UNIVERSITY OF TEXAS AT EL PASO
EE4383/5371 SYLLABUS and OUTCOMES, FALL 2001

TITLE:	DIGITAL SIGNAL PROCESSING (DSP)	
INSTRUCTOR:	SERGIO D. CABRERA, Associate Professor	
OFFICE:	Eng. 339A, tel. 747-5470 (Dept.), 747-7871 (fax)	
OFFICE HRS:	Monday	1:30 – 2:30 PM
(tentative, when in town)	Tuesday	1:30 – 2:30 PM
	Wednesday	(after mid-October)
	Thursday	1:30 – 2:30 PM
	Friday	11:00 AM – 12:00 Noon
E-MAIL:	cabrera@ece.utep.edu (for important matters please)	
SECRETARY:	Ms. Marty Finney, (915)747-6003, E-mail: punky@eng.utep.edu	

TEXTS:
1- "**DIGITAL SIGNAL PROCESSING: Principles, Algorithms, and Applications**", Third Edition by John G. **Proakis** and Dimitris G. **Manolakis**, Prentice-Hall., 1996.
2- "**Computer-Based Exercises for Signal Processing Using MATLAB 5**", MATLAB Curriculum Series, by J. **McClellan** et. at., Prentice-Hall, 1998

PREREQUISITES: EE3353 (Signals and Systems): basics of discrete-time signals and LTI systems; basics of two-sided z-transform; continuous-time Fourier analysis; etc.. Students are expected to have basic proficiency in the use of the Matlab software package.

OVERVIEW OF THE COURSE: (course outcomes are listed separately)

Students will learn the *theory* behind the basic DSP *techniques* currently employed on one-dimensional *signals*, such as speech, audio, and DSP-based Telecomm systems. The emphasis will be on *digital filtering* and the *Discrete Fourier Transform (DFT)*. Student will obtain some experience with the software and hardware *implementation* of DSP techniques and to the procedures to carry out processing of stored signals and to do *real-time* processing.

Computer assignments using MATLAB and other software packages will facilitate getting experience in processing simulated and acquired signals. There will be some demos and assignment in the Real-Time DSP (RTDSP) Laboratory (Eng. 305 or 306) using DSP processor boards which have A/D and D/A capability as well a Texas Instruments TMS320C6000 DSP.

Students enrolled in EE5371 will be assigned different homework problems and computer assignments to guarantee a more in-depth coverage of DSP.

COMPUTER AND SOFTWARE FACILITIES:

MATLAB including the SIGNAL PROCESSING toolbox is available as follows:

1- The ECE Dept. PC/*Vectra* Lab.computer network (room E-314). Use the on-line documentation and run the demos to learn MATLAB.
2- The ECE Dept. *UNIX Lab.* in E-319 (Sun and Lynux systems)

3- You can purchase the STUDENT EDITION OF MATLAB (priced approx. at $150). through the Web site *http://www.mathworks.com/products/studentversion/index.shtml*

COURSE GRADING:

In-class Semester Exams:	45 %
Homeworks, Labs. and Computer Assignments	25 %
Final Exam	30 %

COURSE OUTCOMES (from Fall 2000) (Subject to changes during Fall 2001 in order to accommodate Laboratory assignments, see homework assignment handouts)

The outcomes are grouped by technical topic which is indicated with a roman numeral. Sections and Chapters correspond to the textbook by Proakis and Manolakis. The relevant ABET Outcome Criteria are:

A=Apply mathematics, science and engineering principles
B=Ability to design and conduct experiments and interpret data
C=Ability to design a system, component, or process to meet desired needs
K=Ability to use the techniques, skills and modern engineering tools necessary for engineering practive

The relevant ABET outcome for each course outcome below is indicated in parenthesis in *italics*.

I. Selective Review of Discrete-Time Signals and LTI Systems (see Chap. 2)

After completing this course, students will be able to:

- Compute the convolution between two discrete-time signals using time-domain or z-domain methods. *(A)*

II. Review of Continuous-Time and Discrete-Time Sinusoidal Signals and Discrete-Time Frequency. (Sect.1.3)

After completing this course, students will be able to:

- Analyze Discrete-Time sinusoidal signals, their properties, and their relationships to continuous-time sinusoids that have been sampled. *(A, B)*
- Generate, plot and manipulate samples of sinusoidal signals using the software package Matlab. *(K)*
- State and understand the implications of the sampling theorem for sinusoids and for bandlimited signals in terms of low-pass frequency contents. *(A)*

III. Review of Z-Domain D-T Signal and LTI Systems Analysis (parts of Chap. 3)

After completing this course, students will be able to:

- Find z-transforms and inverses for basic signals and for signals derived from them by basic manipulations using z-transform properties. *(A)*
- Determine the transfer function of a Linear Time-Invariant (LTI) system and relate its poles and zeros to other system properties such as stability and causality. *(A)*

IV. Discrete-Time Fourier Analysis of Signals and Systems. (parts of Chap. 4)

After completing this course, students will be able to:

- Find the Discrete-Time Fourier Transform (DTFT) or the inverse for basic signals and for signals derived from them by basic manipulations using DTFT properties. *(A)*
- Convert time-domain relationships between signals to corresponding frequency domain properties and vice-versa. *(A)*
- Determine the frequency response of an LTI system and categorize it in terms of type of frequency-selectivity. *(A)*
- Evaluate a rough sketch of a magnitude respone for a LTI system from the pole-zero configuration. *(A)*
- Determine the response of an LTI system with know frequency response to sinusoidal and other simple input signals whose DTFT can be determined. *(A)*
- Design by hand and with Matlab, simple low-order frequency selective filters, resonator, and notch filters by pole-zero placement. *(C, K)*

V. FIR and IIR Filter Structures. (Sect. 2.5, parts of Chap. 7)

After completing this course, students will be able to:

- Perform LTI D-T system (digital filter) analysis in the z-domain and the frequency-domain using Matlab for both Infinite Impulse Response (IIR) and Finite Impulse Response (FIR) filters. *(A, K)*
- Compute IIR and FIR filter responses using Matlab and compare results with expected results derived analytically. *(A, B, K)*
- Derive the direct form I and II, their transposed versions, and other standard block diagram realizations/structures for FIR and IIR digital filters. *(A)*
- Derive the cascade and parallel structures for IIR digital filters using second-order sections. *(A)*

- Understand and use the alternatives and tradeoffs in computational efficiency, coefficient quantization sensitivity, and round-off error propagation in the implementation of FIR and IIR filter structures. *(A, C)*

VI. Design of Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) Digital Filters. (Sects. 1.4.2, 4.2.9, and parts of Chap. 8)

After completing this course, students will be able to:

- Understand the linear-phase property of digital filters and the associated time-domain symmetries. *(A)*
- Be familiar with the definitions and properties of at least 4 standard discrete-time window functions. *(A)*
- Understand and use frequency-domain relationships between C-T and D-T signals obtained from each other by time-domain sampling (C to D) or by bandlimited reconstruction (D to C) . *(A)*
- Understand and convert standard digital filter design specification formats to and D-T and C-T frequency units. *(A)*
- Design by hand simple frequency selective FIR linear-phase filters using the window method. *(A, C)*
- Understand criterion and properties of optimum equiripple linear phase FIR filters. *(A)*
- Design and use of window-method and optimum equiripple FIR digital filters using Matlab. *(B, C, K)*
- Understand the process and the resulting filter properties for the *bilinear* transformation method of converting analog filters into digital filters. *(A)*
- Design analog Butterworth low-pass and high-pass filters from the specifications of the resulting digital filter assuming the bilinear transformation is used. *(A, C)*
- Understand time and frequency domain properties of resulting Butterworth, Chebyshev and Elliptic digital filters obtained with the bilinear transformation. *(A, C)*
- Understand the methods of transformation of filters from low-pass to high-pass, band-pass or band-stop filters and how they are implemented in Matlab. *(A, D)*
- Design and use frequency selective IIR digital filters obtained using the bilinear transformation method. *(B, C, K)*

VII. The Discrete Fourier Transform (DFT). (parts of Chap 5)

After completing this course, students will be able to:

- Find the Discrete Fourier Transform (DFT) for basic, Finite-Duration (F-D) signals directly, using properties of the DFT, or by sampling the DTFT. *(A)*

- Understand and perform circular independent variable manipulation operations on F-D signals. *(A)*
- Perform a time-domain circular convolution operation of two F-D signals. *(A)*
- Apply the DFT and circular convolution to perform linear convolution of F-D signals. *(A)*
- Use Matlab to compute DFTs and to perform circular convolutions using DFT multiplication. *(K)*
- Use the DFT to perform spectral analysis from finite data records, by hand and using Matlab, for sums of sinusoids. *(A, B, K)*
- Understand and use frequency and computational resolution bounds in spectrum analysis of sinudoids with and without windowing. *(B, C, K)*

VIII. Fast Fourier Transform (FFT) Algorithms for DFT computation. (parts of Chap. 6)

After completing this course, students will be able to:

- Calculate the arithmetic operations necessary to evaluate the DFT of a F-D signal. *(A)*
- Understand and use the procedures and flow-graphs that describe the Fast Fourier Transform (FFT) algorithms for radix–2, radix-4, and two-factor decompositions using decimation-based index mappings. *(A)*
- Understand the derivation and use of the FFT computational complexity tables in DFT computation and in linear filtering using the DFT. *(A, C)*
- Understand and use the modified FFT procedures for purely real signals. *(A, C)*
- Implement an FFT algorithm using the Matlab programming language. *(A, K)*

IX. Application of the DFT to Spectrum Analysis From Finite Data. (Chap. 5)

After completing this course, students will be able to:

- Understand the motivations behind the Short-Time Fourier Transform (STFT) definition and its use in the analysis of non-stationary signals in the time-frequency domain. *(A)*
- Use Matlab to compute and analyze the resulting STFT magnitude (the spectrogram) plots for real signals such as speech or audio using various window sizes and types and various DFT lengths. *(B, K)*