VOICE RECOGNITION SYSTEMS BASED ON TI DSP PROCESSORS

Isidore Takougang, ***Sophomore Student***

Department of Engineering and Aviation Sciences
University of Maryland Eastern Shore
Princess Anne, MD 21853

Research Adviser: Dr. Ali Eydgahi

Progress Report for:

Chesapeake Information Based Aeronautics Consortium
August 2005

OVERVIEW:

Digital Signal Processing (DSP), simply, is the processing of signals by digital means. A signal in this context can mean a number of different things. Historically the origins of signal processing are in electrical engineering, and a signal here means an electrical signal carried by a wire or telephone line, or perhaps by a radio wave. More generally, however, a signal is a stream of information representing anything from stock prices to data from a remote-sensing satellite. The term "digital" comes from "digit", meaning a number, so "digital" literally means numerical; the French word for digital is numerique. A digital signal consists of a stream of numbers, usually (but not necessarily) in binary form. The processing of a digital signal is done by performing numerical calculations.

Digital signal processors (DSP), such as the TMS320 family of processors, are used in a wide range of applications, such as in communications, control, speech processing, and so on. DSP applications using C an the TMS320C6x board or DSK (Desktop Starter kit) provides a hands-on learning approach to digital signal processing that uses real-time implementation of experiments and projects.

INTRODUCTION:

The project I'm working on is related to speech processing; *Digital Signal Processing using C and the TMS320C6211 board.*

It is an application using DSP techniques and requires at least the basic system as in the figure 1 below, consisting of analog input and output. Along the input path is an antialiasing filter for eliminating frequencies above the Niquist frequency, defined as one-half the sampling Fs. Otherwise the aliasing occurs, in which case a signal with a

frequency higher than one-half Fs is disguised as a signal with a lower frequency. The sampling theorem tell us that the sampling frequency must be a least twice the highest-frequency component f in a signal; Fs>2f meaning also Ts<T/2. for example , if we assume that the ear cannot detect frequencies above 20kHz, we can use a Lowpass input filter with a bandwidth or cutoff frequency at 20kHz to avoid aliasing. We can then sample a music or a voice signal at Fs>40 kHz and remove frequency components higher than 20 kHz.

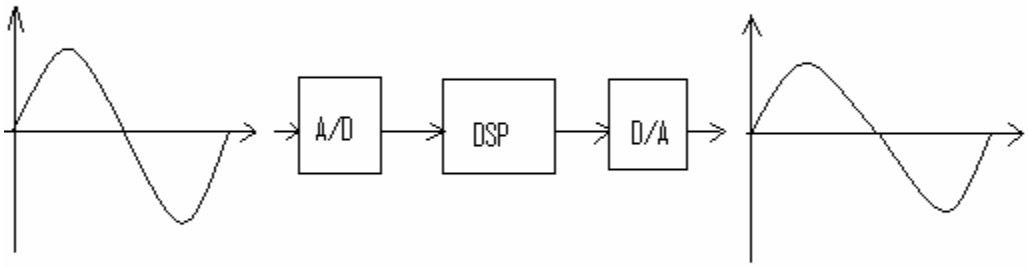


Fig.1 DSP system with input and output

A - for Analog Signal
D - for Digital Signal
DSP - Digital Signal Processor

OBJECTIVE:

The objective for this project is to use the CCS (Code Composer Studio) environment along with the TMS320C6211 board to implement and test signals; they can represent voice, sound, and so on. Convert the input analog signal (voice) into digital signal and convert back into analog using a filter.

IMPLEMENTATION USING MATLB-GUI:

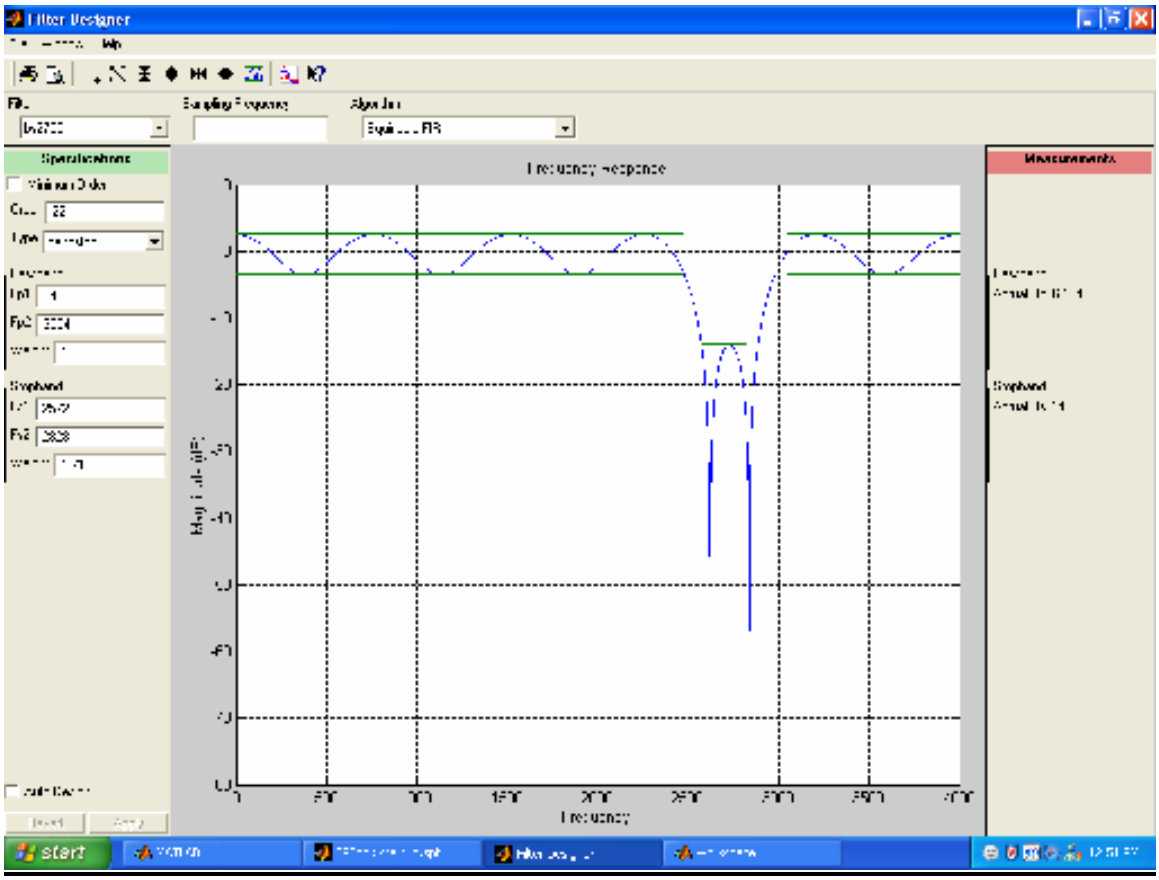
- Access Matlab command window, type: sptool . The GUI designer SPTOOL shows up.

- From the startup window “startup.spt”, select a new design and use the characteristics needed.

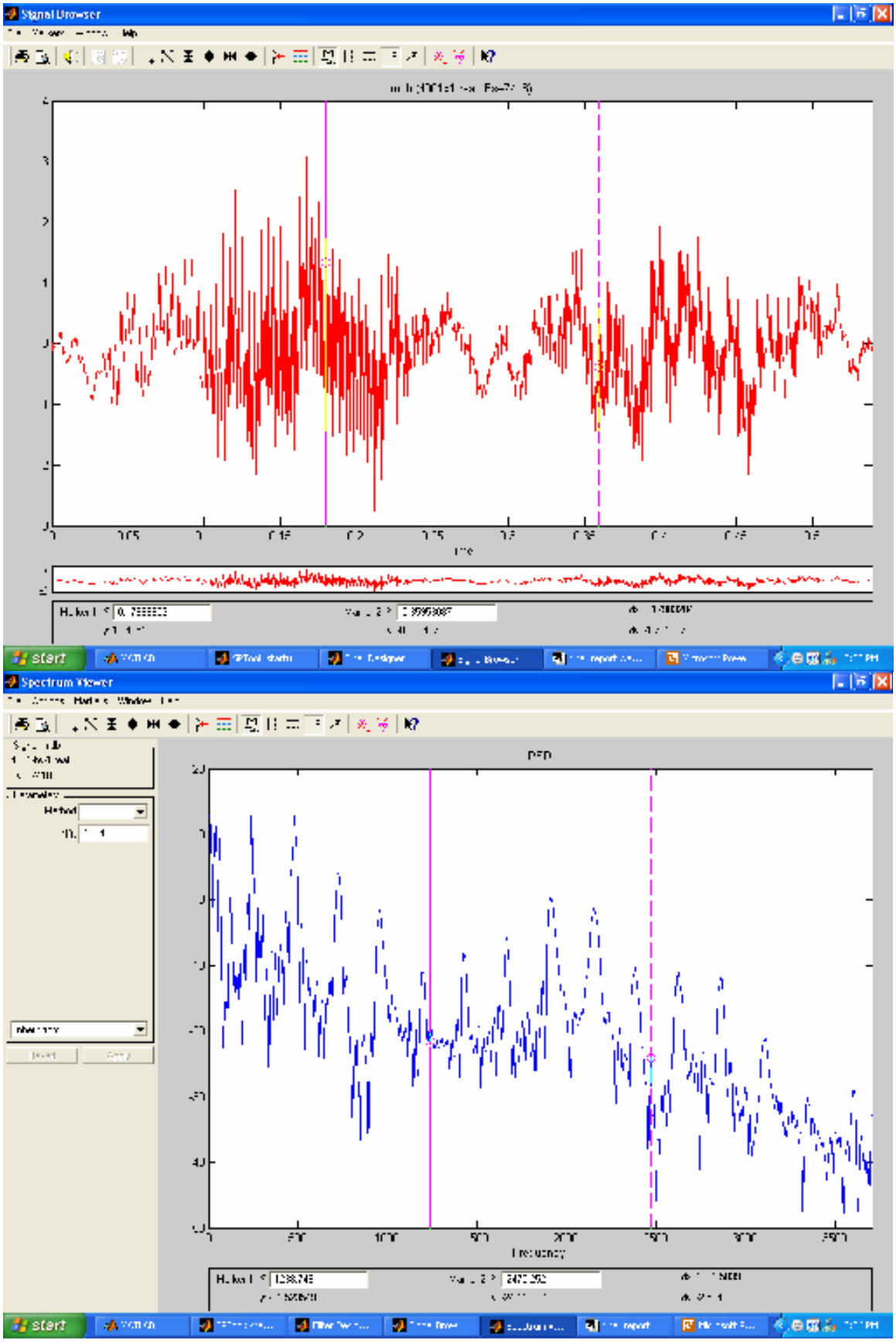
I designed a FIR bandstop filter centered at a frequency of 2700Hz. The filter contains N=89 coefficients and uses the Kaiser window function.

The following figures are the windows displaying the characteristics of the FIR filter:

1) The FIR filter frequency response



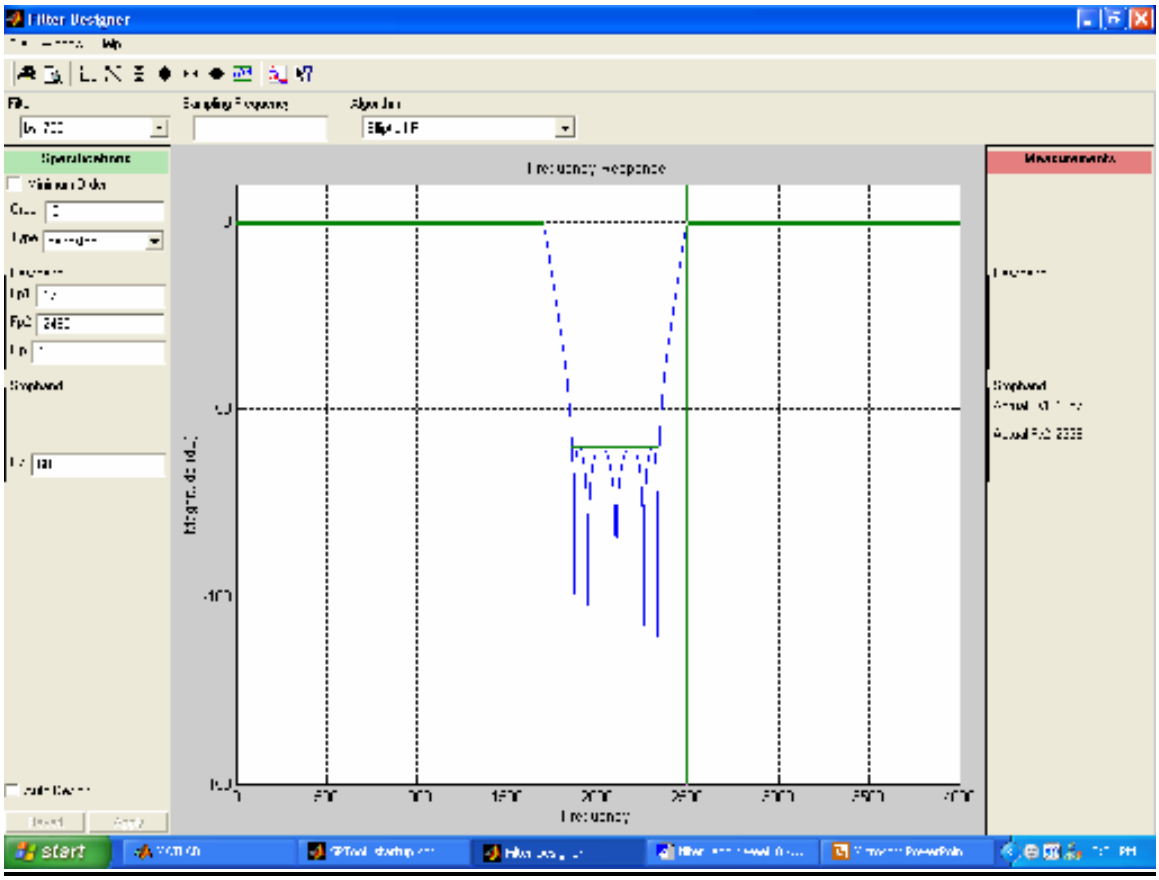
2) The output Signal and Spectra for this filter:


Signal Browser
Spectrum Viewer
psp
Frequency

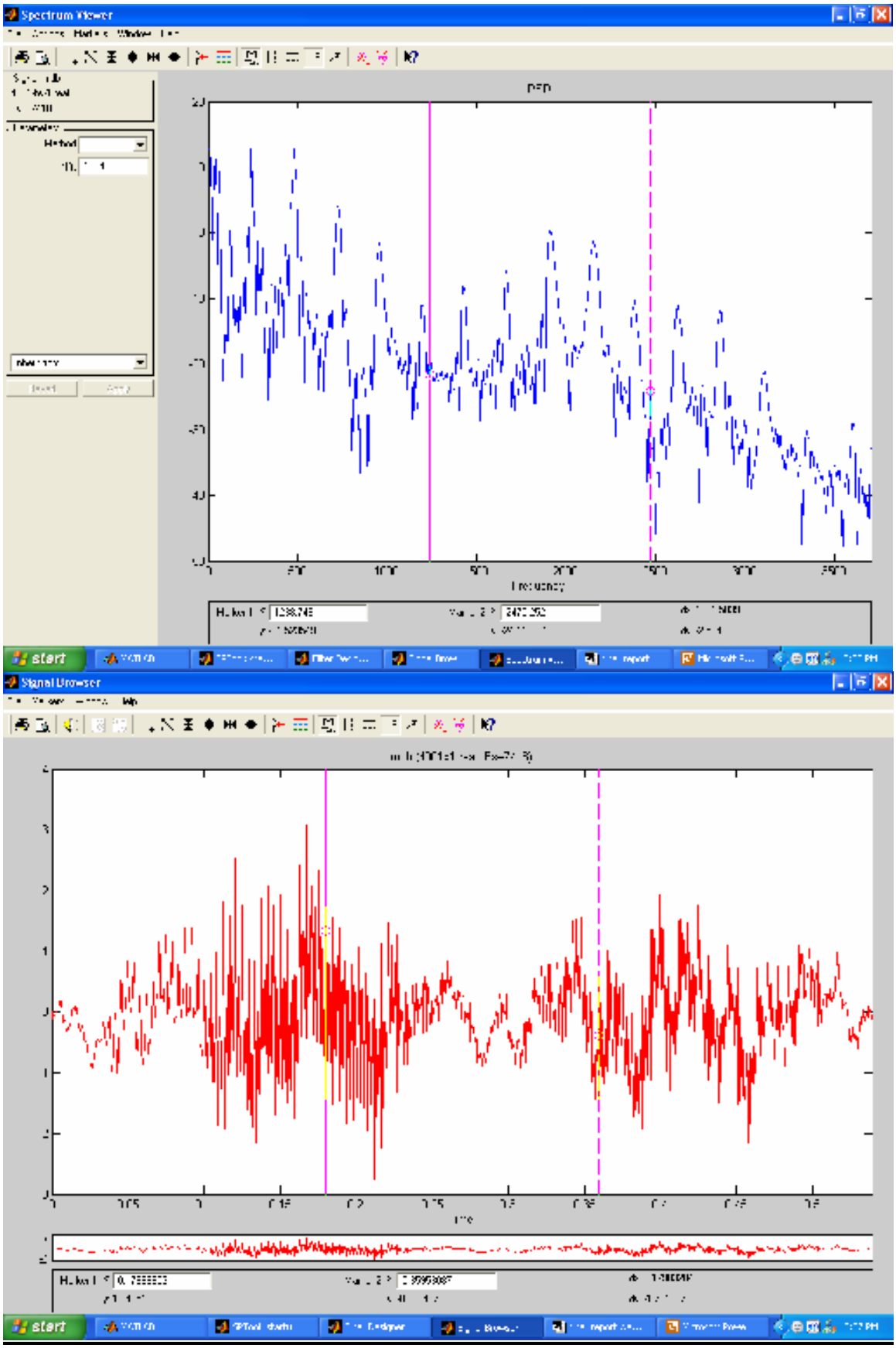
I also designed an IIR bandstop filter centered at a frequency of 1750Hz. The filter contains N=30 coefficients and uses the Kaiser window function.

The following figures are the windows displaying the characteristics of the IIR filter:

3) The IIR filter frequency response



4) The output Signal and Spectra for this filter:

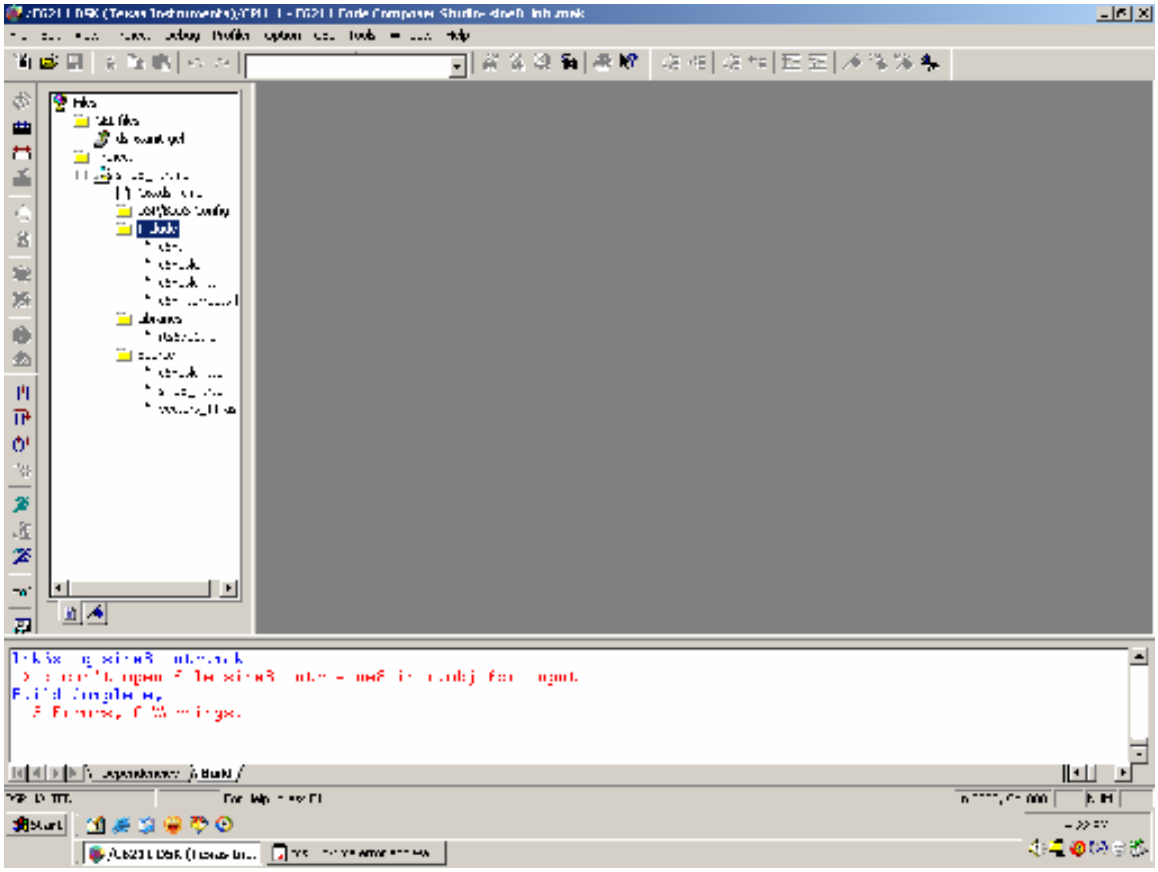

Spectrum Viewer
psd
Frequency
Signal Browser
time

IMPLEMENTATION USING C SOURCE PROGRAMS:

In the CCS environment or window, I created a project called “sine8_intr” to generate a sinusoid which would help later as a folder to implement the filter in the CCS;

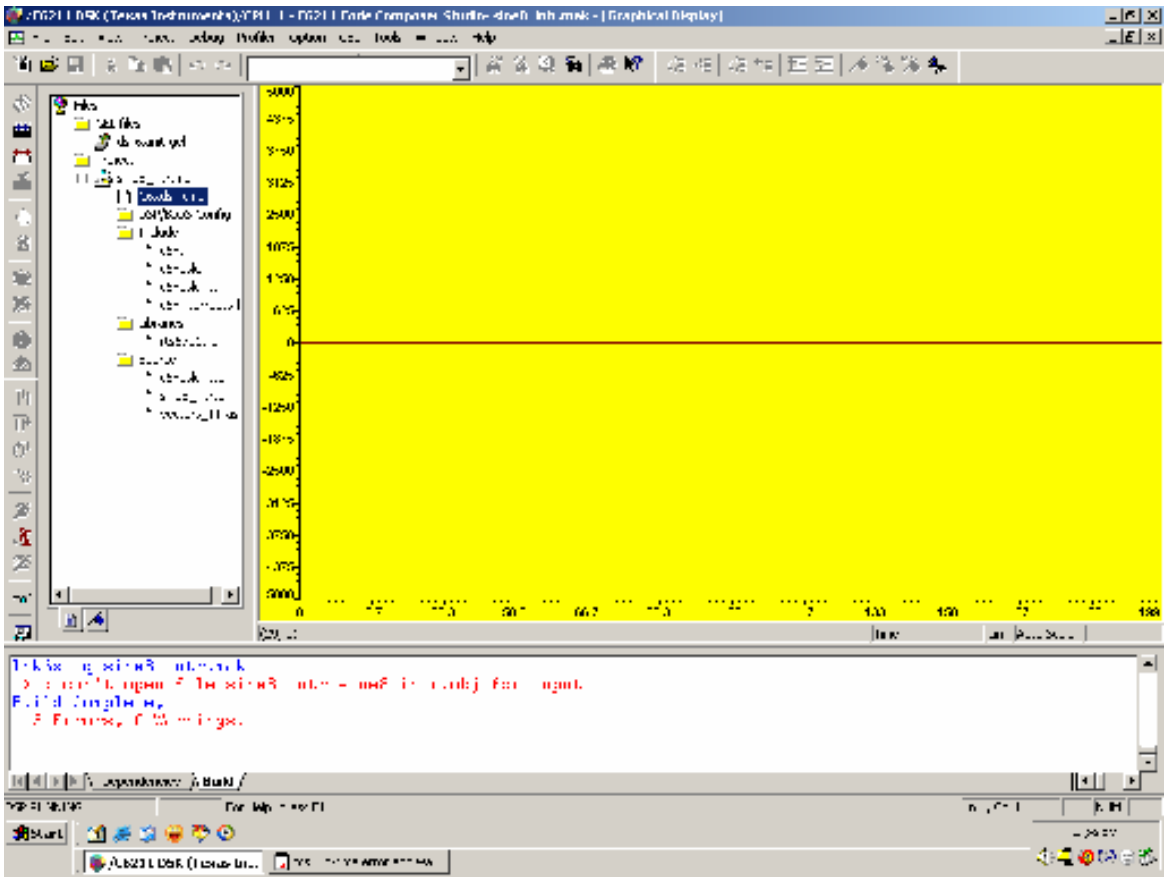
Select Project- new, and type sine8_intr. Then add files from the accompanied disc to project. Built and link project be selecting project-build-options and then linker option. After building project, load program and finally select debug-run to run the project. Connect a speaker to the Out connector of the TMS320C6211 board to hear a tone.

First result and observation:



All the necessary files have been loaded like shown in the file window of the CCS. After linking, building and then trying to run, it failed; the linking process can’t open for input and the building process contain 3 errors. There was a warning before, but I reloaded the

program and the warning disappeared. The following display is just a failure; no output signal of the filter.



CONCLUSIONS:

1)-I've implemented FIR (Finite Impulse response) filter and IIR (Infinite Impulse Response) filter using Matlab support tools. I obtained a frequency response at the output for each filter. Now I'm going to interpret and compare the results obtained from the two filters; they consist of the frequency response signal and spectra at the output (graph on the CCS window or Matlab-GUI) after an input signal which is a voice. This implementation is a speech processing in term of filtering a voice signal.

2)- After loading the project sine8_intr.out, I should debug and run; connect a speaker to the out" of the board and should hear a tone, also observe the signal output on the display graph on the CCS windows. But unfortunately, so far I've been trying to figure out where the errors were coming from and debug but without success. That what I'm going to work on now in order to continue with the C program sources part of the project.