AC 2007-2807: INTERACTIVE LEARNING DISCRETE TIME SIGNALS AND SYSTEMS WITH MATLAB AND TI DSK6713 DSP KIT

Zekeriya Aliyazicioglu, California State Polytechnic University-Pomona

Saeed Monemi, California State Polytechnic University-Pomona

Tim Lin, California State Polytechnic University-Pomona

Interactive Learning Discrete Time Signals and Systems with MATLAB and TI DSK6713 DSP Kit

Zekeriya Aliyazicioglu, Tim Lin, Saeed Monemi
California State Polytechnic University, Pomona,CA

Abstract

Discrete time signal and systems courses are a good starting point to study in the electrical and computer engineering program. It is interesting because of the multimedia capability and the ability of the students to make something happen with audio signals. Also, discrete time signals and systems are used increasingly in a wide spectrum of applications, such as; instrumentation, telecommunications, medical, automotive, control, graphics/imaging, military, consumer electronics, industrial, voice/speech etc. This will help students get an idea on how and where they can use it. For that reason it should be introduced to students early because it would help in recruitment and retention of electrical and computer engineering students. To motivate the beginning engineers to the hard work of connecting mathematics and computation, we can teach discrete time signal and systems interactively and visually using some computer tools.

In this paper, we present a suite of interactive discrete time signal and systems demonstration modules using MATLAB, Simulink and TI DSK6713 DSP kit. Using some practical applications and simulation, we can make the class more interesting and interactive. Most demos use real-world signals such as speech and music so that the student can appreciate the concepts better. We focus on providing visualization tools that accentuate the intuitive aspects of discrete time signal and systems algorithms. Our goal is to design intuitive and flexible tools that the students could use to experiment freely with signals and algorithms, without getting overly involved in programming. This will guide the students through basic concepts of signal representations, sampling, quantizing, coding, frequency domain representation, impulse response and transfer function, digital filters, and basic filter types.

1. Introduction

Digital Signal processing (DSP) technology has changed fast and is extremely growing in the commercials sector such as cellular phones, automobiles, stereo equipment, CDs, MP3 formats, and MPEG formats. This growth supports the discrete time signals and systems courses in the electrical and computer engineering undergraduate curriculums to gain a solid understanding of fundamental DSP theory, implementation, and applications in more detail. The changes in computer and DSP technologies and having multimedia capabilities on selections of DSP boards make it easy to implement visual simulations and real time applications.

Many universities have experience to implement DSP in their freshman or sophomore year as DSP course [1, 2, 3]. Some selective universities may be very successful using this approach, but some may have problem with typical freshman student who don't have enough math background. However, I believe that we can introduce DSP in early years. We are offer a required junior level DSP course (four-hour) with a lab (three-hour), which is called ECE306/L Discrete Time Signals and Systems in our curriculum. The ECE306 course introduces the fundamentals of discrete time signals and systems such as Sampling, quantization, solutions of difference equations, z-transform, discrete-time Fourier series, discrete-time Fourier transform, discrete Fourier transform (DFT) and fast Fourier transform (FFT) decimation-in-time calculation of FFT. The three-hour lab is based on the TI TMS320C67xx DSK and uses some selected experiments and simulations of continuous-time and discrete-time signals and systems. Many of the experiments are intent to generate interest in the fundamentals of DSP.

In the senior level, we offer two elective DSP courses and one lab, which are called ECE408 Digital Signal processing I, ECE 428 Digital Signal processing II , and ECE 408L Digital Signal processing I Lab. These courses primarily focus on FIR and IIR filters design, the lattice filters, multirate digital signal processing design, adaptive filters design, employ Discrete Fourier transform (DFT), Fast Fourier Transform (FFT), fast convolution and fast correlation, and architectures of various digital signal processors. The lab experiments provide a more detailed account of hands on experience using TI TMS320C67xx DSK development board.

The main difficulty in teaching discrete Time Signals and Systems at the beginning level is the large number of mathematical equations. To understand the mathematical concepts, students need to visualize the result or input-output relations. MATLAB, Mathematica, System View, and Lab View have the capability of such visualizations. Using such a visualization tool along with TI DSP boards, we can provide real-time experiments to increase student interest in DSP as an area of concentration. Our DSP Lab involves computer based real time exercises to reinforce the concepts introduced to students. Students become more familiar with MATLAB, Simulink, and CCS and will gain experience using TI DSP boards.

The Embedded Target for the TI TMS320C67xx DSP Platform integrates Simulink and MATLAB with TI eXpressDSP (tm) tools. The software suite lets us develop and validate DSP designs from concept through code and automates rapid prototyping on the TI DSP board. The build process creates a Code Composer Studio project from the C code generated by Real-Time Workshop. All the features provided by Code Composer Studio, such as tools for editing, building, debugging, code profiling, and project management, work to help you develop applications using MATLAB, Simulink, Real-Time Workshop, and hardware. Optionally, the Code Composer Studio project is automatically compiled and linked, and the executable is loaded onto to the board, and is then run on the C67xx DSP [4]. This process provides us with a meaningful first exposure to real time signal processing.

2. Typical Hardware Setup for C6713 DSK in Models

The figure 1 presents a block diagram of the typical setup TI C6713 DSK for the lab.

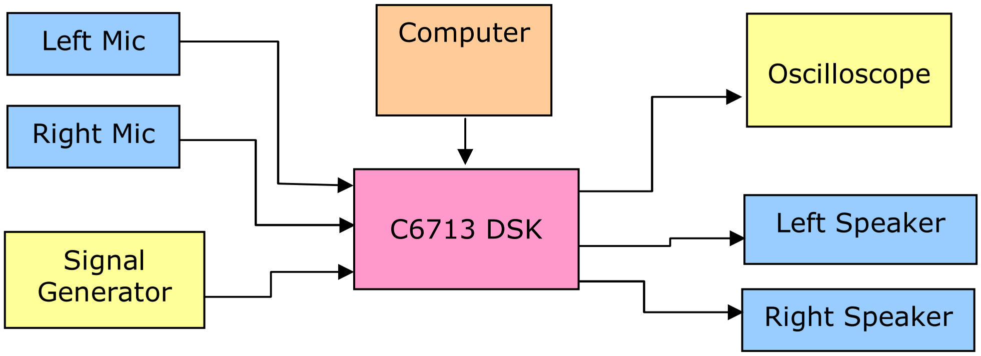


Figure 1. Typical DSP Lab setup

TI C6713 DSK contains a version of each of following blocks:

- ADC block
- DAC block
- DIP Switch
- LED block
- Reset block

Blocks from these libraries are associated with the boards and hardware. As needed, we add the devices to our model. If we choose not to include either an ADC or DAC block in our model, Embedded Target for TI C6000 DSP provides a timer that produces the interrupts required for timing and running your model, either on our hardware target or on a simulator [8].

Figure 2. Student experiment on the lab

3. Lab Contents

Weekly laboratory assignments are provided enhancement on Discrete time signal and system lab. A brief description of the some topics covered and the use of exercises are given below.

3.1 Analog/Digital Conversion

In many continuous time signal applications, it is getting more attractive to convert a continuous-time signal into a digital-time signal. Sampling and quantization determine the accuracy of the digital signal. The ideal condition is that the equivalent discrete-time signal should contain all information of the continuous-time signal. The effects of sampling and quantization can be visualized effectively in the time and frequency domains.

The mathematical theory of sampling and the initiative aspects of sampling can be clarified by visual observation of the effects. Students can see effects of changing the sampling frequency of different input signal such as a sinusoidal and rectangular pulse train signals. The MATLAB Simulink with Embedded Target for TI 6000 Tool Box design gives us the option to change sampling frequency to observe the effects in real time. The aliasing effect can be demonstrated under several sampling cases. Students can then apply different sampling frequencies over speech and music to demonstrate the impact of the sampling rate and required acceptable sampling frequency.

It is important to understand the binary representation of a signal using quantization level. The effects of different quantization levels on sinusoidal signals and voice signals can be demonstrated. Students can apply different number of bit levels to speech and music to determine the number of bit resolution required for acceptable quality conversion.

We can change the sampling frequency and quantization level of the input signal by Using MATLAB Simulink blocks as in figure 3. After generating a code, Real-Time Workshop connects to Code Composer Studio and creates a new project. By compiling and linking the code, Real-Time Workshop then downloads the COFF file to your DSK and begins execution. Students can see the effect on the scope

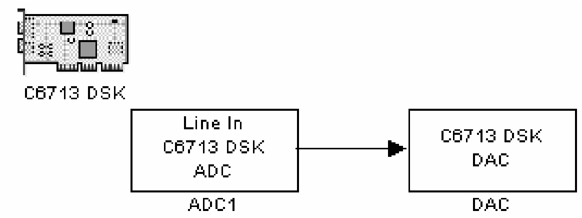
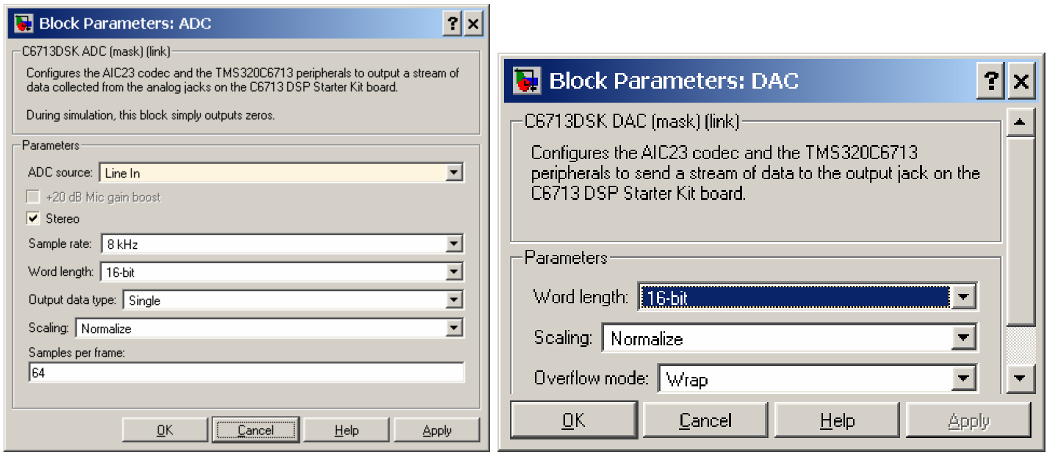


Figure 3 A simple A/D conversion Simulink block



Block Parameters: ADC

C6713DSK ADC (mask) (link)

Configures the AIC23 codec and the TMS320C6713 peripherals to output a stream of data collected from the analog jacks on the C6713 DSP Starter Kit board.

During simulation, this block simply outputs zeros.

Parameters

ADC source: Line In

☐ +20 dB Mic gain boost

☑ Stereo

Sample rate: 8 kHz

Word length: 16-bit

Output data type: Single

Scaling: Normalize

Samples per frame: 64

Block Parameters: DAC

C6713DSK DAC (mask) (link)

Configures the AIC23 codec and the TMS320C6713 peripherals to send a stream of data to the output jack on the C6713 DSP Starter Kit board.

Parameters

Word length: 16-bit

Scaling: Normalize

Overflow mode: Wrap

Figure 4 A simple A/D conversion block parameters.

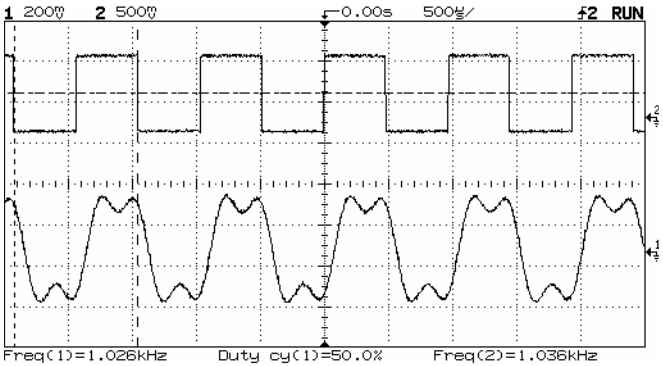


Figure 5 1KHz signal is sampled by 8KHz sampling frequency.

3.2 Digital Filtering Design

3.2.1 FIR Filter

Students learn simple finite impulse response (FIR) in the class. They are able to design simple finite impulse response (FIR) filters by placing zeros at appropriate locations. In the lab, student design simple, less than 10 degree, filters such as LPF, HPF, BPF, and BSF. The expected outputs can be calculated and plotted by MATLAB. Students can design similar FIR filters with Simulink and demonstrate it on TI DSK to compare the results with theory [10].

Example:

Students design a discrete-time system with seven poles at z = 0 and seven zeros at

$$z = \frac{1}{\sqrt{2}} \pm j\frac{1}{\sqrt{2}} \quad , \; z = 0 \pm j \, , \quad z = -\frac{1}{\sqrt{2}} \pm j\frac{1}{\sqrt{2}} \, , \; z = -1 + j0$$

a) Find the transfer function H(z) and find the constant term b_0 such that the gain of the filter at zero angle ($\theta = 0$) is 1, that is,

$$H(\theta)\big|_{\theta=0} = 1.$$

Notice that $H(\theta) = H(z)\big|_{z=e^{j\theta}}$ and $H(\theta)\big|_{\theta=0} = 1$ is equivalent to

.

b) Plot the pole-zero diagram.
c) Plot the magnitude response |H(θ)|. And the phase response ∠H(θ).
d) Find y(n) as a function of x(n), x(n–1), x(n–2), x(n–3), x(n–4), x(n–5), x(n–6), x(n–7).
e) Draw the network structure in direct form.
f) Plot y(n) for x(n) = cos(n2π/16). Use MATLAB.
g) Plot y(n) for x(n) = cos(n2π/8). Use MATLAB.
h) Plot y(n) for x(n) = cos(n2π/4). Use MATLAB.
i) What type of filter (LPF, HPF, BPF, BSF) is this system?
j) Design Simulink block of the filter and demonstrate it on the TI DSK boar to compare results

3.2.2. IIR Filter.

The student will be able to design simple infinite impulse response (IIR) filters by placing poles and zeros at appropriate locations in this lab. Students perform exercises in IIR filters to examine the effect of the filters [10].

Example:

A discrete-time system has a zero at z = –1 and a pole at z = 0.9. The transfer function of this system can be written as

$$H(z) = \frac{Y(z)}{X(z)} = b_0\frac{z+1}{z-0.9} = b_0\frac{1+z^{-1}}{1-0.9z^{-1}}$$

a) Find b_0 such that the gain of the filter at zero angle (θ = 0) is 1, that is,

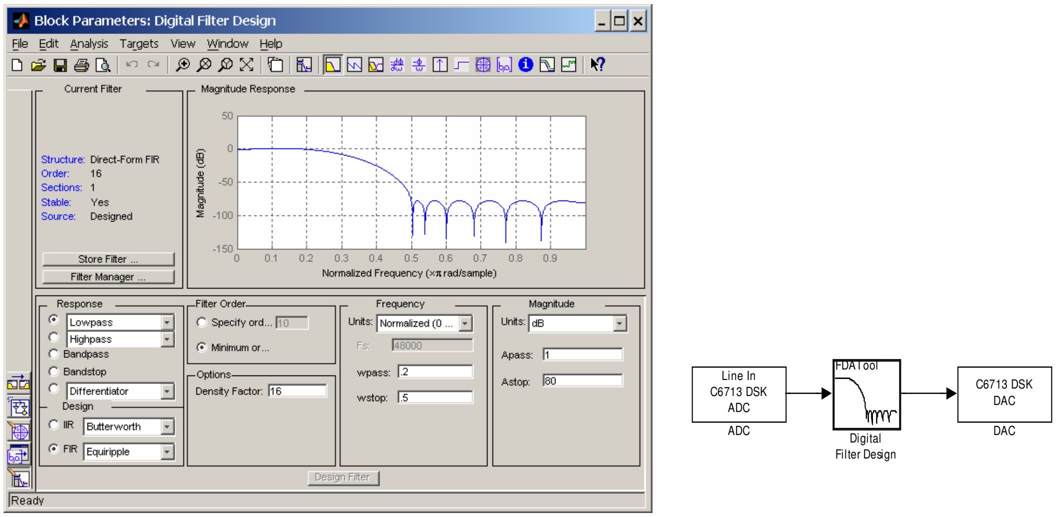
$$H(\theta)\big|_{\theta=0} = 1.$$

Notice that $H(\theta) = H(z)\big|_{z=e^{j\theta}}$ and $H(\theta)\big|_{\theta=0} = 1$ is equivalent to

$$H(z)\big|_{z=1} = 1.$$

b) Plot the pole-zero diagram.
c) Plot the magnitude response |H(θ)|and the phase response ∠H(θ).
d) Find y(n) as a function of y(n–1), x(n), x(n–1).
e) Draw the network structure in direct form II.
f) Plot y(n) for x(n) = cos(n2π/16). Use MATLAB.
g) Plot y(n) for x(n) = cos(n2π/8). Use MATLAB.
h) Plot y(n) for x(n) = cos(n2π/4). Use MATLAB.
i) What type of filter (LPF, HPF, BPF, BSF) is this system?
j) Design the filter with Simulink and apply it to TI SDK board to compare the results.

Later students use a digital filter design block in Simulink to implement higher level filters and test them on the DSK board using speech and music in real time. Students have the option to use different window types such as Bartlett, Rectangular, Hamming, Blackman, and Triangular to choose the best low pass implementation for speech and music. In the experiment, students can change the number of coefficients to find the best tuned signal.


Block Parameters: Digital Filter Design

Current Filter

Structure: Direct-Form FIR
Order: 16
Sections: 1
Stable: Yes
Source: Designed

Response: ● Lowpass

Filter Order: ○ Specify ord... 10 ● Minimum or...

Options: Density Factor: 16

Frequency: Units: Normalized (0 ... Fs: 48000 wpass: .2 wstop: .5

Magnitude: Units: dB Apass: 1 Astop: 80

Design: ● FIR Equiripple

Ready

Figure 6. Digital Filter design

From FIR and IIR filters, students also learn how to evaluate the frequency response of the filters by looking at the oscilloscope for different frequency signals.

3.3 Discrete Fourier Transform:

Theory of Discrete Fourier transform is a difficult concept for students to understand. If we show the effect of Discrete Fourier transform to student visually, they can understand why we need it and how we can use it.

3.4 Fast Fourier Transform

MATLAB and Simulink allow the student to explore the frequency content of signals.

Example: ***Adaptive Filter Algorithm***

Echo cancellation is a signal processing technique for removing reverberant noise from speech and audio signals. Student can demonstrate the Block Least Mean Squares echo cancellation algorithm implemented in the frequency domain, leveraging the computational efficiency of performing FFTs on TI's TMS320C6713 DSP. The algorithm can be implemented using fixed-point arithmetic.

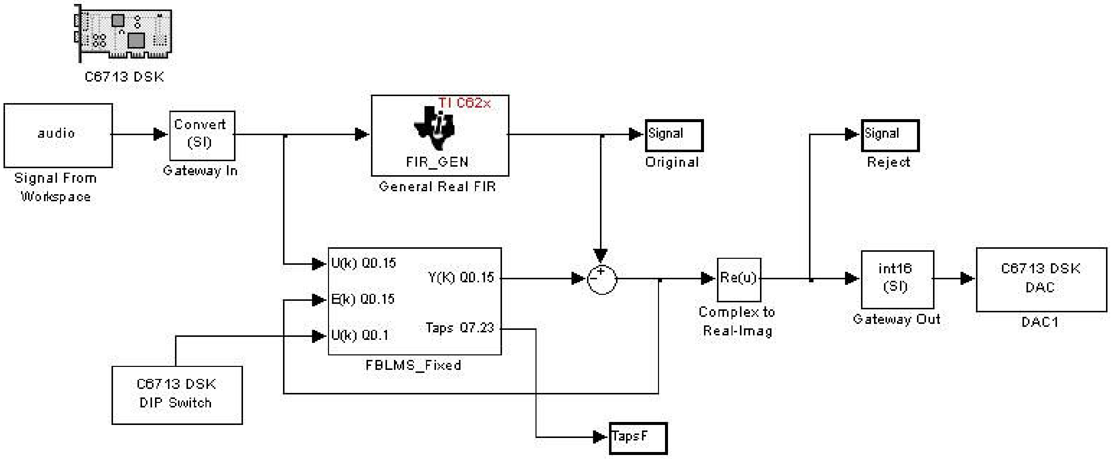


Figure 7. Frequency domain LMS Echo Cancellation [7].

After generating code, Real-Time Workshop connects to Code Composer Studio and creates a new project. After compiling and linking the code, Real-Time Workshop then downloads the COFF file to the DSK and begins execution. If we have connected speakers to the audio output jack of the DSK, we should hear the demo's output signal. We can control the adaptive filter during execution by turning on the first User DIP switch on the board. Set the switch to ON (down) to set the filter coefficients to zero[7].

3.5 Multirate DSP

Using MATLAB, students perform example to implement up-sampling and down-sampling of digital signals. They will be exposed to the aliasing issue, and in return they will be back to design and implement a QFM bank.

Students apply interesting Real-time DSP application lab projects at end of the quarter.

Example: ***Voice removing over the Song*** [9]:

Voice removal is a attending to remove the singer voice from the background instruments. In the basic stereo song, the singer’s voice is equally recorded in the both the left ad right channels and the background instruments have a phase shift in either the left and right channels. Using this approach student can design a voice removal system on Simulink and test it on the TI C6713 DSK board. Specific information can be found on York paper [9].

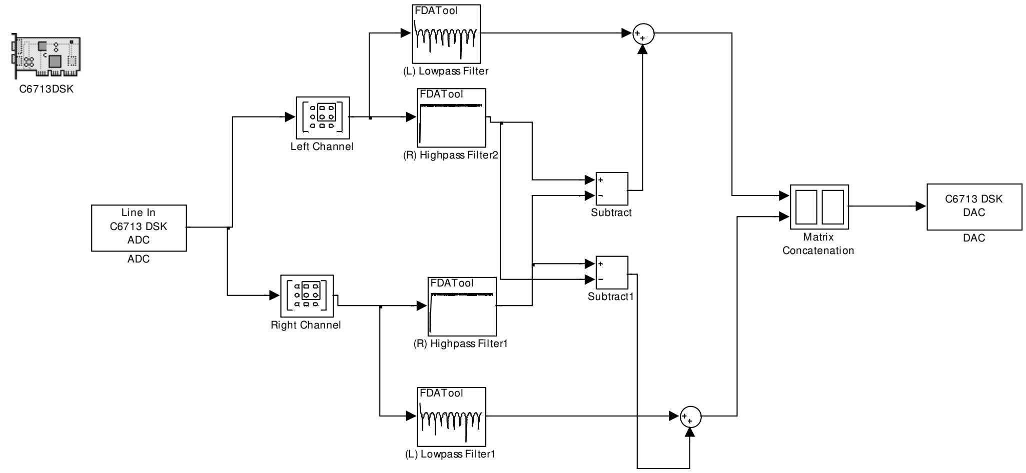


Figure 8. Voice removing over the Song [9]

Conclusion

The effort is to reduce learning time and stimulate comprehension of some basic signal processing concepts. We found that interactive visual interpretation can help students to rapidly increase comprehension of the concepts. Students have less problems using mathematical description after clear visual understanding. At the same time it provides students with real time DSP experience. This required lab also gives familiarities the students with DSP systems to prepare them for the upper level elective DSP courses which cover more complicated applications.

References

1. Don H. Johnson, and , J. D. Wise, ”A Different First Course in Electrical Engineering.” IEEE Signal Processing Magazine, Vol.16 No. 5, pp34-37, September 1999
2. James H. McClellan, Ronald Schafer, and Mark Yoder, “Digital Signal processing First,” IEEE Signal Processing Magazine, Vol 16, No.5, pp. 29-34, September 1999.

3 Sally L. Wood, “DSP Second: A Sophomore Level DSP Architecture course in the Electrical Engineering Core.” Ninth DSP Workshop First Signal Processing Education Workshop 2000

4. T.B. Welch, C.H.G. Wright, and M.G. Morrow, “Real-Time Digital Signal Processing from MATLAB to C with the TMS32C6x DSK,” CRC Press, 2006

5 M. G. Morrow, T. B. Welch, and C. H. Wright, “Enhancing the TMS320C6713 DSK for DSP Education,” in *Proceedings of the 2005 ASEE Annual Conference*, June 2005. Session 3420

6. Texas Instruments, Inc., “C6713 DSK,” 2006. http://focus.ti.com/docs/toolsw/ folders/print/tmdsdsk6713.html.

7. Embedded Target for TI C6000 DSP Demos

8. MATLAB “http://www.mathworks.com

9. George W. P. York, Christopher M. Rondeau, Dane F. Fuller, “Teaching Real-time DSP Applications (Voice Removal) with the C6711 DSK and MATLAB,” Proceeding of 2004 AEEE Conference,

10. James Kang,. Alan Felzer, Zekeriya Aliyazicioglu, “ECE 306 Lab notes”