Noah Erickson
Dario Morote
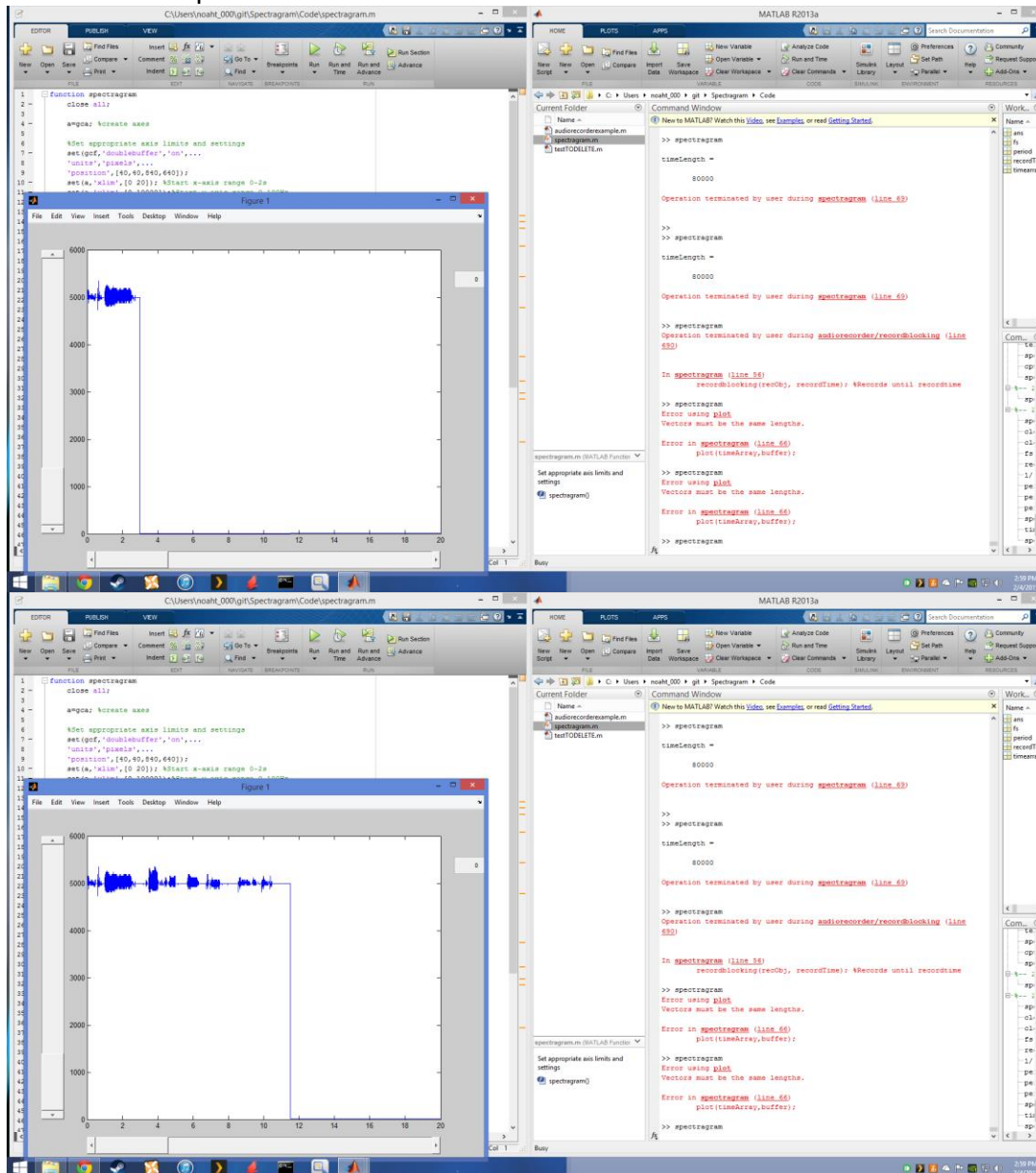February 5th, 2015
ECE 312 Fourier Analysis
Homework #3

Spectragram Progress – Continuously plotting input data from microphone

So far the spectragram will plot input from a microphone continuously for 20 seconds. After the 20 seconds the program quits.

Screenshots of plot:

The code:

```
function spectragram
    close all;
    a=gca; %create axes
    %Set appropriate axis limits and settings
    set(gcf,'doublebuffer','on',...
    'units','pixels',...
    'position',[40,40,840,640]);
    set(a,'xlim',[0 20]); %Start x-axis range 0-2s
    set(a,'ylim',[0 10000]);%Start y-axis range 0-100Hz

    % Relative positiong for sliders and buttons for resizeability
    pos=get(a,'position'); %pos(1) is left - pos(2) is bottom - pos(3) is width - pos(4) is
height
    hSliderPos=[pos(1) pos(2)-0.1 pos(3) 0.05];
    vSliderPos=[pos(1)-0.1 pos(2) .05 pos(4)];
    fsBoxPos=[pos(2)+pos(3)+.05 pos(4) .1 .05];

    %Callback functions for the GUI interactivity
    hSlideCallback=['set(gca,''xlim'',[0 get(gcbo,''value'') ])'];
    vSlideCallback=['set(gca,''ylim'',[0 get(gcbo,''value'') ])'];
    fsBoxCallback=['set(amp,str2num(get(gcbo,''string'')))'];

    %GUI Elements
    hSlide = uicontrol('style','slider',...
        'units','normalized','position',hSliderPos,...
        'callback',hSlideCallback,'min',2,'max',20,'value',2,...
        'SliderStep',[0.0556 .2778]); %Change max and min here to change sliders

    vSlide = uicontrol('style','slider',...
        'units','normalized','position',vSliderPos,...
        'callback',vSlideCallback,'min',100,'max',10000,'value',100,...
        'SliderStep',[0.0556 .2778]);

    fsBox = uicontrol('style','edit',...
        'units','normalized','Position',fsBoxPos,...
        'String','0',...
        'callback',fsBoxCallback);
    loop = 0;
    fs = 8000; %sampling frequency, to be a GUI element later
    recordTime = .5; % 5 milliseconds of recording, refresh rate of sorts
    period = 0:1/(fs):recordTime; %period is 1/sampling frequency until recordtime
    timeLength = (fs*recordTime*20); %How long does the time array need to be
    timeArray = 0:1/fs:20; %Create time array, store data from 20 seconds
    buffer = timeArray; %Initiliaze the buffer array
    x=1;
    while x==1 %infinite loop, CTRL+C to stop
        myRecording = period; %initialize the myRecording to be same as period
        recObj = audiorecorder(fs,8,1); % initiliazes microphone with sampling frequency
        recordblocking(recObj, recordTime); %Records until recordtime
        myRecording = getaudiodata(recObj); %get audio in
        myRecording2 = myRecording*5000+5000; %scale array for hw purposes
        myRecording3 = rot90(myRecording2); %Rotates the recorded array
        myRecording3 = [myRecording3, zeros(1, length(period) - length(myRecording3))]; %pads
zeros to meet period array
        i = 1;
        while i<recordTime*fs+1 %loop until i is equal to the length of myRecording [use
length(myRecording3)??]
            buffer(i+loop*recordTime*fs) = myRecording3(i);
            i=i+1;
        end

        loop=loop+1;
        plot(timeArray,buffer);
        drawnow
        buffer = circshift(buffer,-(recordTime*fs+1));
        timeArray = circshift(timeArray, -(recordTime*fs+1));
    end
```