We want to create a GUI that loads a sound signal, calibrates it and calculates the sound pressure.

Type “guide” in the command window

Choose “Create New GUI” and select “Blank GUI”

Select the “menu editor” and press “new menu”. Select the newly created menu and give it the label “load” and the tag “loadwav” (you can also assign a shortcut key to it). Press “OK”.

Draw an “axes” on your UI.

Save your UI. This will create an m-file with the name you selected for you UI.

Scroll down until you see “function loadwav\_Callback(h0bject, eventdata, handles)” (this may differ slightly between matlab versions). Under the commented lines write:

>>global sig fs

>>[name path] = uigetfile('\*.wav;\*.WAV','load sound file');

>>[sig fs] = wavread([path name]);

This will create a window that lets you select a wav file of your choice and then load it using wavread. It also makes the variables “sig” and “fs” available to the entire program

Lets plot the signal with the correct timescale but with the un-calibrated pressure scale:

>> t = (0:length(sig)-1)\*1000/fs;

>> handles.axes1; %calls the axes that you drew earlier

>> plot(t,sig)

>> axis([0 t(end) -max(abs(sig))\*1.1 max(abs(sig))\*1.1])

>>xlabel('Time (ms)')

>>ylabel('Pressure (Pa)')

To create a calibrated signal, we need to know 1) the sensitivity of the hydrophone/microphone 2) the clipping level of the ADC 3) the amplification. So lets create 3 text input.

On your GUI canvas, draw 3 “edit text” windows. Double click on the first one. This will open the “object inspector”. Scroll down to “string” and write “-200” and at “tag” write “sensitivity”. This will label the text box as “sensitivity” and make it display the value “-200”.

Likewise for nr 2, “string” -> “1” , “tag” -> “cliplvl”

For nr 3, “string” -> “0” , “tag” -> “amp”

To simplify things, we will execute the calculation with the press of a button. Draw a “push button” on your canvas and change the “tag” to “calc”

Save the GUI.

Scroll down to “function calc\_Callback(h0bject,…”

First we need to make “sig” and “fs” usable when pushing the button. Write:

>>global sig fs

Next, we want to obtain the values from the text input we created. These are strings now, so we need to tell matlab that they are numbers. To do so, write:

>>S = str2num(get(handles.sensitivity,'string'));

>>c = str2num(get(handles.cliplvl,'string'));

>>A = str2num(get(handles.amp,'string'));

Now we can calibrate the y-scale as we’ve done earlier:

>>s = 10^(S/20+6);

>>a = 10^(A/20);

>>WAVClippingLevel = c/(a\*s);

>>Sig = WAVClippingLevel \* sig;

Now all we need to do is create the time scale and plot the new signal:

>>t = (0:length(sig)-1)\*1000/fs;

>>handles.axes1;

>>cla %clears the axis

>>plot(t,Sig)

>>axis([0 t(end) -max(abs(Sig))\*1.1 max(abs(Sig))\*1.1])

We can now easily calculate the peak pressure. Lets create a text window that displays this. This time, draw a “static tex” window on your canvas. This value cannot be edited. Set the tag of this window to “peak” and “string” to “N/A”

Under the calc function, after the code you’ve already written, write:

>>peak = round(20\*log10(max(abs(Sig))/1e-6));

>>set(handles.peak,'String',num2str(peak))

Hints:

-If things are not working, the first thing you should try is to close the GUI and write “clear all” in the command window, then run your GUI again. It is crucial that you don’t write “clear” but “clear all”. This will clear all the variables within functions as well, these are not cleared when you only use “clear”.