ECE 403 **403 Lab Manual** **Spring 2017**

Univ. of Illinois **Version 1.1 January 22, 2017** **Prof. Allen**

Introduction

This manual is designed to help organize the labs associated with ECE 403. The main theme of ECE 403 is to learn how loudspeakers (LS) work, and the labs are designed to give you hands-on experience with classic measurements and theories of loudspeakers.

The basics are outlined in the textbook (Beranek & Mellow, 2012). All the course material may be found on Professor Allen's website.[1] Once there, look to the left under **Teaching** and you will find, at the top of the list, a link for ECE 403 (S2017). Once you click on this link, you will see the 403 course website. There is a clickable link titled **"This week's schedule"**, which is updated every week, so you don't need to search for the date.

Lab exercises

There are 7 labs, plus an orientation (Lab 0). The topics of the labs are also defined on the website.[2] They are described below:

Lab 0 (Feb 2, Lect 6) Orientation:

a) Define teams (survey Thursday Jan 25)
b) Define goals of the labs

Lab 1 (Feb 9, Lect 8) Measure test resistor:

a) Install g7play software
b) Construct op-amp circuit
c) Measure impedance of a test resistor (e.g. 1 [kΩ])

Lab 2 (Feb 16, Lect 10) Measure 2-port networks:

a) RC low-pass
b) Twin-T notch filter

Lab 3 (Feb 23, Lect 12) Measure loudspeaker electrical impedance:

a) Speaker alone (unloaded)
b) Speaker cone mass-loaded
c) Compute speaker motional impedance

Lab 4 (Mar 16, Lect 18) Reciprocity calibration:

a) Calibrate by placing the speakers face to face
b) Verify calibration with probe microphone (ER-7C)

Lab 5 (Mar 30, Lect 20) Acoustic Thévenin calibration:

a) Measure 2 known acoustic loads (Duct tape cavities) by playing a wideband chirp from the loudspeaker and measuring with the probe microphone (ER-7C).

---

[1] `http://auditorymodels.org`

[2] `http://auditorymodels.web.engr.illinois.edu/index.php?n=Courses.ECE403-2016AudioEngineering-S17#Labs`.

b) Solve for acoustic Thévenin equivalent circuit (e.g. source pressure and source impedance)

Lab 6 (Apr 11, Lect 23) Measure speaker acoustic radiation impedance $Z_{rad}(f)$

Lab 7 (Apr 25, Lect 27) Optional lab (team's choice):

a) ear-bud in ear simulator
b) Wave horn baffle
c) AR-3 Acoustic Suspension baffle

Network-analyzer (MU Box)

The labs require a specialized piece of equipment that can make precise frequency response measurements of a 2-port circuit. This is done with what we call the '`MU box`', which is being provided to you by Mimosa Acoustics (Freemont St, Champaign IL). We have four of these systems.

The `MU-box` is not replaceable, so if one were lost or damaged, it would be very serious. So I beg you to be careful with these units. Do not remove them from the Lab; when you're done with them, always return them to the locked cabinet against the west wall, and return the key to its default location. We shall discuss these details during Lab 0 and again during Lab 1.

Unfortunately such analyzers are not readily available. If they were, at a reasonable price, we would be using it rather than the Mimosa MU-box. Hewlett Packard once sold such a system that worked in the audio range (HP-3317), but I believe these are now only available second-hand. A list of application notes are still on line however, and these are worth a look.[3]

Theory of use: The Network-analyzer (NA) puts out a periodic stimulus, typically a short wideband signal (e.g. a chirp), and then captures the response to this signal, in a time-synchronous manner. This requires the output and input to be precisely locked in time. It is this requirement which makes this equipment special. This time-synchronization is best done at the codec (hardware) level. To make accurate phase measurements, it is essential these be synchronized exactly, to the sample. Furthermore there can be no jitter between the output and input samples, or the system will totally fail to make an accurate measurement.

The output stimulus is repeated N_r times; the response blocks are simultaneously captured and then averaged together. In this way, noise is averaged out (as it is not locked to the stimulus, thus appears random over time).

Finally the response is *de-chirped*, which means the phase of the stimulus is removed from the time-averaged block. Both the stimulus block and response block are typically taken to be a power of 2 (e.g., $NFT = 2048$ samples), so that a standard FFT may be used to dechirp the stimulus. You will find the steps that do this operation in the Matlab/Octave routine `./bin/show_data.m`, at the `G7-software` link on the website (second bulleted item at top of the page).

The MU Box has 2 output channels, CH0-OUT & CH1-OUT, and 2 input channels, CH0-IN & CH1-IN. It is designed to function over the audio frequency range, (e.g. human hearing 0.01-20 [kHz]).

---

[3] `http://na.support.keysight.com/faq/Document_list.html`

Lab 0 (Feb 2)

In some ways, Lab 0 is not a lab, since there is no assignment. The goal is to set up the teams, of 4 or 5 members. These are selected so that there are four competencies on each team (i.e., each team has a good balance of skills):

1. Organizer
2. Matlab/Octave programmer
3. Technical document writer
4. Electronics specialist

Organizer: This is the person who makes sure that all is running smoothly. This does not mean he/she has control of the project. They are not a "boss," rather this person is responsible for making sure that the team is working efficiently, and keeps track of the schedule, so there is minimal chaos.

The first job of the Organizer is to work out a unique name for your group. The team needs to agree on this name. It is an important identifier that will be used by all of us to know who is doing what.

Programmer: This person takes primary responsibility to assure the code is properly maintained. If others contribute to the code base (hopefully everyone contributes), the Programmer assures that the code is backed up, and organized in a proper manner.

Technical writer: This person has a very important job, of collecting the results, and putting it in an organized coherent format. Hopefully this person knows LaTeX, as that is the only documentation system that delivers a high quality output. By comparison, Microsoft Word is a very poor substitute (it is quite poor at doing math and figures). Learning how to use a text editor is very important. I don't care what you use, as long as it is not MS-word.

Electronics: This person is someone who understands how to make an op-amp circuit, and keep it running. He or she knows how to verify that the battery is not dead, and knows how to verify that the circuit is working. This person knows how to use a scope and a multi-meter.

Lab 1 (Feb 9)

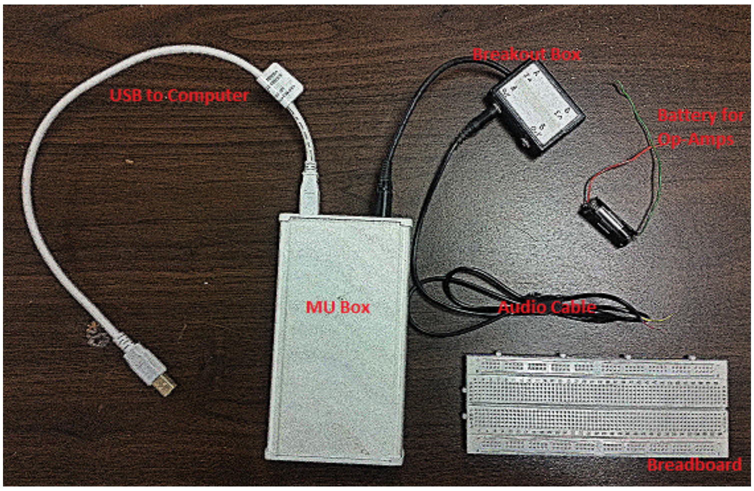


Figure 1: Some equipment you will use in the labs.

Part 1: Installing the G7 Software package

What you need:

1. One (or more) computers with access to either Matlab or Octave. Matlab requires a software license, which you can get through the university as an engineering student (but it requires a remote connection[4]). Octave is open-source (free), and the G7-Software package should run on Octave as well (*TA NOTE: Octave's graphics are quite slow for me, but remote connections can also be slow*).

   Octave has one bug that we know of, which has a simple work-around: It will crash if you try to save the graph using the `print` command (e.g., `print -djpg Filename`). For details type either `doc print` or `help print` at the Matlab/Octave command line. The workaround is to turn off the graph before calling the print command, and then turning it back on once the file is written. The command to do this is

   ```
   % Not tested on every possible platform
   h=figure;
   %make figure, ready to print
   ... (your code to make figure) ...
   %now print
   set(h, ‘visible’, ‘off’); %work-around for printing with octave
    print -depsc2 figure     %Generate a level 2 color eps file
   set(h, ‘visible’, ‘on’);
   ```

   If this fails, please let us know.

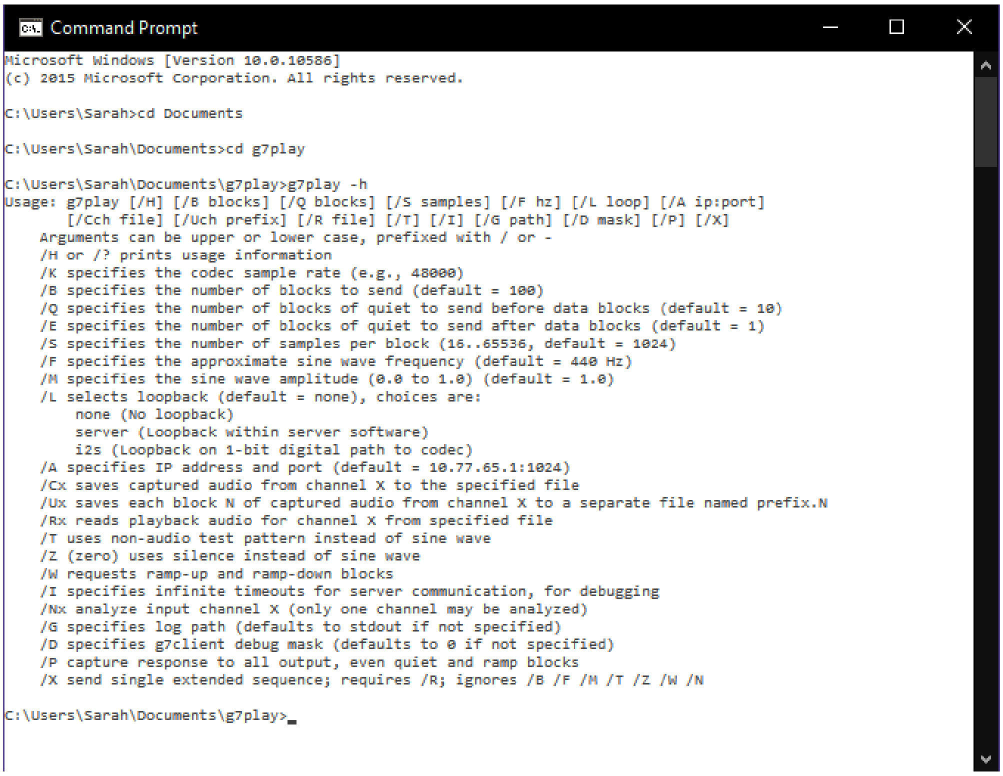
2. An ‘MU box’ (the white box with 2 cable inputs). There are 4 MU boxes. These will be shared between teams (each team will have their own during their lab session, but another team will use the same box in a different session).

3. USB cable connection between the MU box and your computer (the other MU box input/output will be connected to your test circuit, you don't need this now)

---

[4] https://it.engineering.illinois.edu/ews/lab-information/remote-connections

4. The G7-Software package (to download all files, download ‘all.zip’) from the course website, which runs the g7play executable to send and receive signals from the MU box. *NOTE: The directory structure matters when you run the top-level program `run_g7play.m`!*



```
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Sarah>cd Documents

C:\Users\Sarah\Documents>cd g7play

C:\Users\Sarah\Documents\g7play>g7play -h
Usage: g7play [/H] [/B blocks] [/Q blocks] [/S samples] [/F hz] [/L loop] [/A ip:port]
       [/Cch file] [/Uch prefix] [/R file] [/T] [/I] [/G path] [/D mask] [/P] [/X]
    Arguments can be upper or lower case, prefixed with / or -
    /H or /? prints usage information
    /K specifies the codec sample rate (e.g., 48000)
    /B specifies the number of blocks to send (default = 100)
    /Q specifies the number of blocks of quiet to send before data blocks (default = 10)
    /E specifies the number of blocks of quiet to send after data blocks (default = 1)
    /S specifies the number of samples per block (16..65536, default = 1024)
    /F specifies the approximate sine wave frequency (default = 440 Hz)
    /M specifies the sine wave amplitude (0.0 to 1.0) (default = 1.0)
    /L selects loopback (default = none), choices are:
        none (No loopback)
        server (Loopback within server software)
        i2s (Loopback on 1-bit digital path to codec)
    /A specifies IP address and port (default = 10.77.65.1:1024)
    /Cx saves captured audio from channel X to the specified file
    /Ux saves each block N of captured audio from channel X to a separate file named prefix.N
    /Rx reads playback audio for channel X from specified file
    /T uses non-audio test pattern instead of sine wave
    /Z (zero) uses silence instead of sine wave
    /W requests ramp-up and ramp-down blocks
    /I specifies infinite timeouts for server communication, for debugging
    /Nx analyze input channel X (only one channel may be analyzed)
    /G specifies log path (defaults to stdout if not specified)
    /D specifies g7client debug mask (defaults to 0 if not specified)
    /P capture response to all output, even quiet and ramp blocks
    /X send single extended sequence; requires /R; ignores /B /F /M /T /Z /W /N

C:\Users\Sarah\Documents\g7play>
```

Figure 2: The command **g7play -h**, and resulting information about g7play.

`g7play` Basics: You may run the MU box by interacting with executable routine `g7play` (`g7play.exe` in MS-Windows) using a command-line shell, such as `bash`, or MS-Windows `cmd`.

1. Download/install the g7play software from the course website
   (a) Go to https://web.engr.illinois.edu/~jontalle/uploads/403/G7-software/
   (b) Just g7play: Find the g7play program for your operating system and download that program only.
   G7 Software Package: Download ‘all.zip’ or ‘all.tgz’ to get all programs to control the MU box from Matlab or Octave (you may need a specialized program to unzip ‘tgz’ files - in Windows, you can use a program called ‘7-Zip’).
   NOTE: For Windows, you need to install a driver ‘mimosa.inf.’ This can be found the in the ‘DRIVERS’ folder. Download the file, then right-click and choose ‘install.’
2. Testing g7play in the command line (you don’t need the MU box for this)
   (a) Open a command prompt and navigate to the folder where the g7play program is located (e.g., using the command **cd** to ‘change directory’ - if you aren’t familiar with the command line, find a web tutorial).
   (b) Type **g7play -h**. This command should return detailed information regarding the use of g7play (Fig. 2).

(c) Try typing **g7play** (this command alone will run g7play using all default parameters). Because the MU box is not connected, this should return an error message 'Not connected.'

NOTE: For a Mac, it may only work if you use the command './g7play' rather than 'g7play' - in Windows, '.\g7play' should also work, and in Linux './g7play' should work. In general, Mac and Linux file paths use forward slashes, while Windows uses backslashes. Therefore, if you want to call g7play when your current directory is one folder above it, the command would be './foldername/g7play' (Mac/Linux) or '.\foldername\g7play' (Windows).
NOTE: If you get the error message "g7play' is not recognized...' check your syntax and current directory (relative to the location of the executable file).

3. Testing g7play with the MU box.
NOTE: Many of the following tests can be run from the command line or from Matlab (or Octave) using the function `system.m`, which takes a string input (e.g. type `system('your command')` into Matlab's command window). Remember, the syntax of how you call the g7play executable depends on what directory you are in relative to where the executable is located!

   (a) (Just g7play) Connect the MU box to your computer via USB. The LED on the box will blink blue while connecting. Once the LED turns green, the MU box is ready.

   (b) The IP Address for the box is 10.77.65.1:1024. You can try to 'ping' the MU box using the command 'ping 10.77.65.1' - this test communication over the USB connection.

   (c) Now try the command **g7play**. With the default parameters (shown in Fig. 2) g7play is sending 100 blocks of length 1024 samples playing a sine wave at approximately 440 [Hz]. You should see many messages such as 'got 2 blocks from g7_read_input...' showing that data blocks are being returned (these are empty, because you have no voltage input plugged in).

   (d) Try modifying the g7play parameters (e.g. using commands such as **g7play -b 10 -f 1000**).

   (e) (G7 Software Package) Run g7play from Matlab and view data (or in this case view noise - since there is no circuit hooked up!)

      i. Try running the program `run_g7play.m` which calls g7play from Matlab (or Octave). Make sure your current directory in Matlab is the level where `run_g7play.m` is located. Go to the top of the program and uncomment/comment code to choose your operating system.

      ii. Go to the program `show_data.m` and try switching the binary flags that allow you to plot the impedance (note this plot is for a specific test circuit) or the stimulus.

Interlude: Getting to know the 'Breakout Box' The 'breakout box' takes 4 RCA-mini audio cable inputs, for each of the two outputs (CH0-OUT and CH1-OUT) and two inputs (CH0-IN and CH1-IN) of the MU Box. Each of these inputs and outputs has two terminals: signal and ground. Because the RCA cables have 3 wires (for stereo sound: left, right, and ground), one of the wires will remain unused. Figure 3 one of the cables we have for the lab, marking which wire is which so that you will know which wires to connect to your breadboard circuit. Using a multimeter, you can easily check which wire connects to which region of the plug (which wire is left, right, or ground). Some other cables we have with the lab equipment have 2 wires (left and right channels) plus a shield (ground).

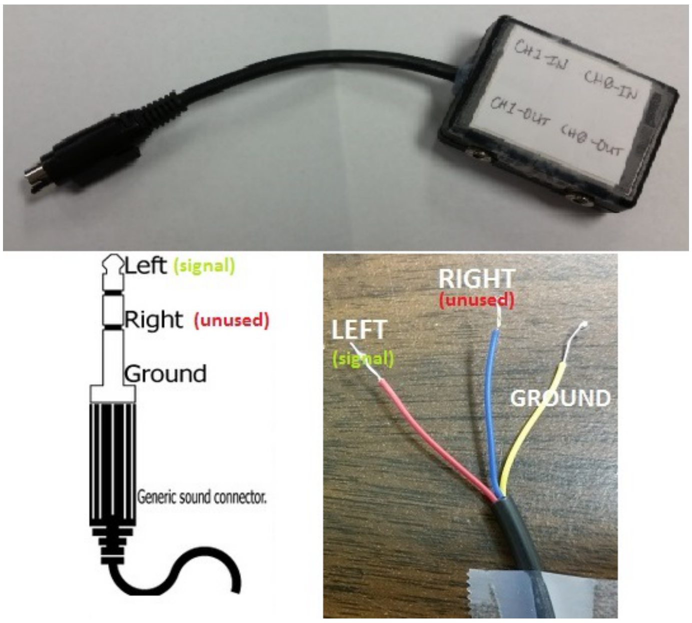


Figure 3: Description of 3-wire audio cables and breakout box. Each channel on the breakout box has two terminals: signal and ground. Because the audio cables are for stereo sound, we have one unused wire on each cable. You will connect the signal and ground wires to your breadboard.

Part 2: Building the Op-amp circuit

NOTE: Once you have made the breadboard, store it in a protective container (e.g., Tupperware) with your Team's (or your Organizer's) name on it, for quick identification. Good to leave a phone number as well.

Your breadboard will have two main interconnected circuits: A circuit to power your op-amps, and a circuit to measure a test impedance. Please study both parts of the circuit (Figs. 4 and 5) before you begin, so you can plan you breadboard accordingly.

1. Op-Amp circuit: Build the Op-amp breadboard that we will use to measure the loudspeaker impedance, as described below and shown in Figure 4. The purpose of the Op-amps is to isolate the measurement from the finite impedance of the MU box. The circuit diagrams of the MU box are provided in the 'DOC' folder of the software package. Based on these circuit diagrams, one may determine the Thévenin source impedance of the output voltage, and the input (load) impedance of the input circuits. There are two critical op-amps, one for output and one for input. The first input does not need an isolation Op-Amp since the output of the the first Op-Amp is zero ohms.

2. Impedance-measurement circuit: Build the circuit shown in Figure 5, connecting to the op-amps and the MU box as shown in the diagram. Using a voltage divider with a reference resistor R_{ref}, you will be able to measure an unknown impedance Z_{test}.

   The voltage across the unknown impedance Z_{test} is $V_{test} = V_0 - V_1$, and the voltage across the reference resistor is V_1.

   $$V_{test} = V_0 - V_1, \qquad I_{test} = \frac{V_1}{R_{ref}}, \qquad Z_{test}(f) = R_{ref}\frac{V_0 - V_1}{V_1} = R_{ref}\left(\frac{V_0}{V_1} - 1\right),$$

   The test resistor is typically $\approx 0.5 - 2$ kΩ. The optimum value depends on the load being measured. If you have a poorly chosen R_{ref} value, the answer you get for $Z_{test}(f)$ will be

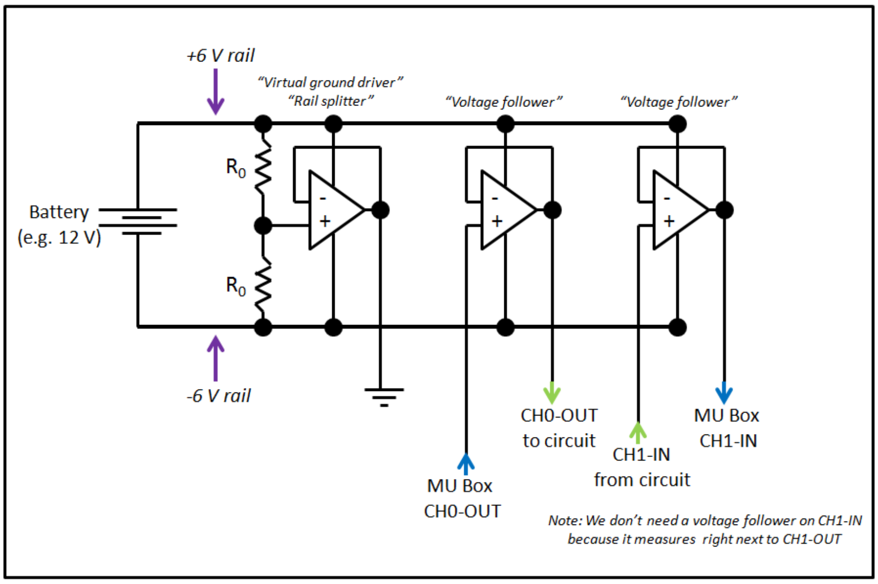


Figure 4: Circuit to power the op-amps. We split the battery voltage to power the op-amps using a voltage divider (e.g. $R_0 = 1[\text{k}\Omega]$).

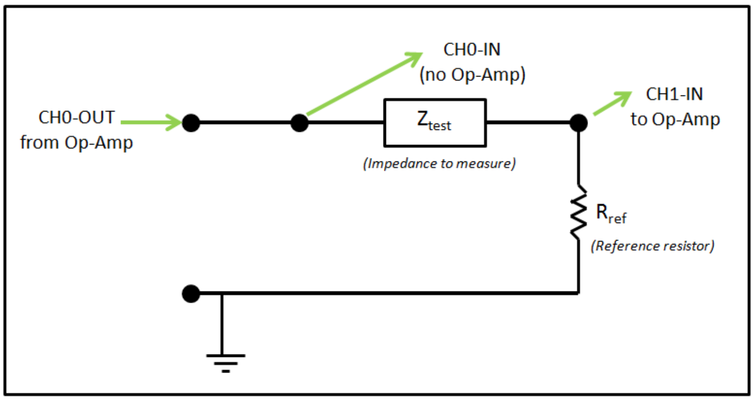


Figure 5: Circuit to measure a test impedance. The reference resistor R_{ref} should have an impedance magnitude similar to the test impedance.

noisy. Be sure to look at the ratio of the two impedances (this requires a guess regarding Z_{test}, or an initial noisy test, followed by a new test with a better chosen R_{ref}). If the ratio of the impedances is either too small or large compared to 1, there is a problem that you need to resolve; either the two measured voltages are almost the same ($Z_{test}/R_{ref} \to 0$), and or the current is very small ($Z_{test}/R_{ref} \to \infty$).

To test/debug this circuit:

(a) Place a "test resistor" (of known impedance) in series with the reference resistor R_{ref}. Your test resistor should occupy the place of Z_{test} in Figure 5.

(b) Drive the circuit from CH0-OUT of the MU-box. *NOTE: If you do not use an Op-amp with a low impedance output (0 [ohms]), as described in Fig. 4, your results will not be as good as they could be.*

(c) Use two inputs (into the MU-box), taken across the 'test resistor' terminals. CH0-IN should be connected to the "high" side of the speaker, and CH1-IN connected at the place where the speaker and reference resistor are joined, as described in Fig. 5.

(d) Run the Matlab program `run_g7play.m` to play a wideband chirp on CH1-OUT (*NOTE: The G7-software package we distributed plays chirps on both output channels by default. Since CH2-OUT is not hooked up, this is not a problem*). The program `show_data.m` computes the impedance using the equations given above:

   i. Make sure to set the reference resistor value `Rref` in the code to match the actual reference resistor in your circuit

   ii. Set the Boolean flag `plot_impedance=1;` so you can view the impedance you are trying to measure.

If all goes well, your impedance plot should show a magnitude impedance matching your test resistor (constant across frequency); the impedance should be real, meaning its phase is approximately 0 for all frequencies. In future labs, you should use this test resistor method to check that your circuit is working, before measuring an unknown impedance (e.g. the loudspeaker).

Lab 2 (Feb 16)

NOTE: Every time you set up your lab circuit for the day, we suggest that you do a quick test with a resistor, as in Lab 1 (Fig. 5) to make sure everything is running properly!!!

For this lab we will measure *2-port* networks (e.g. networks having both an input and an output). You will analyze the data you collect in this lab as part of Homework B (assigned Feb 7, DUE Feb 21).

By definition a *1-port* network has two terminals (signal and ground), while a 2-port has 4 terminals, two of which are likely connected to a common *ground* point. Thus in practice most 2-port networks (or simply 2-ports) are three wires, broken out as 4 wires. For example, a transformer is a true 2-port (four wire) network. An FET transistor is a 3-terminal network. A diode and a resistor are 1-port (2-terminal) networks.

A 1-port may be converted into a 2-port by placing a resistor in series with it, allowing one to simultaneously measure the current through it, and the voltage across it. When properly measured, one may find the complex impedance, defined as

$$Z(f) = \frac{V(f)}{I(f)}, \tag{1}$$

where $V(f)$ is the complex voltage and $I(f)$ is the complex current across the network we wish to characterize.

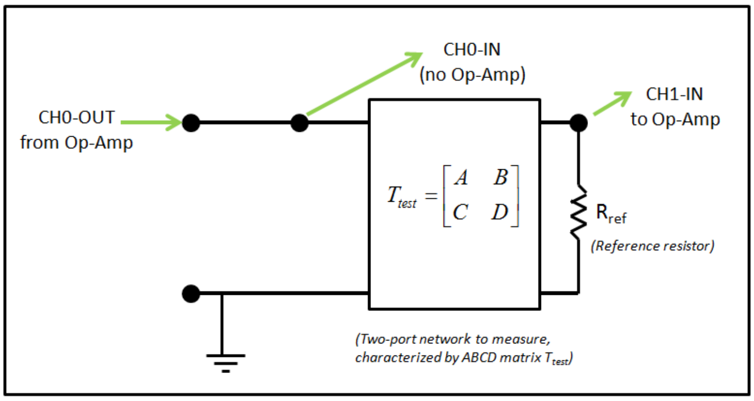

CH0-IN
(no Op-Amp)
CH1-IN
to Op-Amp
CH0-OUT
from Op-Amp
$T_{test} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$
R$_{ref}$
(Reference resistor)
(Two-port network to measure, characterized by ABCD matrix T_{test})

Figure 6: Circuit to measure a 2-port network. The reference resistor R_{ref} should have an impedance magnitude similar to the test impedance.

You will measure two networks provided to you, a **low-pass filter** and a **twin-T** network. Your test circuit should be a slightly modified version of the circuit from Lab 1 (Fig. 5), as shown in Figure 6. The primary difference is that the unknown two-port network you want to test has a ground terminal. The relationship between the two sides of the unknown two-port is given by the transmission matrix

$$T_{test} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \qquad \begin{bmatrix} V_0 \\ I_0 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V_1 \\ -I_1 \end{bmatrix}$$

where both currents are define as pointing into the port (e.g. I_0 flows to the right, I_1 flows to the left). This transmission matrix is related to the impedance matrix by

$$Z_{test} = \frac{1}{C} \begin{bmatrix} A & \Delta_T \\ 1 & D \end{bmatrix} \qquad \begin{bmatrix} V_0 \\ V_1 \end{bmatrix} = \frac{1}{C} \begin{bmatrix} A & \Delta_T \\ 1 & D \end{bmatrix} \begin{bmatrix} I_0 \\ I_1 \end{bmatrix}$$

where $\Delta_T = AD - BC$ is the determinant of the transmission matrix T.

Lab 3 (Feb 23)

NOTE: Every time you set up your lab circuit for the day, we suggest that you do a quick test with a resistor, as in Lab 1 (Fig. 5) to make sure everything is running properly!!!

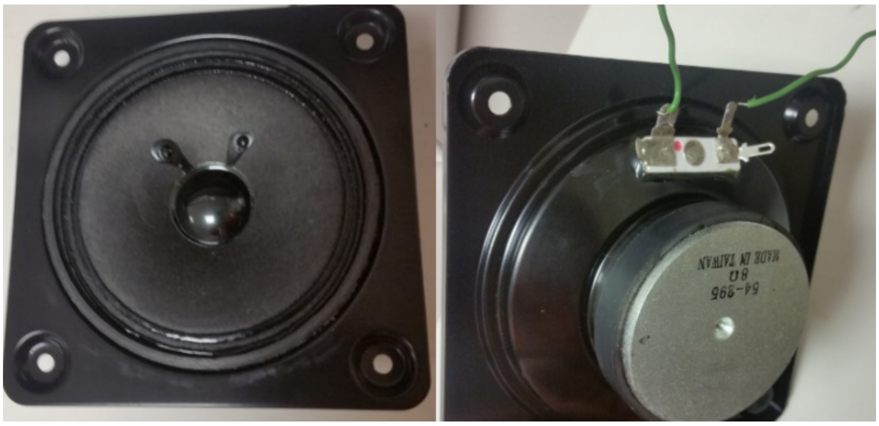


Figure 7: The loudspeakers we will measure in this lab.

This is the first lab to work with a real loudspeaker, shown in Figure 7. These speakers are of low quality (the cone is made of cardboard!), but you will still be able to recognize audio signals you play through them. In this lab you will measure the electrical input impedance Eq. 1, where V is the voltage across the speaker and I is the current through it. The loudspeaker will be in **series** like your resistor from Lab 1, so you should connect the loudspeaker according to Figure 5. The loudspeaker has two terminals; the direction of the terminals should not matter.

A typical speaker impedance magnitude $|Z_{spkr}(f)|$ is given in the text. You should use this value to choose an initial value for R_{ref}. Remember, if your measurement is noisy you should check your op-amps and/or adjust your value of R_{ref}. When measuring and analyzing the loudspeaker impedance, do not forget that $Z_{spkr}(f)$ is complex, and it is necessary that the real part be strictly non-negative. In practice, it must be positive. This means that in the equation $Z_{spkr} = R_{ref}(V_0/V_1 - 1)$ we require $|V_0/V_1| > 1$.

Lab 4 (Mar 16)

Reciprocity calibration of the LS sensitivity, using a probe-microphone to verify the results. This is a famous method because it was first used to calibrate a microphone, before any other was available as an acoustic standard.

Two LS (sources) are place face to face, and the first is driven electrically, and the second is used as a microphone. This method requires the following to be true:

1. The two LS are identical in there characteristics.
2. There is a know acoustic load impedance between the two speakers (normally assumed to be a compliance C, as determined by the volume of air trapped between the two).
3. the two LS are *reciprocal*, as defined in our lectures. This means that the ratio of the voltage $V_2(f)$ at the unloaded voltage at the second loudspeaker, given the current $I_1(f)$ at the first loudspeaker, is identical when the system is reversed in direction.

Lab 5 (Mar 30)

Next we measure the acoustic Thévenin impedance of the loudspeaker. This is much more difficult than measuring the electrical impedance. In fact it is hard enough that you may not be successful. But we will try none the less.

The theory here is to measure the speaker pressure at the input to two acoustic loads, and from this, in theory, one may calculate the source impedance and pressure. This method uses the *voltage divider equation* (i.e., pressure divider equation, for acoustics). This equation says that if two known impedances, in series, are placed across a voltage (i.e, pressure) source, then the volume-velocity is

$$U(f) = \frac{P_s(f)}{Z_s(f) + Z_l(f)},$$

where $P_s(f)$ is the unknown *Thévenin source pressure*, $Z_s(f)$ is the unknown *Thévenin source impedance*, $Z_l(f)$ is one of the two the load impedances (both assumed to be known), and $U(f)$ is the resulting unknown volume velocity.

The pressure at the input to the cavity is measured with a microphone, for each load impedance. This results in the linear relationship

$$P_l(f) = P_s(f)\frac{Z_l(f)}{Z_s(f) + Z_l(f)}. \tag{2}$$

There are two such equations, one for each load condition $Z_l(f)$ for $l = 1, 2$.

To find $P_s(f), Z_s(f)$ we must solve a 2x2 system of equations

$$\begin{bmatrix} -Z_1 & P_1 \\ -Z_2 & P_2 \end{bmatrix} \begin{bmatrix} P_s \\ Z_s \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix}. \tag{3}$$

This equation comes from the reorganization of Eq. **??**. Taking the inverse of this equation gives the source parameters

$$\begin{bmatrix} P_s \\ Z_s \end{bmatrix} = \frac{1}{\Delta} \begin{bmatrix} P_2 & P_1 \\ Z_2 & Z_1 \end{bmatrix} \begin{bmatrix} P_1 \\ -P_2 \end{bmatrix}, \tag{4}$$

where the determinate $\Delta(f) = P_1 Z_2 - P_2 Z_1$.

Lab 6 (Apr 11)

Next we attempt to measure the radiation impedance. Of course finding this requires knowledge of the Thévenin source parameters. Now Eq.2 holds, but the meaning of the variables changes. The load impedance in this case is the radiation pressure, which we can directly measure, while the radiation volume velocity must be estimated from Eq. 2.

Lab 7 (Apr 25)

Finally we have a lab where you get to pick one of three different things to measure. The first option is switching to an ear-bud and measure the pressure in your ear (or an ear simulator). The second option is to make a horn baffle (out of cardboard) and look at the speaker performance with and without the baffle.

The third option is to put the speaker in a sealed box, of just the right size, so that the air volume behind the speaker acts like a spring. This configuration is call an Acoustic suspension loud-speaker, first make popular the Edward Villcher, sometime around the 1950s, with his famous AR-3 (the name of the company was Acoustics Research).[5]

---

[5] `http://www.vintage-speaker-review.com/2014/03/Acoustic-Research-AR-3-Speaker-Review-Specs.html`