve	EBME 308: Biomedi- cal Signals and Sys- tems http://convolve.cw- ru.edu/ undergrad_dir/under- grad.html, 4/13/99	Quantitative analysis of biomedical sig- nals and physiological systems. Fourier and Laplace transforms. Frequency re- sponse of systems and circuits. A/D con- version, sampling, and discrete-time signal processing. Filter design. Labora- tory and computational experiences with biomedical applications.
Duke University	BME 164: Biomedi- cal Electronics and Measurements http:// www.duke.edu/~krn/ bmepage.html, 4/13/ 99	The goal of this course is to teach the ba- sic principles required for the design of biomedical instruments that can be used to monitor physiological processes. Transducers/Sensors. Op-Amps: Filters, Differential and Instrumentation Ampli- fiers. Signal Processing: Filtering, Cor- relating, Averaging. Analog to Digital Converter (ADCs), Digital to Analog Converters (DACs). Fourier Trans- forms, Fourier Series, Sampling. Re- cording and Display Devices. Biopotentials, Cardiac System Over- view, Cardiac Instruments. Telemetry. Medical Imaging Overview.
Duke University	BME 171: Signals and Systems http://regis- trar.duke.edu/ACES/ course_descrip/ BME/171.html, 4/13/ 99	Convolution, deconvolution, Fourier se- ries, Fourier transform, sampling, and the Laplace transform. Continuous and discrete formulations with emphasis on computational and simulation aspects and selected biomedical examples.

Table 2	Course D	escriptions
---------	-----------------	-------------

Institution	Course, WWW page ¹	Description
Harvard University – Massachusetts Insti- tute of Technology	HST 582J: Biomedi- cal Signal and Image Processing (Same subject as 6.555J, 16.456J) http://web.mit.edu/ 6.555/www/, 4/13/99	Fundamentals of digital signal process- ing with particular emphasis on prob- lems in biomedical research and clinical medicine. Basic principles and algo- rithms for data acquisition, imaging, fil- tering and feature extraction. Laboratory projects provide practical experience in processing physiologic data, with exam- ples from neurophysiology, cardiology,
University of Iowa	51:180: Biomeasure- ments II	Part 1): Introduction and review of linear system theory
	http://www. engineering.uiowa. edu/~eldove/Courses/ 51_180.html, 4/13/99	Part 2): The theory and application of biomedical sensors and transducers Part 3): The theory and application of digital signal processing Part 4): The theory and application of real-time programming
University of Miami	BME614: Microcom- puter-Based Medical Instrumentation http://www.eng. miami.edu/cgi-bin/ CoE/show_course? BME+614, 4/15/99	Principles and design of microcomputer- based biomedical instruments, analog and digital signal conversion, micro- computer hardware and software design, algorithm development for medical ap- plications, medical signal processing with microcomputers, software safety in life support systems and current applica- tions.
University of Miami	BME622: Biomedi- cal Signal Processing http://www.eng. miami.edu/cgi-bin/ CoE/show_course? BME+622, 4/15/99	Applications of the digital signal pro- cessing techniques to biomedical signals originating from cardiovascular, respira- tory, neural, sensory and motor systems. Digital filtering, averaging, spectral analysis, signal estimation, detection and classification. Real-time and on-line applications will be emphasized.
University of Michi- gan	EECS/BME 458: Biomedical Instru- mentation http://www.bme. umich.edu/courses/ eecs-bme-458/ index.html, 4/15/99	Measurement and analysis of biopoten- tials and biomedical transducer charac- teristics; electrical safety; applications of FETs, integrated circuits, operational amplifiers for signal processing and computer interfacing; signal analysis and display on the laboratory minicom- puter. Lectures and laboratory.

Table 2 (Continued) Course Descriptions

Institution	Course, WWW page ¹	Description
Milwaukee School of Engineering	BE 433: Biomedical Digital Signal Pro- cessing http://www.msoe. edu/~barnicks/ courses/be433/ be433.htm, 4/15/99	The objective of this course is to present the principles of digital signal process- ing and to have the student apply these methods to the analysis of biological sig- nals such as EEG, ECG, evoked poten- tials, etc. Topics include correlation, signal averaging, z-transform, discrete- time system analysis, discrete and fast Fourier transform, and digital filtering. In the laboratory, the student is required to design hardware and software to per- form the analysis on various biopotential signals.
Northwestern Univer- sity	C20: Biomedical Sig- nals and Imaging http://www.nwu.edu/ bme/bmecrs.htm, 4/15/99	Fundamentals and signals and systems, time and frequency domain issues. Fou- rier series and transforms, convolution, frequency response, filtering, and labo- ratories in biomedical systems and imag- ing.
University of Pennsyl- vania	301: Bioengineering Systems http://www.seas. upenn.edu/be/ courses_page/ courses.html, 4/15/99	Properties of signals and systems and ex- amples of biological and biomedical sig- nals and systems; linear, time invariant systems; Fourier analysis of signals and systems with applications to biomedical signals such as ECG and EEG; frequen- cy analysis of first and second order sys- tems; the frequency response; introduction to filtering, sampling and feedback with biomedical applications.
University of Pennsyl- vania	BE 465: Biomedical Instrumentation (see also 565) http://homepage.seas. upenn.edu/~kfoster/ be465.htm, 4/15/99	Design of electronic instrumentation for the recording and analysis of physiolog- ical signals, with special emphasis on computer applications. Noise and inter- ference. Design and analysis of simple amplifiers and filters for signal condi- tioning; applications of digital filters. Electrical safety requirements. Course will require extensive circuit and com- puter design work.

 Table 2 (Continued) Course Descriptions

Institution	Course, WWW page1	Description
University of Pitts- burgh	EE 1580/2580/ BIOENG 1580/2580: Biomedical Applica- tions of Signal Pro- cessing http://jupiter.ee.pitt. edu/~boston/ EE1580.htm, 10/25/ 98	The goal of this course is to provide stu- dents with experience in applying signal processing techniques to medical and bi- ological data. Electrophysiological sig- nals will be used to illustrate the techniques, and the physiology of these signals, including how they are generat- ed and how they are used clinically, will be presented. Feature extraction, digital filtering, spectral analysis, and automat- ed interpretation of signals will be dis- cussed, with examples drawn from current problems in clinical medicine and research.
University of Pitts- burgh	BIOENG 2510: Bio- medical Signal Pro- cessing and Analysis http://jupiter.ee.pitt. edu/~boston/ bioeng2510.htm #home, 4/15/99	This course reviews the mathematical techniques used to analyze continuous and discrete signals and systems. Topics include methods to describe continuous, discrete, and stochastic signals; repre- sentation of continuous and discrete sys- tems; simulation of systems in MATLAB; frequency domain analysis, using Laplace, Fourier, and z-trans- forms; sampling; the discrete Fourier transform; filter design; feedback con- trol systems. Biomedical engineering applications that illustrate system stabil- ity, frequency response analysis, signal sampling, and filtering are investigated through theoretical exercises and MAT- LAB computer software.
Purdue University – Indiana University	Biosignal Process- ing ² http:// BioMed.www.ecn. purdue.edu/BioMed/ Courses/Courses/ Biosignal_Proces- sing, 4/15/99	This course will cover topics in signal processing which apply to biomedical signals and use the results of signal pro- cessing tools to model and interpret physiological systems. This will include signal acquisition, digital filters, signal detection and modeling, noise reduction, spectral estimation, and adaptive signal processing techniques. Typical signal sources will be cardiac and neural in or- igin ranging from ion channel recordings to body surface measurements, e.g. ECG and EEG.

 Table 2 (Continued) Course Descriptions

Institution	Course, WWW page1	Description
Vanderbilt University	BME 263: Signal Measurement and Analysis http://www.bme. vanderbilt.edu/ Undergraduate/ classes/ bme263.html, 4/15/ 99	Discrete time analysis of signals with de- terministic and random properties and the effect of linear systems on these properties. Brief review of relevant top- ics in probability and statistics and intro- duction to random processes. Discrete Fourier transforms, harmonic and corre- lation analysis, and signal modeling. Im- plementation of these techniques on a computer is required.
Wright State Universi- ty, Dayton	466: Biomedical Sig- nal Processing http://www.cs. wright.edu/bhe/ bmecourses.html, 4/ 15/99	Introduction to biomedical signal pro- cessing and analysis. Topics include characteristics of various biomedical signals; time-domain and frequency-do- main; continuous and discrete signal representations; and applications of dig- ital and random signal processing meth- ods to the analysis of biomedical signals.

Table 2 (Continued) Course Descriptions

Table Notes:

1. WWW references are to specific course home pages when possible. Date of access is given after each URL.

2. This course no longer exists. Students now take a DSP course in EE concurrent with a laboratory from the BME department.

2.3.2 Context

2.3.2.1 Course Goal

The goals of these BME courses with respect to the DSP content fall into three categories:

1 Familiarity with DSP as a tool—i.e., a survey experience, with an emphasis on a "cookbook" approach to using DSP techniques. Often, the main focus of the course is on a broader BME topic.

2 BME equivalent of traditional DSP course. The course covers DSP topics in a way that provides a foundation for more advanced DSP coursework. Emphasis on more theoretical aspects, theory over application.

3 Biomedical-specific DSP. These courses assume some background in DSP (possibly from a traditional EE course), and address advanced DSP topics that may be specific to BME applications.

The majority of courses described in Table 1 fall into the first two categories, since introductory DSP was the area of interest in collecting this data. However, elements of the third category appear in some of the graduate-level courses noted, and figure into the philosophical comments by some instructors, to be discussed later.

2.3.2.2 Student Background

The survey results show great variety in student backgrounds. Most BME students, as engineering students, have a background in mathematics through differential equations. However many of the courses listed do not assume or require any background in other areas such as linear systems and signals that are usual prerequisites for DSP study in a traditional EE curriculum.

Students may have had exposure to biomedical topics such as biosignals, bioinstrumentation, and safety, prior to studying DSP, but in some cases these topics are covered concurrently with DSP. Further, for courses with a practical emphasis, topics from

electronics including analog circuits, digital logic/microprocessors, and programming may not be part of the students' shared background and must be covered.

2.3.2.3 Department

Department-level issues have a large influence on BME courses. The interest and knowledge of faculty is reflected in the goals, content, frequency of offering, and methods of courses. Related is the issue of department size; in some cases, the BME department is much smaller than other engineering departments or may be an elective subspecialty in one or more traditional departments. In such cases students may take many traditional engineering courses that have little or no biomedical emphasis. In other cases, where the BME department is larger than other departments (e.g., Duke University), most engineering classes might be taken within the department, and hence will have a biomedical emphasis.

ABET (Accreditation Board for Engineering and Technology) criteria, which among other expectations include engineering design and a "major engineering design experience" (e.g., "capstone" course) [5], are also influential in structing BME programs.

2.3.3 Content

DSP is not necessarily the main topic of the classes surveyed. Often, DSP is merely a tool in a context such as the design of bioinstruments, or a subject treated at the tail end of a study of biosignals. Table 1 shows courses with a great deal of content in both biomedical and electronic areas, areas that are both uncommon in traditional DSP courses, and leave less time for an in-depth treatment of strictly DSP issues.

An almost invariant prerequisite for the traditional study of DSP is the knowledge of linear signals and systems. Many of the surveyed courses include this prerequisite information, since students have not had this coursework previously.

With respect to DSP topics, BME courses vary widely. (It must be emphasized, however, that not every syllabus or course description provides the same level of detail, so "fine-grained" comparisons are not always possible.) In some cases, advanced topics such

as detection and estimation are discussed (e.g., Purdue University - Indiana University "Biosignal Processing"), while syllabi from other courses show so few days on DSP that only broad treatments of DSP topics are possible.

One recurrent item is the topic of signal averaging, which may be given more emphasis in even introductory BME DSP due to its simplicity and its application to common biosignals such as the ECG.

Two other items of note will be discussed later—image processing and the practical implementation of DSP.

2.3.4 Homework and Laboratories

Again reflecting specific goals, many of the surveyed courses include laboratories. In some cases, for example at Duke University, students process data acquired using themselves as subjects, employing instruments they construct themselves.

MATLAB, a numeric mathematics program from The MathWorks, plays a role in almost every class listed. It is used for homework and laboratories, often with real data. LabView (from National Instruments) was the only other software mentioned. Several instructors mentioned the usefulness of real data. With the availability of several databases (e.g., MIT-BIH ECG database [10], Signal Processing Information Base (SPIB) [11]), real data may also be easier to provide than synthetic signals. The relatively low bandwidth of common biosignals (e.g., ECG) with respect to current analog-to-digital converters (ADCs) and digital processors (including microprocessors) means real-time processing is possible without requiring extraordinary skills in areas such as high-speed analog data acquisition or tightly-coded assembly language programming.

Some classes involve projects (usually in classes with laboratories). It is not clear that these projects always involve significant DSP, since DSP is not often a central topic in the class. However, with relatively inexpensive hardware (typically less than \$100), realtime processing of signals is feasible even in the context of a course offering a fairly broad overview of DSP. For example, MATLAB can be used to generate filter parameters, which

can then be introduced into an existing assembly-language program for download to an inexpensive DSP board for real-time processing.¹

2.4 Conclusions About DSP in Current BME Courses

Introductory DSP in BME is taught in a variety of contexts, ranging from brief overview to in-depth study. Student background can vary widely as well, often lacking prerequisite material that can be assumed in an EE curriculum. BME courses may also introduce DSP techniques (e.g., signal averaging) that would not ordinarily be featured in a DSP course. These techniques have particular application to biomedical signals and, as in the case of signal averaging, are fairly easily approached on an intuitive level. Laboratories and the use of MATLAB provide hands-on DSP experience and motivation in many courses.

3 DSP for "Non-EE" Biomedical Engineering Students

The previous chapter has defined some challenges produced by the diversity in BME DSP education. Several issues explain this diversity and lay the groundwork for the material in the final chapter of this thesis. Clearly, biomedical DSP courses may have several different goals. In addition, there may be significant differences between BME DSP and "traditional" DSP courses in terms of content. Finally, several pedagogical themes are present in both existing classes and in the comments of instructors to be mentioned below that may be important in teaching biomedical DSP.

3.1 Biomedical DSP vs. "Traditional" DSP

Instructors interviewed for this survey² were asked if biomedical DSP differed in any way from that of "traditional" DSP, and what implications this had for how DSP should be taught to BME students. There was variation in the background and experience

¹. This approach was used in producing assembly language examples of FIR and IIR filters in [7].

^{2.} While there is considerable overlap, not every instructor interviewed teaches a course listed in Table 1, nor was the instructor for every course in the list interviewed.

of the instructors contacted, ranging from M.D.'s with many years of biomedical teaching experience to more recent professors. Some instructors stated that, at least at the introductory level, BME DSP did not have any special characteristics. However most instructors did offer reasons why BME DSP was different from traditional DSP. These differences fell into the following categories: the nature of biosignals, the role of problem-solving in BME, background and goals of students (already discussed), and the purposes of processing biomedical data.

3.1.1 The Nature of Biosignals

More than any other factor, instructors stated that BME DSP differed from "generic" DSP in terms of the signals processed. The low signal-to-noise ratio (SNR) was frequently cited. Some fields of seismology or space communications may deal with low SNR, but many biological signals (e.g., EEG) are routinely acquired in the midst of much noise from environmental and other biological phenomena, often as the result of taking measurements at some distance from the source. These signals also typically have very small amplitudes, due to both the source itself and attenuation caused by intervening tissue. This requires the solution of non-DSP problems (e.g., analog signal conditioning) and hence rewards a systems-level approach to biomedical signal processing.

Biomedical signals also are often of low bandwidth. This makes signal processing in real time easier for students since it is not necessary to use high-speed processors (e.g., DSP chips) or carefully optimized software. Low-bandwidth signals also make acquisition and other non-DSP tasks more tractable for novices. Thus a systems-level approach can be explored in laboratory experiences.

One instructor pointed out the similarities between speech processing and biomedical DSP. In both cases, a general DSP background is necessary. However, for many of the well-studied biosignals (e.g., ECG, EEG, EMG, etc.), specific DSP techniques have been developed that, while very efficient and useful in studying a particular biosignal, have little application to other signals (biomedical or not). This is also the case with speech pro-

cessing, where techniques with broad application (e.g., autocorrelation) are used along with highly specific techniques tuned to the special characteristics of the speech signal (e.g., low bit-rate speech coders). Since it is not possible for a single advanced DSP course to cover the many specific techniques from the dozens of fields that use DSP (including biomedicine), this is justification for BME-specific DSP courses in the opinion of those interviewed.

3.1.2 Problem-Solving in BME

Another aspect mentioned by instructors is the need for interdisciplinary problemsolving abilities in BME, addressed by courses that have students drawing upon several different disciplines, e.g., from biochemistry and programming microprocessors, to solve a signal processing problem. These courses work well as "capstone" courses, drawing together a number of fields the student has studied in previous years, as described by the following excerpt from the ABET Criteria for Accrediting Engineering Program [5]:

Each educational program must include a meaningful, major engineering design experience that builds upon the fundamental concepts of mathematics, basic sciences, the humanities and social sciences, engineering topics, and communication skills. The scope of the design experience within a program should match the requirements of practice within that discipline.... Design cannot be taught in one course; it is an experience that must grow with the student's development. A meaningful, major design experience means that, at some point when the student's academic development is nearly complete, there should be a design experience that both focuses the student's attention on professional practice and is drawn from past course work. Inevitably, this means a course, or a project, or a thesis that focuses upon design. "Meaningful" implies that the design experience is significant within the student's major and that it draws upon previous course work, but not necessarily upon every course taken by the student.

3.1.3 Purpose of Processing Data

The processing of biosignals also has two non-technical characteristics that distinguish it from many other DSP tasks. First, the correct processing and interpretation of some signals can be life-critical. Second, the results of processing biosignals are often interpreted by medical doctors who are trained to interpret results presented in certain formats which may not be efficient, optimal, or even appealing from an engineering point of view. While concerns about safety are inherently present in other engineering disciplines (e.g., civil engineering, control systems), these concerns are rarely addressed in EE-based DSP courses. In contrast, safety issues are very important both in an immediate sense (e.g., students collecting real data in a laboratory) and as design considerations for the practicing biomedical engineer.

The fact that some types of biosignal processing are interpreted by medical doctors not only adds a need for interdisciplinary communication (useful to any engineer, but critical in BME), but may place restrictions on the types of processing that is useful. For example, ECGs are recorded and processed in standard ways, and can be interpreted by doctors in ways that are generally consistent with other doctors. However, other methods of detecting, processing, and presenting this activity might be more accurate, easier to interpret, etc. The conventional format of presentation of the ECG is firmly entrenched in the training and practice of medicine, however, and unlikely to change for merely incremental benefit. (It might be argued that in fact, since the ECG in particular is so firmly embedded in medical practice, any replacement would have to offer quite substantial benefits.) The problem the biomedical engineer is solving, therefore, is not a purely technical one, but one influenced by historical and other issues. (On the other hand, one may argue that the extraordinary explosion of new sensing techniques is assurance the field is not bound helplessly to ingrained practices, since many of these signals will have no precedent.)

3.2 Useful Pedagogical Themes

From course descriptions and instructor comments, a few themes in biomedical DSP courses emerge. Not every course embodies all these, though most of these themes are mutually compatible. Among these are the use of laboratories and projects, MATLAB, actual biosignals, real hardware, and real-time programming.

Laboratories and projects were often mentioned as highly motivating (though challenging) aspects of courses. Laboratories allow students the opportunity to use hardware

and/or software to which they otherwise might not have access, or to construct devices or perform experiments in a controlled environment. Courses that discuss the acquisition of biosignals allow students to acquire, process, and display biosignals, promoting systemslevel design. Projects also allow more open-ended and creative work, and can be designed to require interdisciplinary knowledge and communication skills that homework or laboratories alone do not exercise. Both laboratories and projects are especially appropriate to biomedical DSP courses with a strong applications orientation.

No other mathematics program was mentioned by the interviewed instructors nearly as often as MATLAB, regardless of the orientation of the course. Several reasons are likely: the ability to process discrete-time signals of reasonable length (e.g., 16K samples in recent educational versions), fairly low cost, frequent support in popular textbooks, widely used in engineering, built-in support for many signal processing functions (including digital filters, frequency analysis), and good output support (screen and print graphics, sound). The ability to explore both theoretical concepts (e.g., the effect of zero-padding in a DFT) and practical applications (e.g., removing noise from a pre-recorded ECG) is a powerful combination. However, several instructors commented that they would like to be able to have students interact more easily with physical analog I/O devices, and even do real-time processing, which is not easy to do with MATLAB.

Even when it is not possible for students to acquire biosignals from subjects, many of the surveyed courses employ real biosignals as examples in class, homework, and laboratories. Real biosignals are motivating for students to use, and can often be successfully processed by students using even modest signal processing knowledge. Real biosignals also provide the opportunity to observe subtle real-life issues that makes the students aware of the limitations of simple models of biosignals. These limitations must be acknowledged, but according to some instructors interviewed, may also provide motivation for more advanced study.

The use of real hardware is an issue influenced by both course content and the state of technology. Unless a course has hardware as a main emphasis, the topic may require far more course time than is available, particularly if students require a review of hardware topics in preparation for the study of DSP. However, much as the use of actual biosignals is both motivating and instructive, the use (and perhaps design) of hardware for acquisition and processing of biosignals can be a worthwhile aspect of a course, as seen by their inclusion in several courses in Table 1.

Closely allied with the use of real hardware is the topic of real-time programming. One instructor noted the job-hunting advantage students receive from experience in programming DSP chips for real-time DSP. Several recent trends make providing this experience a realistic proposition (though presently realized in few of the classes surveyed). First, many manufacturers of DSP chips have very inexpensive evaluation systems available that include serial-interfaced hardware, analog input/output, on-board memory, and development software.³ Second, software such as MATLAB can easily be programmed to present the user with a simple set of options, then generate the necessary file or files for download to the evaluation board. The student can thus explore a wide range of topics, from choice of filter type to real-time DSP programming issues or even analog signal conditioning. The low bandwidth of many biosignals considerably reduces the processing required—and even makes possible the use of common microprocessors and microcontrollers for real-time processing of some biosignals.

Biomedical image processing has several interesting characteristics that set it apart from one-dimensional biosignals. Images have motivational value for readers (discussed in a later section), but they can be a "door to nonlinear processing," as one instructor put it. Even in an introductory setting, it may be very useful to use easily accessible non-linear

^{3.} See for example Motorola DSP56002 Evaluation Module Kit (\$149, http://www1.motorola-dsp.com/docs/kit-evm.html), Texas Instruments 'C5x DSP Starter Kit (\$99, http:// www.ti.com/sc/docs/tools/dsp/tmds3200051.html), Analog Devices ADDS-21xx-EZLITE (less than \$100, http://products.analog.com/products/info.asp?product=21XX-EZLITE).

image enhancing operations as examples of (otherwise ignored) nonlinear operations. Computed imaging also offers another set of tools that integrate linear filters with more geometric operations.

4 Outline for a Proposed Text

This chapter outlines a particular approach to teaching DSP in a biomedical context, based in part on [7], and refined through the investigation undertaken in this work. This outline was developed with a textbook in mind, but might be useful for other purposes, such as a course syllabus. First, the audience for the book is identified, one that is centered on a particular subset of BME students. Next, the goals and objectives of the text are presented. The approach is then discussed—a set of broad ideals that guided the development of the outline. The outline itself is then presented, along with commentary on the choices made regarding specific content and approaches. Finally, some topics are presented that were not addressed in the outline, but would be valuable to include.

4.1 Audience

The audience for this text is described in terms of the students' background and interests with respect to biomedical DSP.

The educational background assumed is (by category):

• Electronics—basic AC/DC circuits theory; no linear signals and signals

• Mathematics—trigonometry; calculus; complex numbers (will also be reviewed in text)

• Biomedical—some background in biosignals and bioinstrumentation helpful

• Programming—exposure to basic programming (e.g., C, BASIC)

The assumed electronics background would ordinarily be provided by an introductory circuits course (which would typically include the Laplace transform, though that is not assumed). No linear signals and systems background is required, so students without this background are not at a disadvantage. The assumed mathematics background should be well within the comfort level of engineering students, even in engineering disciplines

where there is less emphasis on the calculus and other mathematical tools. Some exposure to biosignals and bioinstrumentation would be very helpful in creating a context and motivation for DSP, and would most appropriately come from a prerequisite BME course. Finally, since MATLAB will be used throughout the text, some exposure to simple procedural programming is assumed. (Use of MATLAB will be illustrated through the text, though probably it would be good to supplement with a MATLAB student manual.)

The goals of the audience are assumed to be quite simple: within the limits of diverse backgrounds, to get a broad, application-oriented exposure to DSP for use in BME careers. Students will learn to apply simple DSP techniques (including filtering and spectral analysis) to biomedical data, and to have a coherent (though informal) framework of DSP concepts. The text will provide a (non-rigorous) introduction to DSP that later could be filled-in with more traditional and theoretical courses.

There are alternative audiences that this text can address: technicians, practicing engineers in non-EE fields, researchers without engineering backgrounds, and medical practitioners who want a practical understanding of DSP techniques.

4.2 Goals and Objectives

The specific objectives for the text whose outline is presented here is as follows:

1 To expose the reader to DSP concepts with an emphasis on biomedical applications.

• To convey how DSP techniques relate to analog signal processing techniques.

• To explain the nature of discrete-time (DT) signals and systems vs. continuous-time (CT) signals and systems.

• To define measures of signal and noise, especially in biomedical applications.

• To describe the power of linear processing and provide examples of nonlinear processing.

2 To enable the reader to use basic DSP concepts to process DT signals and images.
To allow the reader to understand and specify filter type and parameters for a particular task.

• To allow the reader to design and apply basic IIR and FIR digital filters using computer-based tools.

- To allow the reader to compute and analyze spectra (e.g., FFT).
- To allow the reader to use other techniques (e.g., correlation) to process signals.

• To describe basic digital image processing, including representation, processing, and storage/communications.

• To explain typical compression approaches and allow the reader to implement simple compression techniques.

- To allow the reader to perform the above tasks using MATLAB.
- 3 To provide a foundation and motivation for further study.
- To use terminology and notation consistent with the DSP field.
- To point out areas where simplifications of concepts have been made.
- To refer to resources for further study.

4.3 Approach

Several principles have been used to guide the goals of the text within the constraints ofrelaxed prerequisites. These are providing a coherent framework, presenting abstract or highly mathematical concepts as they are needed and/or replacing them with design procedures, using concrete examples and realistic data, and use of MATLAB throughout the text.

4.3.1 Coherent Framework

The connections (often mathematical) among topics in DSP that may be obvious in a mathematically rigorous presentation are not as easily appreciated in less formal presentations. It is important, then, to consciously construct a framework for the reader that explicitly makes useful connections among topics that might otherwise appear disjoint. For example, without the relationship between convolution in time and multiplication in the frequency domain, the phenomenon of spectral leakage in the DFT must be explicitly linked to the magnitude spectra of windows. In addition to emphasizing connections, the

"destination" of the material should also be explicit, so that key topics such as linear systems are seen as being on the path to interesting goals, even if they do not appear to have immediate, practical application to the reader.

4.3.2 Presentation of Abstract Concepts

The volume of material, coupled with the background of readers, makes it difficult to thoroughly present every topic in basic DSP. Then too, the motivation of the reader may be attenuated by too many abstract mathematical formulations. Since the goal of the book is not to provide a rigorous foundation for DSP, mathematical concepts are introduced when needed, and are presented as tools for achieving a specific end, rather than as subjects of study themselves. So, for example, the *s*-plane can be employed as a tool for analog filter design, without necessarily requiring a discussion of the mathematics of the Laplace transform. Similarly, FIR filter design can be presented as a tool introduced to adjust the weights of a simple moving-average filter so as to enhance some aspect of the filter's frequency response. In addition, selected concepts can be reduced to design procedures—for example, conversion from lowpass to highpass filters in the *z*-domain through "cookbook" formulas. Built-in MATLAB functions can shield the reader from many issues (clearly at a cost), allowing the focus to be on problem-solving. The emphasis on applications is at the cost of superficial treatment of the the underlying theory, but this trade-off is appropriate if the goals of the text favor breadth over depth.

4.3.3 Use of Examples

Concrete examples, especially those linked to MATLAB scripts, are valuable in illustrating not only single concepts, but also problem-solving involving multiple steps. The use of realistic signals in examples, especially those available to the reader electronically, has several advantages.

Examples need not be just numeric or visual—most computers can output sound, and for the special case of speech, this is the most natural output method. Using multiple

output modes may not only increase the reader's motivation, but may also convey concepts and processes more clearly than through numerical or graphical representations.

One instructor noted that two-dimensional graphics (images) likewise may have significant value in illustrating concepts, particularly processes for which one-dimensional or auditory representation may be difficult to appreciate. Recent versions of the student edition of MATLAB have the ability to process images that are sufficiently large for interesting and motivating examples.

4.3.4 Use of MATLAB

Several arguments exist for using MATLAB in the proposed text. First, MATLAB allows students to explore applied mathematical concepts in an environment that combines ease of use with a rich set of tools. (However, a symbolic mathematics tool like Mathematica or MathCad can have distinct advantages when exploring concepts of a more theoretical and symbolic nature.) Second, MATLAB specifically supports such signal processing operations as filter design and frequency analysis, and is well suited to operations on arrays and matrices. Third, MATLAB is widely used in engineering in general, and in signal processing in particular. Readers can expect to encounter MATLAB in further readings or courses, and the ability to use MATLAB itself may therefore be a valuable skill.

In the proposed text, most illustrations will be associated with MATLAB scripts located within the text. This serves several purposes. The reader will obtain additional insight into what the illustration represents. The reader may experiment with changes to the code, or modify it for other purposes. Finally, the listings illustrate basic MATLAB programming, for those readers unfamiliar with MATLAB. These goals are more easily realized by making the scripts and data available electronically (on CDROM and/or via the World-Wide-Web), versus having them only appear in print in a text.

4.3.5 Exercises

The exercises to accompany the proposed text will incorporate the following ideas:

• Exercises will cover a range of complexity, including very simple exercises likely to be solved successfully by most readers, to more complex topics (e.g., project-level).

• Some exercises will involve multiple concepts, requiring several stages of problemsolving and/or integration of several concepts.

• Design problems will ask for high-level problem solving (i.e., to select appropriate techniques, rather than "plug in" numbers).

• Exercises requiring the reader to demonstrate/illustrate a concept from the text or a new concept. (This new concept will usually be an extension or extreme case of a concept covered).

• Answers or hints to some subset of problems. (Also valuable to readers not in a course.)

4.4 Realization

In this section, the outline of a realization of such a text will be described. First, the overall structure of the text will be discussed. Next is a more detailed chapter-by-chapter discussion. This is followed by some comments on topics not included in the text, then a description of additional end material.

4.4.1 Outline

Appendix A is the outline of a text on introductory biomedical signal processing, aimed at the limited audience described in Section 4.1. The seven chapters are 1) Introduction, 2) Signals & Systems, 3) Linear Digital Filters, 4) Frequency Analysis, 5) Correlation and Signal Averaging, 6) Image Processing, and 7) Compression.

The first four chapters cover material that is central to basic DSP. The last three chapters cover material that, while not usually found in EE-based introductory DSP courses, nonetheless have biomedical applications and are accessible to the intended audience. The next chapter discusses (auto- and cross-) correlation and signal averaging. Without a stochastic systems background, the discussion must remain on an intuitive level in several areas (e.g., the topic of setting a threshold value for detecting a known signal in
noise using cross-correlation). The following chapter on image processing illustrates nonlinear processing and computed imaging as well as other techniques. The final topic, data compression, is also both motivating for readers, and illustrative of a problem-solving approach distinct from one based on linear systems. In particular, some forms of data compression (particularly those used for compressing ECGs) illustrate empirical, time-domain oriented algorithms. (Whether these empirical techniques give the field of biomedical DSP an inaccurate image⁴ is open to debate.)

4.4.2 Commentary on the Chapters

4.4.2.1 Introduction

The purpose of this chapter of the text is to provide a brief overview of signal processing and motivate the study of DSP by comparing analog and digital signal processing. This chapter provides motivation for many of the following chapters, since DSP applications are discussed along with limitations of analog processing.

4.4.2.2 Signals & Systems

Chapter 2 addresses continuous and discrete-time signals and systems in three parts. First, the ECG and other one-dimensional biosignals (e.g., speech) are used to establish the time domain representation. At this stage MATLAB is also introduced. The inherently discrete nature of MATLAB is not noted here, but deferred to a later section in the chapter treating discrete-time issues.

Second, the frequency domain representation of signals is discussed. There are two approaches to the concept—a tunable-filter illustration (i.e., measuring the energy passed as a function of center frequency for a tuned filter) and a synthesis argument (that one can build "any" signal out of sinusoids of appropriate amplitude and frequency). The latter illustration is used to introduce the concept of phase, once it is shown that magnitude and frequency alone give an incomplete representation. The inverse (Fourier) transform is also

^{4.} No pun intended.

introduced conceptually, as is the idea of filtering by attenuation (multiplication) in the frequency domain. A reference is made to the implicit assumption of stationarity (addressed in the chapter on frequency analysis).

Some notation and other practical concepts are discussed, such as the use of relative logarithmic measures (dB), logarithmic frequency scales, and radians/sec measures of frequency.

The third section discusses discrete and digital signals and systems. The basics of sampling are covered (Nyquist rate, frequency), but the additional topics of phase and negative frequency (spectral inversion), and undersampling are also included. These topics provide a more complete view of sampling, but could be omitted without much loss of continuity.

The issue of discrete versus digital signals is given some attention, both as a practical matter (readers who wish to consult other texts will benefit from knowing the rough equivalence in meaning for textbook titles) and philosophical, since it explains why DSP design is typically done assuming discrete-time. This topic also leads into the issue of quantization (and digital-to-analog conversion). Several tables and illustrations give parameters of ADC and digital-to-analog converters (DACs).

As a final point, discrete frequency is briefly addressed (mostly a reference to the chapter on frequency analysis where it is dealt with in greater detail). However, normalized frequency is covered, as well as the topic of how "real world" and normalized frequency are symbolized (i.e., F, f, Ω and ω).

4.4.2.3 Linear Digital Filters

Digital filters are introduced by examining the simple moving average filter. This leads to a difference equation, and also (via MATLAB) the observation that using different weights can "improve" the filter response. Having introduced the problem (for FIR filters at least), the first section of this chapter covers the specification of filters, including the

difference between "digital" and "analog" filter ripple.⁵ Time-domain specifications are also discussed since signals such as the ECG are often analyzed in the time domain.

Little attention is given to FIR filter design methods other than Parks-McClellan, due to the many advantages of that algorithm. Simple integer filters are discussed however, since they are very simple to implement, easily designed, easily derived from the moving average filter, and quickly executing (though not spectacular in performance). Some FIR implementation issues (e.g., common structures, fast convolution) are discussed briefly, along with comments on the multiply-accumulate (MAC) operation of DSP chips.

A discussion of IIR filters typically emphasizes the design of analog filters, however a cookbook approach (MATLAB oriented) is used here. After an overview of IIR filter design methods, the analog prototype method is introduced, but MATLAB is used for design of these filters and even used to perform conversion to the z-domain. It is worthwhile exploring the characteristics of the actual analog prototype filters since some of these have utility in analog processing steps (apart from their use in IIR design), and the use of analog prototypes remains a popular IIR design method even in the face of direct software-based design methods. The issue of implementation is somewhat more involved for IIR filters than FIR—here issues such as optimum/efficient structures, finite-wordlength effects, and stability can not be addressed in great depth, and the goal is to provide familiarity with these issues. (This lack of depth is problematic to the extent the higher efficiency of IIR filters can be useful in low-speed processors like those used in portable or implantable medical devices; the low bandwidth, and need for linear phase, of many biomedical signals is a point in favor of often slower FIR filters.)

The chapter ends with a summary comparing FIR and IIR filters in table form.

^{5.} Filter ripple specifications can be measured peak-to-peak or maximum deviation from a mean. The former is often associated with passive analog filters (whose gain is always less than unity), and the former with digital filters. See [7], p. 77.

4.4.2.4 Frequency Analysis

There are six main parts to the chapter on frequency analysis. First, a brief introduction, which is followed by the discrete Fourier transform (DFT), the fast Fourier transform (FFT), the inverse DFT, and time-frequency analysis. A section on other transformations ends the chapter.

The DFT is approached by looking at the defining equation (in trigonometric form) from the point of view of cross-correlations of sinusoids with the signal to be analyzed. (Since the correlation operation is not discussed until a later chapter, the operation is described intuitively.) This leads to a discussion of phase and interpretation of the complex output (sin/cos or magnitude/phase). Leakage and resolution (i.e., window shape, length, and zero padding) are also discussed, with several tables summarizing key parameters.

The FFT is presented as a class of efficient DFT algorithms, with emphasis on input and output requirements (i.e., the need to perform bit reversal depending on the implementation) and the relative performance over a simple implementation of the DFT. The complexity of efficient FFT routines is such that more detailed information about FFT routines is not justified—FFT routines are rarely written by users, as compared to digital filters (especially FIR) that are often easily programmed, even in assembly language.

The inverse DFT/FFT (IDFT/IFFT) was foreshadowed by earlier discussions in Chapter 2, so the discussion here is more practical—for example, the common practice of sharing FFT and IFFT code with a flag to indicate the direction of the transform.

Time-frequency analysis is motivated by two examples of nonstationary biosignals: speech and EEG. Multiple, overlapping FFTs are described, the discussion of which leads to a discovery of the trade-off between time and frequency resolution.

Other transforms, particularly the discrete wavelet transform, close out the chapter. The wavelet transform is worthwhile mentioning at this introductory level for several reasons. First, wavelet transforms can offer a useful time and frequency resolution trade-off,

providing a linear time resolution and logarithmic frequency resolution that is a good match to many signals. Second, and a consequence of the first, some formulations of the discrete wavelet transform offer substantial savings in computation over the FFT.⁶ Third, some signals are more efficiently represented using wavelet bases than the sinusoidal bases of the DFT.⁷ For all these reasons, the wavelet transform is likely to become more popular in many fields, including BME, so it is important to discuss at least the broad characteristics.

4.4.2.5 Correlation and Signal Averaging

Signal detection and estimation is usually covered after some study of stochastic systems, but some operations are quite simple, yield good results, and do not suffer a lot from a lack of rigor in developing the theoretical background. The three techniques discussed in this chapter are crosscorrelation, autocorrelation, and signal averaging.

Crosscorrelation is introduced as a tool to address two different problems: detection of a known signal in additive noise, and estimating time delay. Both have direct biomedical applications (e.g., ECG patterns, ultrasound imaging) and can be effectively explored using MATLAB. As is true for all the topics in this chapter, the mathematical justification for some aspects of these techniques is omitted, but cookbook formulas for setting thresholds are offered.

Autocorrelation is treated primarily as a tool for detecting periodicity; biosignals such as voiced speech and ECG with easily measured periods make good examples.

Signal averaging ends the chapter, though it is conceptually the simplest of the techniques covered. Here explicit equations for SNR versus number of signals averaged are given, and a version of the central limit theorem is given. Biosignals that are not pre-

^{6.} The student should be made aware that nothing "magic" is happening with respect to the required computation-the savings derives from the logarithmic frequency scale, and when calculated such that the frequency scale is linear, wavelet transforms have the same order of computations required as other methods like the FFT. See [2], p. 40.

^{7.} See, for example, [2], pp. 6-7.

cisely periodic present some issues surrounding the identification of fiducial points; this can be addressed using autocorrelation or a brief overview of time-domain techniques for locating relatively robust fiducial points. The high-resolution ECG is used as an example. (See, for example, [3].)

4.4.2.6 Image Processing

This chapter concentrates on operations on images and the construction of images from projections.

Operations on images are divided into operations on single pixels and operations over areas. Single-pixel operations offer opportunities to explore linear and non-linear operations. Contrast adjustments, clipping, thresholding, and other simple operations require little mathematical rigor for their implementation, and the degree of image enhancement can be evaluated subjectively (e.g., comparing "before" and "after" images). Area operations are examined using separable two-dimensional linear filters (which can be developed using one-dimensional filter material from earlier chapters) and one- and two-dimensional nonlinear operations (e.g., median filter).

Image construction from projections (computerized tomography or CT) is a powerful tool that can be described using simple trigonometry and the image filtering concepts just discussed. Using back-projection, the basic principles of CT are shown, while revealing the need for additional filtering of either the resulting image (using a two-dimensional filter) or of the projections (using a one-dimensional filter). Readers will be asked in an exercise to write a simple CT program, or to take an existing CT program and modify it (e.g., by adding the pre- or post-filtering). A static file or MATLAB program can provide the raw projection data for student exercises. A MATLAB program can generate data with an arbitrary number of projections, resolution, etc., which provides more flexibility when crafting exercises.

4.4.2.7 Data Compression

Data compression is a subject that, while not frequently included in the classes surveyed in this project, has several useful insights to offer to the reader. Like image processing, it is a field that most readers will have encountered (e.g., ubiquitous .ZIP files) and can easily evaluate (objectively for lossless compression, subjectively for lossy). It is an important element of communication and storage of biomedical data. In studying compression, links are made to concepts such as efficient representations (e.g., DFT coefficients). Finally, many compression algorithms (especially those in the time domain) are simple and easy to implement.

Lossy and lossless compression are discussed, leading to issues of redundancy and noise. Another issue discussed is the importance of knowing the characteristics of the human sensory systems, which can reveal methods for greatly reducing the amount of data necessary to convey the same information. (A familiar example is that of conveying the illusion of movement by sequential static images, as in motion pictures.)

The compression techniques discussed fall into two (arbitrary) categories—techniques based on expressing the signal in terms of an efficient basis (see, for example, [4]), and other methods. Both categories are essential to BME—wavelets and other bases can allow efficient representation of many types of signals; on the other hand, run-length encoding and empirically-derived time-domain algorithms can produce substantial compression with minimal computation and simple implementations.

4.4.3 Additional Content

The outline of the proposed text lists the following additional contents:

• Glossary: Provides both a brief definition and a reference to the place in the text where the term is first introduced. Abbreviations are also included in the glossary.

• Table of symbols used: (Ideally on the inside covers) A list of the symbols used in the book, with brief description and reference to the place in the text where they are first introduced.⁸

• CDROM/electronic files: Available either on a CDRCM bound into the book or via a website, these files include sample signals and images, MATLAB examples, and a "home page" of Internet resources.

4.5 Topics Not Covered But of Interest

The proposed text could be expanded to cover real-time biomedical DSP issues, including the framework for using real-time DSP hardware. This is an area that would be valuable to add since it is not an area covered well in existing texts, can be fairly inexpensive to add to a course, and is a skill that adds to a student's marketability. However, this area would be difficult to cover well unless some microprocessor background (architecture, programming, or both) could be assumed on the part of the student. One compromise would be to include such a chapter for students with an appropriate background, but that could safely be omitted in situations where time or background was lacking.

Students can easily perform real-time DSP on biomedical signals. Since many biomedical signals are low bandwidth, it would not be necessary for students to write highly efficient real-time software. As a result, students could write software in C, with little or no concern for the actual architecture. The quality and resolution of the audio analog-todigital converters on low-cost DSP boards would suffice for many biomedical signal applications, assuming proper signal conditioning was used.

Another area not addressed in the proposed text is that of laboratory exercises and projects. Both are opportunities to explore topics in detail, but are problematic if the required resources (e.g., measurement equipment, DSP board) are not available. Restricting laboratory exercises to those that can be accomplished using supplied data and MAT-LAB avoids this problem. In this case, laboratory exercises might be grouped collections of exercises with a common or developing theme.

^{8.} See, for example, [6], a table of symbols used in [7].

4.6 Comparison with Grover & Deller

Foremost among similarities between the present work and Grover & Deller [7], the subject matter of the earlier book overlaps considerably with the topics covered in the outline offered here. For example, basic continuous- and discrete-time linear signals and systems, filtering, linear digital filters, correlation, and frequency analysis are all topics included in [7]. Hence, the overall structure of several sections of the outline presented here closely follows that of the earlier book.

The earlier book places few requirements on the reader's background, and assumes that the reader wishes to invest a minimum of time and effort into topics not directly related to DSP. The former assumption led to the inclusion of a subset of linear signals and systems material, a review of basic complex mathematics, trigonometry, and circuits, and a discussion of analog filters. The latter assumption led to the exclusion of powerful (but theoretical) tools such as the Laplace transform (at least in their explicit, mathematical formulation). These same issues appear in the present work.

There are several differences, however, between [7] and the current work. The goals of [7] are very explicit in addressing the needs of the reader to be able to efficiently implement basic DSP on small processors. The potential readers of the current work have a different and more varied set of goals outlined earlier in this paper, many shaped by the special characteristics of biomedical DSP.

Other differences include the organization of the presentation. The current work reduces what is three chapters in [7] ("Analog Signals and Systems," "Analog Filters," and "Discrete-Time Signals and Systems") to one (Chapter 2, Describing Signals). A single chapter on linear digital filters covers what was two separate chapters in [7], one each on the FIR and IIR filters, and includes some material on filter specification that had previously been discussed in terms of analog filters. In part, this reduction is the result of eliminating topics that can be addressed in "cookbook" fashion (e.g., in MATLAB). In other cases, topics were not included when they offered alternative paths to the same ends.

For example, FIR filter design by the Parks-McClellan program is given almost exclusive attention; windowing receives but a brief mention.

SUMMARY AND CONCLUSIONS

In this paper the results of a survey of biomedical DSP courses were presented. This survey showed that DSP is taught in a wide variety of settings within BME. Students' background can be diverse and may not include traditional prerequisites for DSP. DSP is often taught as a tool, or a course may offer only an overview of selected DSP topics. In other cases, BME courses may teach DSP topics rigorously as a foundation for further study.

Issues that differentiate biomedical DSP courses from EE-based DSP courses were then discussed. The nature of biosignals, the need for problem-solving in biomedicine, background and goals of students, and the purpose of processing the biosignals all contribute to making biomedical DSP unique. Useful pedagogical methods were also noted, including laboratories and projects, use of MATLAB, processing real signals, processing images, acquiring real signals, and designing, constructing, and/or using real hardware.

The focus of this thesis has been on creating the outline of a text appropriate for a particular biomedical DSP course. The audience chosen is that of BME students without a linear systems and signals background who want a broad exposure to DSP topics relevant to BME. The goals of the text are that the student be able to use a number of fundamental DSP techniques, including some that are not developed rigorously, such as simple detection and estimation. Image processing and compression are also covered, again with the goal of introducing some of the simple techniques.

The creation of an outline is very different from writing an actual text, and any realization of the proposed text would almost surely vary from this outline. However, the author's experience has been that a detailed outline is a necessary first step. This outline is useful in describing one particular route to addressing the needs of one particular group of BME students.

APPENDICES

APPENDIX A: OUTLINE FOR A TEXT

The following is an annotated outline for a text as described in the body of this paper.

Front matter		Trademarks, disclaimers, safety	
0 Preface		What is being said; who we think we're say- ing it to; how we're going to say it	
0.1 C	ontent		
0.1.1	Why important		
0.1.2	Introduction to subject		
0.1.3	Processing signals digitally	Figure: Before and after filtering (60 Hz noise on ECG)	
0.1.4	Images	Figure: Image sharpening or despeckling of medical image	
0.2 A	udience	People in biomedical engineering who need an overview understanding of DSP, but who lack a traditional electrical engineering background (i.e., linear signals and systems)	
0.2.1	Background by subject area		
0.3 Approach			
0.3.1	Minimal mathematics	However, point out that complex mathemat- ics will be used (and briefly reviewed).	
0.3.2	The limits this imposes on material	Without a more mathematical approach, many connections will not be clear, and that some flexibility is given up by relying on "cookbook" approaches.	
0.3.3	Use of MATLAB	Why MATLAB (powerful, built-in func- tions for DSP, widely used).	
0.3.4	The need for additional back- ground for further study	This presentation will include only the ba- sics; good engineering will probably require deeper understanding, <i>especially if applied</i> to a life-critical application.	
0.4 A	cknowledgements		
1 Intro	oduction		
1.1 B	iomedical signals		
1.1.1	Philosophical	What is measured vs. information sought- i.e., the signal we get is often related only empirically to the information we are seek- ing. Example: EEG, ECG, MR	

1.2 No	eed for processing signals	
1.2.1	Conditioning (amplitude, impedance)	Example/Figure: ECG amplifier
1.2.2	Noise	Example/Figure: ECG signal from above w/ 60 Hz noise) Note: Sometimes actually want to detect the 60 Hz from line/environment so we can syn- chronize to it (or subtract it accurately).
1.2.2	.1 Filtering	In-band vs. out of band noise, notch filters.
1.2.3	Compensation	Example: Frequency response
1.2.4	Linearization	Example/Figure: thermistor
1.2.5	Time scale	The time scale of biomedical signals is sometimes not convenient for our direct use.
1.2.5	.1 Fast, slow	Example: Temperature via strip chart re- corder
1.2.5	.2 Non-repetitive	Transient (freeze time) Example: Eye cycads (sp?)
1.2.6	Image "reconstruction"	Example: From projections Figure: MRI, ultrasound Compare with handheld image scanners
1.2.7	Interpretation (incl. freq domain)	
1.3 Si	gnal processing methods	
1.3.1	Analog	Example/Figure: RC filter
1.3.1	.1 Advantages	
1.3	.1.1.1 Necessary for condition- ing	
1.3	.1.1.2 Often cheap	
1.3	.1.1.3 Fast, high-frequency	Compare with digital bandwidths
1.3	.1.1.4 Well understood, widely known	Reference: [8]
1.3	.1.1.5 Interface to existing equipment	Example: Modular electronic test equipment
1.3.1	.2 Disadvantages	
1.3	.1.2.1 Initial precision, drift	Example: R and C variations from ideal
1.3	.1.2.2 Size	
1.3	.1.2.3 Cost	
1.3	.1.2.4 Power	

1.3.1.2.5	Limits to types of process- ing	Analog electronics is limited by physics. Some physical phenemena are useful in pro- cessing signals (such as integrating or differ- entiating voltage signals using capacitors, inductors, and resistors). Not every useful operation can be (easily) done using these devices.
1.3.1.2.6	Fixed (often) parameters	
1.3.2 Digit	al	Example: CD player, digital hearing aids, cardiological instruments, etc.
1.3.2.1 Be	enefits	
1.3.2.1.1	Precision, lack of drift	
1.3.2.1.2	Compression	Store digitally, eliminate noise, efficient storage/transmission, arbitrary SNR (non- realtime, comment on bandwidth of chan- nels), lossless/lossy, multi-function channels (e.g., Internet) Example: FBI fingerprint compression
1.3.2.1.3	Superset of analog func- tions	
1.3.2.1.4	Apply mathematics to sig- nal	Example: Geometric operations, such as in constructed images
1.3.2.1.5	Change parameters	Example: Adaptive filters in modems
1.3.2.1.6	Simple algorithms	Figure: Listing of core of digital filter
1.3.2.1.7	Share hardware with other control tasks	
1.3.2.1.8	Design tools available for DSP	
1.3.2.1.9	Ability to process in batch mode	I.e., non-realtime (including faster than real- time)
1.3.2.1.9	0.1 Time scale of change vs. time scale of parameter of interest	Example: Can capture a signal, then spend longer than that time processing it if the pa- rameter of interest can be detected from a short set of samples, but changes slowly compared to the amount of time necessary to process the signal.
1.3.2.2 D	isadvantages	Note that all of the disadvantages below have analogues with using microprocessors in digital control applications.
1.3.2.2.1	Theory not always acces- sible	

1.3.2.2.2 Base price is expensive

1.3.2.2.3 Inherently slower than analog	
1.3.2.2.4 Finite precision	Finite word length
1.3.2.2.5 Multiple technologies	Hence, multiple knowledge areas Example: Signals/systems, programming, analog hardware, digital hardware
2 Describing signals	Images are discussed in a later chapterjust a brief mention here as spatial vs. temporal, and additional dimension(s).
2.1 Time domain representations	Example/Figure: ECG, speech
2.1.1 Introduction to MATLAB	Why MATLAB is used, what it can do MATLAB/Figure: Start up MATLAB; add two numbers; generate a vector; print vector; plot vector; vector operations; getting help; some functions used; adding scales, grid, ti- tles to plots
2.2 Frequency domain representa- tions	Example: EEG, speech
2.2.1 How to think about spectra	Two approachesanalysis and synthesis. The first is important in explaining how one could generate a frequency domain repre- sentation, but the second is important in "proving" the completeness of that represen- tation.
2.2.1.1 Tuned filters (magnitude)	Might mention radio receiver (analog tun- ing!) Figure: Tunable filter & AC voltmeter
2.2.1.2 Synthesis (from sinusoids)	Figure/MATLAB: Square wave for sinuso- ids Forward reference: other basis functions
2.2.2 Using the FFT	FFT as discretly-tunable filter Forward reference: correlation MATLAB/Figure: FFT as tool to produce spectra; reading data files; read speech file and produce spectra (long enough to use rect window) Forward reference: abs() and log10() (dB)
2.2.3 Magnitude	MATLAB/Figure: FFT of large and small sinusoids at different frequencies
2.2.4 Phase, time delay	MATLAB/Figure: Phase of sine, cosine Figure: speech signal, random phase Forward reference: inverse FFT
2.2.4.1 Equation	

2.2.5	Completeness & information	Inverse transformation Real signals, why magnitude + phase is not twice as much information Forward reference: FFT discussion (why the input to FFT routines is often complex)
2.2.6 F d	iltering in the frequency omain	MATLAB/Figure: Simple filtering by atten- uating magnitudes before IFFT Forward reference: Chapter on digital filters (i.e., comment on efficiency of filter imple- mentation)
2.2.7 L	ogarithmic scales	Figure: Linear/linear, linear/log, log/log scales (frequency/magnitude)
2.2.8 R	ad/sec vs. Hz	Equation MATLAB: Plot using rad/sec
2.3 Disc	erete-time	Forward reference: Digital (and occasional ambiguity of "digital" vs. discrete-time)
2.3.1 I	ntroduction	
2.3.1.1	Sampled CT signals (peri- odic measurements)	Figure: Plot of DT signal, dashed CT
2.3.1.2	Phenomena that are dis- crete by nature	Example: Number of people (counts)
2.3.2 A	liasing	A fundamental question: how fast should one sample? Upper and lower limits. Example: Sampling breathing rate (approxi- mately periodic signal where aliasing can be demonstrated)
2.3.2.1	Nyquist rate, frequency	
2.3.2.2	Practical sampling rates	Sampling right at twice highest frequency Typical sampling rates (e.g., 5-10 times Nyquist frequency)
2.3.2.3	Phase and negative fre- quency	(Sidebar)
2.3.2.4	Undersampling on purpose	Figure: Realworld frequency to aliased fre- quency Equation
2.3.3 I	Digital versus discrete-time	Example: Money, EEG
2.3.3.1	Implications of digital pro- cessing	Values are approximated, but essentially in- finite SNR
2.3.3.2	Why design is of discrete- time systems instead of digi- tal	Discrete-time effects vs. quantization effects
2.3.3.3	Quantization noise	Table: Noise vs. bits of resolution
2.3.4	Juantizing	

2.3.4.1 ADC	
2.3.4.1.1 Simple examples	Figure: Flash converter
2.3.4.1.2 Anti-aliasing	Figure: Typical filter/ADC
2.3.4.1.3 Real-world ADCs	Parameters, types, interfaces Table/Figure: Typical converter speed/reso- lution/type (see Figure 4-26 in [7])
2.3.4.2 DAC	
2.3.4.2.1 Simple examples	Figure: Simple DAC
2.3.4.2.2 Anti-imaging	Figure: Magnitude spectrum of output, out- put with anti-imaging filter Figure: Time-domain output, with filtering (see Figures 4-30, 4-31, 4-32 in [7])
2.3.4.2.3 $Sin(x)/x$	Figure: Magnitude spectrum with and with- out sin(x)/x correction
2.3.4.2.4 Real-world DACs	Parameters, types, interfaces Combination ADC/DAC Typical values for biomedical applications
2.3.5 Discrete frequency	Forward reference: Section on DFT
2.3.6 Normalized frequency vs. real- world frequency	Equation
2.3.6.1 Notation	Ω vs. ω, F vs. f
3 Linear digital filters	Why linear, importance Overview of two types (FIR, IIR), catego- rized by presence of feedback
3.1 Averaging -> difference equations	Use a simple averaging example (e.g., air temperature, take mean of last 4 observa- tions) to get to difference equations. Figure: Signal with noise, result after aver- aging
3.2 MATLAB filtering evaluation	Figure/MATLAB: Apply simple averaging filter (from difference equation) to above signal
3.3 Finding coefficients (overview/ motivation)	Figure/MATLAB: Modify filter coefficients, observe magnitude spectrum
3.4 Specifying filters	
3.4.1 Magnitude, phase response	
3.4.1.1 Low/band/high-pass, etc.	Figure: LP, HP, BP responses
3.4.1.2 Using decibels	
3.4.1.3 Filter specifications	Figure: LP filter labeled with parameters be- low
3.4.1.3.1 Passband cutoff / edge	

3.4.1.3.2 Passband ripple	Caution: Differences in specifying ripple pa- rameters between "analog" and "digital" fil- ters (passive vs. active)see section 3.5.2.2 in [7]. Also, 3 different ways of specifying the rip- ple!
3.4.1.3.3 Stopband cutoff / edge	
3.4.1.3.4 Stopband ripple	
3.4.1.3.5 Center frequency	
3.4.1.3.6 Bandwidth / Q	
3.4.1.4 Phase	Causality, time delay
3.4.1.4.1 Linear phase	Figure: Linear phase response Figure: Time domain linear phase, nonlinear phase
3.4.2 Time-domain issues	Note importance in applications where time- domain features are traditionally (and/or em- pirically) emphasizede.g., ECG
3.4.2.1 Impulse, step response	Figure: Impulse and step response of a filter
3.4.2.2 Time-domain measurements	Figure: Step response labeled Figure: ECG with "good," "bad" filter
3.5 FIR filters	Table: Characteristics Forward reference: Comparison table of FIR vs IIR Figure: FIR diagram
3.5.1 Parks-McClellan (Optimal Equiripple)	
3.5.1.1 Example	MATLAB: FIR filter design (i.e., remez) Figure: Noisy signal input/output
3.5.1.2 Specifying parameters	
3.5.1.2.1 Determining order n	Equation MATLAB: use remezord
3.5.1.3 Linear phase	MATLAB/Figure: Phase response plot (use freqz)
3.5.1.4 Advanced use	MATLAB/Figure: Multi-passband or non- uniform passband (sloped passband) exam- ple
3.5.2 Simple integer (frequency-sam- pling) FIR filters	Equivalence to efficient averaging algorithm (for lowpass filter) Limitations MATLAB/Figure: Simple integer filter

3.5.3 Windowing	Brief discussion onlynot emphasized. (P- M receives more attention in this text) MATLAB: FIR filter by windowing Forward reference: Table of windows in DFT section
3.5.4 Frequency response from coef- ficients	Discuss simplistic (inefficient) approach
3.5.4.1 H(ej ^ω)	
3.5.4.2 FFT	MATLAB/Figure: freqz function example
3.5.5 Implementing	
3.5.5.1 Direct structure	Saving data movement, saving multiplies (linear phase structure)
3.5.5.2 Fast convolution	Calculations, appropriate situations
3.5.5.3 MAC instruction	(sidebar)
3.5.6 Summary of FIR design meth- ods	Table: (see Table 5-5 in [7])
3.5.7 Summary steps in FIR filter design	Figure: (see section 5.3 of [7])
3.6 IIR filters	
3.6.1 IIR characteristics	Can view as superset of FIRaddition of feedback. Other issues: Stability (feedback), phase, faster, complexity of implementation vs. FIR multiply-accumulate
3.6.2 Design process options	Indirect, Ad Hoc, time-domain, and frequen- cy-domain (iterative) methods.
3.6.3 Analog prototypes	Overview of indirect design method.
3.6.3.1 Frequency response func- tion H(jΩ)	
3.6.3.2 Analog prototypes	Optimal for different parameters (ripple, sharpness of transition, phase)
3.6.3.2.1 Simple table of filter types	See Table 6-1 of [7].
3.6.4 Converting from "analog" to "digital" filters	
3.6.4.1 BZT	Mention frequency warping, but use MAT- LAB function bilinear and/or other functions with built-in frequency warping.
3.6.5 Implementing	Problems/issues directly implementing the IIR difference equation.
3.6.5.1 FOS, SOSs	

3	.6.5.2	Forms and structures	Direct form I, II (canonic); parallel and cas- cade structures.
3	.6.5.3	Notes on converting lowpass to band/high pass	See Tables 6-4 and 6-5 of [7].
3.7	FIR	vs. IIR filters	Comparisons Table: Advantages, disadvantages (see [7]) Usual criteria for selection
3.8	Oth	er filters	Time-domain, empirical Forward reference: Image processing
4 F F	reque ourie	ency Analysis and the Fast r Transform	Stress importance, especially of FFT, but also wavelets. Application to many fields
4.1	Intr	oduction	
4.1	.1 E	example of FFT results	MATLAB/Figure: Speech data (steady-state voiced)
4.1	. 2 I	nterpretation	DFT "assumes" input is periodic (forward reference to windowing), output is discrete frequency magnitude & phase. Can synthe- size signal from this description (i.e., it is complete).
4.1	l.3 V	Vider view of spectral analysis	Draw attention to assumptions: stationary input, linear frequency resolution output. Decomposition into orthogonal bases (con- vey intuitively, forward reference to wave- lets).
4.2	The	DFT	
4.2	2.1 I	ntuitive operation	Assumption input signal is stationary and periodic with period a multiple of window. What discrete frequencies are present (and what phase), by correlation. (Correlation de- scribed as a measure of similarity between signals.)
4	.2.1.1	Example of sinusoids corre- lated	
4	.2.1.2	Magnitude and phase	
4.2	2.2 I	nterpretation of output	
4	.2.2.1	Complex	Use sine & cosine to show one needs a mag- nitude (not real part of complex number) and phase.

Note exponential and trigonometric representations used for DFT and complex numbers in general. Euler relates the two.

4.2.2.2	2 Magnitude	Discrete frequency, equation to relate to re- alworld frequency. Forward reference to res- olution in context of "what frequencies correspond to what numbers."
4.2.2.3	3 Squared, power spectral density	Advantage in that power is often the mea- sure of interest; also eliminates need to take square root in finding magnitude from com- plex numbers in rectangular coordinate sys- tem.
4.2.3	Leakage and resolution	(No mention of DTFT.) Windows intro- duced as consequence of DFT interpreting input as periodic, so need to "smooth" the ends of the input.
4.2.3.	1 Window shape: spectral leakage	Side lobe magnitude. Example: Compare rectangle to Blackman windowing of single frequency input.
4.2.3	3.1.1 Window types (table)	See Table 8-1, Figure 8-6 of [7].
4.2.3.	2 Window length: frequency resolution	Main lobe width. Example: Compare rectangle to Blackman windowing of two nearby frequencies.
4.2.3	3.2.1 Zero padding	Apply after windowing! Interpretation as increasing sampling of ex- isting spectrum. If highest "sampling rate" is not enough, must increase length of data.
4.2.3	3.2.2 Adding data	For summary of window length and pad- ding, see Figure 8-15 of [7].
4.3 Th	ie FFT	
4.3.1	Need (computational)	Example: DFT computational requirements for 1024-point.
4.3.2	Relationship to DFT	Except for numerical issues, same result, computionally efficient, family of algo- rithms, limitations on input length (e.g., power of two).
4.3.3	How the FFT works	Divide and conquer strategy; quick deriva- tion of $(N/2)log_2N$ (hence power of two re- quirement of input length), decimation-in- time (DIT), decimation-in-frequency (DIF), butterfly. MATLAB: Verify approximate number of FLOPS for a 1024-point FFT.
4.3.4	Input/output formats	Bit-reversed input (DIT), output (DIF); for- ward reference to complex input.

4.4 Inverse DFT/FFT	Show DFT and IDFT equations; scaling and sign are only difference, can use same code (with a flag) for both operations. Discuss "fast convolution"now have framework to discuss FFT/IFFT alternative to time-domain convolution for filtering.
4.4.1 Complex inputs for FFT/IFFT	Complex signals; IFFT input; the real-input FFT
4.5 Time-frequency analysis	The assumption of stationarity in the DFT; conditions when this is justified, and when it is not. Example: Speech, can be treated as station- ary over 5-20+ msec.
4.5.1 Stationary & non-stationary signals	Example: Speech, EEG. Figure: Overlapping FFT of speech sample
4.5.2 Multiple, overlapping FFTs (time vs. frequency resolution)	An informal appeal to Heisenbergi.e., res- olution in frequency vs. localization in time. Example: Swept sinusoid.
4.5.3 Wavelets	Example: Swept sinusoid. (Compare with above)
4.6 Other transforms and techniques	Why other transforms: computational effi- ciency, good match to digital processors, better match to the signal (i.e., better model of underlying stochastic process), time-fre- quency issues, selective or non-linear fre- quency resolution.
4.6.1 Wavelets	Table, comparison to FFT. Note computational requirements and result- ing information (i.e., the Wavelet transform is usually faster because the common imple- mentation does not produce linear frequency resolution).
4.6.2 Multiple filters	Example: Displays for stereo equalizers
4.6.3 Goertzel	A special, limited case of the DFT (calcu- lates subset of outputs, efficient IIR imple- mentation)
5 Correlation and Signal Averaging	The goals are to detect presence of known signal or estimate some parameter of a known signal Example: Time delay Example: Heart beat modulating micro- waves (e.g., under building rubble)
5.1 Crosscorrelation	Outline basic idea of comparing two signals- one known. Reference to correlation in the DFT. Two applications discussed:

5.1.1 Detecting a known signal in additive noise	Example: ECG and template Figure/MATLAB: See Figure 9-3 of [7], substitute ECG and template for sinusoid (non-optimum heart beat detector).
5.1.2 Estimating time delay	Example: Echo-based imaging (e.g., ultra- sound) Figure/MATLAB: Received signal (w/ noise) correlated with transmitted signal
5.1.3 Pseudo-Random Noise	Brief discussion of need for signals with nice correlation properties (i.e., "random"), out- line some simple schemes for same (e.g., lin- ear feedback shift register).
5.2 Autocorrelation	I.e., compare one section of a signal with an- other section of same signal. Assumption: noise is uncorrelated (or low energy)
5.2.1 Estimating periodicity of signal	Example: Speech, pitch detection (non-opti- mum)
5.2.2 Energy	Equivalence of autocorrelation with lag=0 and equation for signal energy (carefulnote that some finer points are glossed over with respect to definition of power and energy signals for finite signals, etc.)
5.3 Signal averaging	Give simple example, point out assumption made about noise (uncorrelated).
5.3.1 High-resolution ECG	Note need for finding robust fiducial points. (See [3].) Figure/MATLAB: Add noise to long ECG sequence, use correlation to align for averag- ing.
6 Image processing	
6.1 Why digital image processing?	
6.1.1 Enhance established practice	Example: Enhance x-rays (reduce noise, in- crease constrast, etc.)
6.1.2 New technology	DSP enables new tools, not possible with old technology Example: CTUsing mathematics to assem- ble 3-d picture vs. invasive surgery.
6.1.3 Analyze existing images	Can perform exacting analysis without fa- tigue-induced error. Example: Automated detection of malignan- cies
6.2 Creating a digital image	

6.2.1 Sampling in two dimensions	Samplinginherent in detector, or in sam- pling rate and line spacing of "serialized" image (e.g., video signal). Example/Figure: Long, fairly narrow image, with increasing spatial frequency, shows spatial aliasing. Or, make it square and show aliasing in both dimensions.
6.2.2 Quantizing	Light intensity to numeric value. Note that typically detectors do not output in intensity/hue; typical is use of monochrome sensor or multiple sensors in (roughly) or- thogonal hues.
6.2.3 Color, monochrome	
6.2.3.1 Comments on human visual system (properties)	Logarithmic; use of color, intensity, "trick- ing" the eye using RGB. (See, for example, [12])
6.2.4 Source, application	An attempt to categorize computed imaging by method and imaging (summarize in a ta- ble)
6.2.4.1 Grouping by method	
6.2.4.1.1 Shadows	Example: X-ray
6.2.4.1.2 Reflections	Example: Ultrasound, MR
6.2.4.1.3 Emissions	Example: Nuclear medicine, thermal
6.2.4.2 Grouping by image	
6.2.4.2.1 Direct imaging	Example: X-ray, some nuclear
6.2.4.2.2 Computed imaging	Example: Ultrasound, MR, PET, SPECT, x-ray CT
6.3 Enhancement	
6.3.1 Operations on a single point	Note the phenomenon of x-ray absorption in bones, the sensitivity of x-ray detectors or film to x-rays, etc., are non-linear and not necessarily optimum for interpretation by humans. Single-point operations can make this information more visible by matching our sensitivity better.
6.3.1.1 Color/shading modification	
6.3.1.1.1 contrast/false color	Example: Give raw data of x-ray with bad contrast, and simple operation to map (non- linearly) pixel values. Note that single-point processing is in some sense analogous to linearization of (1-d) sig- nals, though in that case the mapping is not arbitrary.

6.3.2 Area operations

6.3.2.1 Spatial filtering	Example/Figure: Use a separable 2-d filter to reduce the noise in an image. (Must justify separable property, but it pays off in allowing existing 1-d filtering to be used.) Notecomment that images can have arbi- trarily "sharp" edges, in contrast to many other signals that are limited in bandwidth by the underlying physics involved. (Has implications for efficient decomposition e.g., wavelets.)
6.3.2.2 Edge enhancement etc	Example: High-pass filter. Spatial-domain techniques.
6.4 Analysis	Just an overviewrecognition, detection. Example: Edge detector
6.5 Reconstruction from projections	This, with the spatial filters above, is a major focusmotivating, relatively simple, basis for important new technologies.
6.5.1 Reconstruction by back projec- tion	Start with a square matrix example (2 pro- jections), then generalize.
6.5.1.1 Filtering the resulting image	Introduce need to apply a filtereither filter- ing the image, or projection
6.5.1.2 Filtering the projections	
7 Compression	
7.1 Motivation	
7.1.1 Storage	Show some calculations for typical images (more dramatic than 1-d signals like ECG) Example: ECG, images
7.1.2 Communications	Arbitrary SNR, all digital communication/ storage, as part of efficient communication Historic need to reduce bandwidth. Example: Data over phone lines, via modem
7.2 Lossless, lossy	When information can be thrown away, presence of noise, optimizing based on hu- man perception
7.3 Common approach to compres- sion	Compression often a two-part processex- pression in an appropriate (orthogonal) ba- sis, then encoding. (Sinusoids are not the best basis for everything!). See article [4].
7.4 Examples	
7.4.1 Run length encoding	Exercise: Write simple RLL routine
7.4.2 Spatial frequency techniques	Example/Figure: FBI wavelet
7.4.3 Empirical time-domain	Example: ECG compression

8	Afterword	Additional comments about areas that were not covered, now that the reader has a frame- work.
9	Bibliography	Also includes web resources, such as the SPIB database, MIT-BIH, etc.
10	Glossary	With page numbers
11	Index	
12	2 Table of symbols	Table of symbols used in the text, with brief description (and name, for Greek and other unusual symbols) and page number where defined.
13	CDROM	A CDROM should be included, which would allow a huge amount of data and pro- grams to be included.
1	13.1 Contents	
	13.1.1 M-file examples	As with the prior book, MATLAB m-files for the illustrations and examples would be useful to include.
	13.1.2 Example biomedical signals	Notepermission has already been granted by MIT-BIH to include data (e.g., ECG) from their database [10], with appropriate credit.
	13.1.2.1 EEG	
	13.1.2.2 ECG	
	13.1.2.3 EMG	
	13.1.2.4 Physiological (biomechan- ics)	
	13.1.2.5 Speech	Some example speech, perhaps from SPIB [11]
	13.1.2.6 Images	
	13.1.3 Description of format of signals	MATLAB compatible, but also provide de- tails to allow use on multiple platforms and with other or custom software.
	13.1.4 1.44M floppy disk image of minimal data set	Just a subdirectory, would allow instructor or student to create a floppy disk with a sub- set of data useful in doing exercises.

APPENDIX B: EXAMPLE EXERCISES

Below are examples of the type of exercises that would accompany the text outlined in this paper. The exercises are not necessarily in the order they would occur in the text. In some cases explanatory comments have been added to the start of the exercise, though this material would normally occur in the text itself.

Exercise 1 MATLAB can plot vectors (lists of discrete numbers), but cannot directly plot continuous-time functions. Instead, we choose enough discrete points to plot that give the appearance of a smooth function. To plot a few cycles of a cosine wave of amplitude 4, frequency 2 Hz (i.e., $4cos(2\pi t^2)$, as shown in Figure 1, we could use



Figure 1 Exercise 1, 2 cycles of a cosine wave

```
start_time=0;
end_time=1; % 1 sec
points=200;
t=start_time: (end_time-start_time)/points: end_time;
plot( t, 4*cos(2*pi*t*2));
```

Note that \mathbf{t} is a vector of times, equally spaced so that approximately 200 points are included. (The number 200 is arbitrary--lower numbers are less smooth, higher numbers take more memory and in other contexts may take longer to process.)

Generate a plot using MATLAB for the continuous-time signals described below. Use enough points to produce a relatively smooth graph, and use a time scale that shows all relevant features. Be sure to label the axes. Use an amplitude of one and a phase of zero unless specified. To save paper, you can use the **subplot()** function in MATLAB.

a) 1 KHz cosine; b) 1 MHz sine; c) 17 Hz cosine + 60 Hz sine; d) 17 Hz cosine

(amplitude 0.2) + 60 Hz sine; e) 20 Hz sine + 20 Hz cosine; f) 377 rad/s sin, phase $\pi/4$; g)

10 Hz squarewave times a 60 Hz cosine; h) 35 Hz cosine (amplitude 0.4, phase 90°).

Answer:

subplot(4,1,1); % 4 subplots, horizontal % Pick, say, 20x for plotting, 5 waveforms m = 20;cyc=5; % number of cycles to plot % a) 1 KHz cosine f=1000; T=1/f; % frequency & period t=0:T/m:(cyc*T-T/m); % time vector % time 0 to time at end of `cyc' cycles, % incrementing T/m (sample period) $y = 1 * \cos(2 * pi * f * t);$ subplot(4,1,1); plot(t*1000,y); grid on; xlabel('Time (msec)'); ylabel('Amplitude'); % b) 1 MHz sine f=1e6; T=1/f; % frequency & period t=0:T/m:(cyc*T-T/m); % time vector y = 1 * sin(2 * pi * f * t);subplot(4,1,2); plot(t*1e6,y); grid on; xlabel('Time (usec)'); ylabel('Amplitude'); % c) 17 Hz cosine + 60 Hz sine; m=5; cyc=20; % revise to get better picture f=60; T=1/f; % frequency & period t=0:T/m:(cyc*T-T/m); % time vector y = cos(2*pi*17*t) + sin(2*pi*60*t);subplot(4,1,3); plot(t*1000,y); grid on; xlabel('Time (msec)'); ylabel('Amplitude'); % d) 17 Hz cosine (amplitude 0.2) + 60 Hz sine; m=5; cyc=20; % revise to get better picture f=60; T=1/f; % frequency & period t=0:T/m:(cyc*T-T/m); % time vector $y = 0.2 \cos(2 \sin 17 t) + \sin(2 \sin 60 t);$ subplot(4,1,4); plot(t*1000,y); grid on; xlabel('Time (msec)'); ylabel('Amplitude'); m=20; cyc=5; % restore

% next figure figure; subplot(4,1,1); % 4 subplots, horizontal % e) 20 Hz sine + 20 Hz cosine f=20; T=1/f; % frequency & period t=0:T/m:(cyc*T-T/m); % time vector y = 1*sin(2*pi*f*t) + 1*cos(2*pi*f*t);subplot(4,1,1); plot(t*1000,y); grid on; xlabel('Time (msec)'); ylabel('Amplitude'); f f) 377 rad/s sin, phase p/4; w=377; T=2*pi/w; % frequency & period t=0:T/m:(cyc*T-T/m); % time vector y = 1 * sin(w * t + pi/4);subplot(4,1,2); plot(t*1000,y); grid on; xlabel('Time (msec)'); ylabel('Amplitude'); % g) 10 Hz pulse times a 60 Hz cosine; m=20; cyc=10; % revise to get better picture f=60; T=1/f; % frequency & period t=0:T/m:(cyc*T-T/m); % time vector $y = ((\cos(2*pi*10*t))>0) .* \cos(2*pi*60*t);$ % definition of phase wasn't specified for squarewave subplot(4,1,3); plot(t*1000,y); grid on; xlabel('Time (msec)'); ylabel('Amplitude'); m=20; cyc=5; % restore % h) 35 Hz cosine (amplitude 0.4, phase 90°). f=35; T=1/f; % frequency & period

t=0:T/m:(cyc*T-T/m); % time vector y = 0.4*cos(2*pi*f*t+pi/2); subplot(4,1,4); plot(t*1000,y); grid on; xlabel('Time (msec)'); ylabel('Amplitude');



Figure 2 Exercise 1, signals a) through d)



Figure 3 Exercise 1, signals e) through h)

Exercise 2 The Fourier transform (FT) and its inverse allow us to move between the time domain and frequency domain representations of continuous-time signals. We can use the discrete Fourier transform (DFT), usually available in a form known as the fast Fourier transform (FFT), to perform this transformation on discrete-time signals. MATLAB has this function available, as the following example shows. (See Figure 4.)

```
s_rate=200; % sample 200 points per second
t=0:1/s_rate:1; % 1 second
x=(sin(2*pi*t*10) >= 0)-0.5; % square wave, 10 Hz
Y=fft(x,512); % 512-point DFT
f=(0:255)*s_rate/512; % frequency scale
subplot(2,1,1);
plot(t,x);
subplot(2,1,2);
plot(f,abs(Y(1:256))); % plot magnitude of first half only
```



Figure 4 Exercise 2, FFT of a square wave

Review if necessary any MATLAB syntax that is unfamiliar, especially how ranges of numbers (e.g., **1:256**) and indexing into vectors works.

The FFT produces more information than we need, so notice that we only use the first half of the resulting vector. The **abs** function gives us the magnitude of the complex output of the FFT. Use the **angle** function to produce the phase. The **real** and **imag** functions return the real and imaginary parts of the complex output.

Replace the expression for \mathbf{x} in the listing above to generate a sine wave of amplitude 4 and frequency 47 Hz plus a cosine wave with amplitude 1 at 35 Hz. Produce a frequency-domain magnitude & phase plot. Plot the real and imaginary components. **Exercise 3** a) Use the IFFT to produce a time-domain signal using the FFT from Exercise 2. b) Now zero out the phase and perform the IFFT. c) Zero out the magnitude, keeping the phase. d) Make the phase random. e) Zero out the real part. f) Zero out the imaginary part. Compare the results of each of these manipulations with the original time-domain signal. **Exercise 4** Determine whether the human auditory system can differentiate between the signal reconstructed with unadjusted phase and random phase. (Hint: Use the "sound" function in MATLAB and the techniques of Exercise 3.)

Exercise 5 Calculate the phase in radians and degrees between the input and output of a "black box" that delays the input signal by 0.05 seconds, given the following input signal:

 $5 \cdot cos(2 \cdot \pi \cdot 2000)$

Exercise 6 The following commands in MATLAB will read in an ASCII data file consist-

ing of data samples:

```
% open and read data file
fid=fopen(`samples.dat','r');
x = fscanf(fid,'%d');
```

First, read in the data file v5a.dat, which contains a single channal of ECG data from the MIT-BIH database. The sampling rate is 360 Hz. Plot this signal, with the correct time along the x-axis. Add 60 Hz interference to the signal as follows:

```
n= cos(2*pi*t*60) .* 20; % 20 is arbitrary
xn=x + n;
```

where t is a vector containing the time, in seconds, of each sample, and x is the original data vector. Plot this signal. Now use the **remez** and **filter** functions in MAT-LAB to design a filter with the following parameters:

Stopband: 55-65 Hz

Deviation in the passbands: 0.01

Deviation in the stopband: 0.1

You may want to use the function **remezord** to estimate the order and calculate other parameters for the **remez** filter design function. Plot the filtered signal (you may want to omit the first 100 samples or so to avoid the transients caused by the filter starting up). Compare the filtered signal to the original signal.

Answer:



Figure 5 Exercise 6, original signal



Figure 6 Exercise 6, signal with 60 Hz noise





```
dname='v5a.dat';
s_rate=360; % sampling rate
% open and read data file
fid=fopen(dname,'r');
x = fscanf(fid,'%d');
x=x';
fprintf('file "%s" opened, %d elements
read\n',dname,length(x));
% plot it w/o any noise, just 2 seconds worth
dur=2; last=dur*s_rate;
```

```
x=x(1:last);
t=(0:(last-1))/s rate; % time in seconds
plot(t,x);
grid on;
% now add some 60 Hz interference
n= cos(2*pi*t*60) .* 20; % 20 is arbitrary
xn=x + n;
plot(t,xn);
grid on;
% now filter with FIR filter
F=[50 55 65 70]; % cutoff frequencies, Hz
A=[1 0 1]; % amplitudes per band
Dev=[0.01 0.1 0.01]; % deviation per band
[N,Fo,Ao,W]=remezord(F,A,Dev,s_rate); % calculate params
b=remez(N,Fo,Ao,W); % calculate filter coefficients
y=filter(b,[1],xn); % filter the signal
% throw out the first 100 samples and compare
subplot(2,1,1);
plot(t(100:length(t)), y(100:length(y)));
arid on;
subplot(2,1,2);
plot(t(100:length(t)),x(100:length(x)));
grid on;
```

Exercise 7 Filtering using FFT: Create waveform by adding sinusoid + noise. FFT. Zero out coefficients outside desired freq range. IFFT and plot. Repeat using sinusoid both inside and out of desired freq range. Repeat for highpass, bandpass.

Exercise 8 Complete Table 3. Unless otherwise specified, measurements are "rms." When

necessary, assume the loads for both measurement a and b are the same. If a quantity can

not be computed, state the reason.

a	b	dB
	1 V	90 dB
1 V		90 dB
1 V	0.001 V	
1 W	0.001 W	
1 V	1 V	
0 V	0.001 V	
0.70795 V	0 V	
	1 V	0 dB

Table 3	Exercise	8
---------	----------	---
a	b	dB
------------	-----------	--------
	1 V	-3 dB
	1 V	6 dB
	1 V	-10 dB
	1 V	20 dB
	1 W	0 dB
	1 W	-3 dB
	1 W	6 dB
	1 W	-10 dB
	1 W	20 dB
-0.70795 V	1 V	
1 V	1 W	
1 W	0.70795 V	

 Table 3 Exercise 8 (Continued)

Exercise 9 Complete Table 4.

Table 4 Exercise 9

	Radians	Radians		
Degrees	In terms of π	Decimal approx.		
0				
1				
15	π/12			
20				
30				
45				
	1	1		
60				
90				
	π	3.1416		
270				
360				
450				
720				
	7π			
-90				
-270				

Hertz	rad/s
0	
0.001	
60	
8 K	
10 K	
44.1 K	
	377
	1
	2 K
	1 M

Exercise 10 Complete Table 5, expressing the frequencies in Hertz and radians/sec.

Table 5 Exercise 10

Exercise 11 Categorize the following signals as analog, digital, discrete-time (/continous-value), discrete-value (/continuous-time). (If necessary, specify any conditions you feel are ambiguous.)

a) Hourly blood pressure readings (3-digit digital display)

b) Body core temperature of surgery patient

c) Fetal heart rate

d) Heart beats in 1 minute of a patient

e) Thermostat (old-style, bi-metalic strip bends to tip a mercury switch)

f) Frequency generated by a swallowable temperature monitor that varies fre-

quency with core body temperature

g) Sampled ECG signal stored on disk

Exercise 12 Give examples of analog, digital, and discrete-time signals that are distinct from any examples in Exercise 11.

Exercise 13 For the given sampling frequency f_s and input frequencies, find the aliased frequencies. (You may ignore the sign of the aliased frequencies, or specify negative frequencies, but be consistent.)

a) f_s=8 kHz. f={60 Hz, 1 kHz, 3.9 kHz, 4 kHz, 4.1 kHz, 7.9 kHz, 8 kHz, 8.1 kHz,

15.9 kHz, 16.1 kHz, 40 kHz, 870 kHz (AM radio station), 92.7 MHz (FM radio)}

b) f_s =100 Hz. f={10 Hz, 50 Hz, 60 Hz (AC line frequency), 120 Hz, 1.02 kHz}

c) f_s =60 Hz. f={10 Hz, 50 Hz, 60 Hz (AC line frequency), 120 Hz, 1.02 kHz}

Exercise 14 Using MATLAB, write a script that demonstrates the effect of sampling a sinusoid with a sampling frequency of 10f, 5f, 2f, 1f, 0.7f where f is the frequency of the sinusoid. (Hint: Use **subplot(5,1)** to place all the signals on one plot. Use **comb** to plot sampled points.)

Real-world frequency (f)	Sampling fre- quency (ω)	Normalized frequency (F)	Normalized frequency (Ω)
60 Hz	100 Hz		
60 Hz	200 Hz		
60 Hz	1 kHz		
10 Hz	100 Hz		
40 rad/s	100 rad/s		
1 Hz	1 Hz		
2π Hz	1 rad/s		
	100 Hz	0.2	
	8 kHz		0.1π rad

Exercise 15 Complete Table 6.

Table 6 Exercise 15

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] Biomedical Engineering Program Home Pages. [Online] Available http://fairway.ecn.purdue.edu/v1/bme/biomed-homes.html, April 13, 1999.
- [2] Burrus, C. Sidney, Ramesh A. Gopinath, and Haitao Guo, Introduction to Wavelets and Wavelet Transforms: A Primer. Upper Saddle River, NJ: Prentice-Hall, 1998.
- [3] Craelius, William, Mark Restivo, and Nabil El-Sherif, "Techniques for Processing Cardiac Signals: Fiducial Formulae for Fidelity," in *High-Resolution Electrocardiography*, Nabil El-Sherif and Gioia Turitto, Eds. Mount Kisco, NY: Futura, 1992, pp. 21-49.
- [4] Donoho, David L., Martin Vetterli, R. A. DeVore, and Ingrid Daubechies, "Data Compression and Harmonic Analysis," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2435-76, October 1998.
- [5] EAC Criteria for 1999. [Online] Available http://www.abet.org/eac/EAC_99-00_Criteria.htm, April 15, 1999.
- [6] Grover, Dale, The Cast of Characters. [Online] Available http://www.redcedar.com/ castofch.htm, April 17, 1999.
- [7] Grover, Dale, and John R. Deller, *Digital Signal Processing and the Microcontroller*. Upper Saddle River, NJ: Prentice-Hall, 1999.
- [8] Horowitz, Paul, and Winfield Hill, *The Art of Electronics* (2nd ed.). New York: Cambridge University Press, 1989.
- [9] Jain, Anil K., Fundamentals of Image Processing. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [10] MIT-BIH Database Distribution, [Online] Available http://ecg.mit.edu/, April 17, 1999.
- [11] Signal Processing Information Base (SPIB). [Online] Available http:// spib.rice.edu:80/spib.html, April 17, 1999.
- [12] Tufte, Edward, Envisioning Information. Cheshire, CT: Graphics Press, 1990.

