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

Univ. of Illinois **Version 1.52 April 5, 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 26)
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

---

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

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

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).

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 something other 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 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).

---

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

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]).

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.

1 Lab 1 (Feb 9)

1.1 Part 1: Installing the G7 Software package

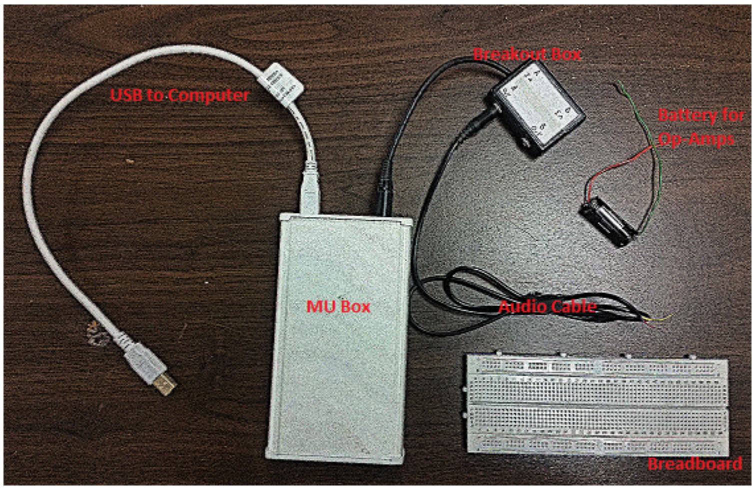


Figure 1: Some equipment you will use in the labs.

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.

---

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

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. 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.

1.1.1 `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 tests 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.

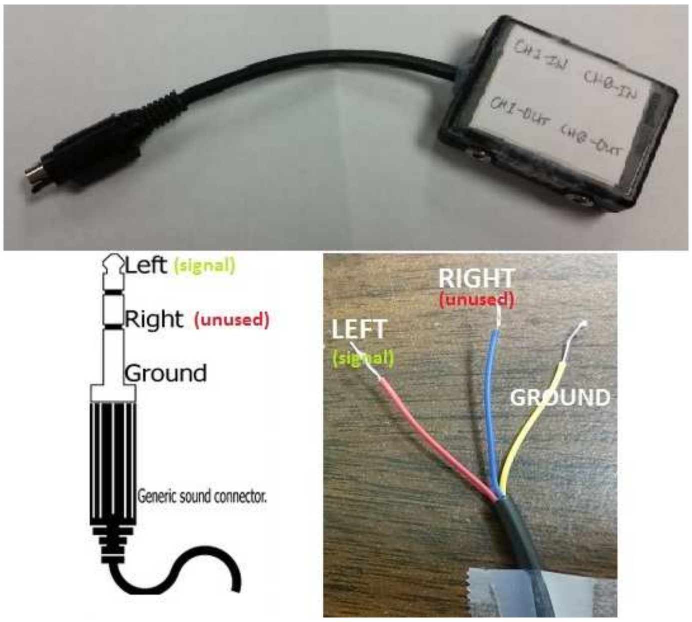


Figure 3: Description of 3-wire audio cables and breakout box. Each channel on the breakout box has two terminals: signal and ground. Because the audio cables are for stereo sound, we have one unused wire on each cable. You will connect the signal and ground wires to your breadboard.

1.1.2 Interlude: Getting to know the ‘Breakout Box’

The ‘breakout box’ takes 4 headphone jack (3-contact, tip-ring-sleeve (TRS) 3.5 [mm] plug phone connector) 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 audio 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).

1.2 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.

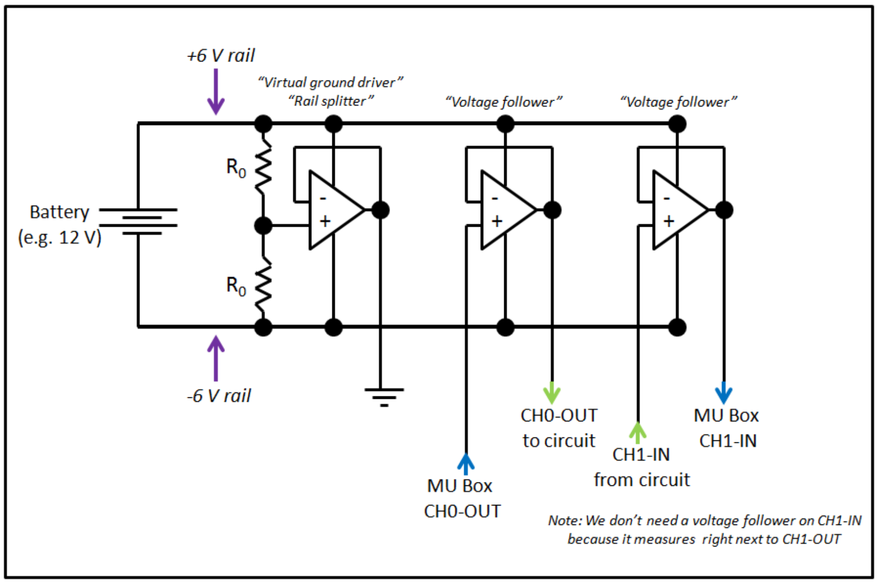


Figure 4: Circuit to power the op-amps. We split the battery voltage to power the op-amps using a voltage divider (e.g. $R_0 = 10$ [kΩ]).

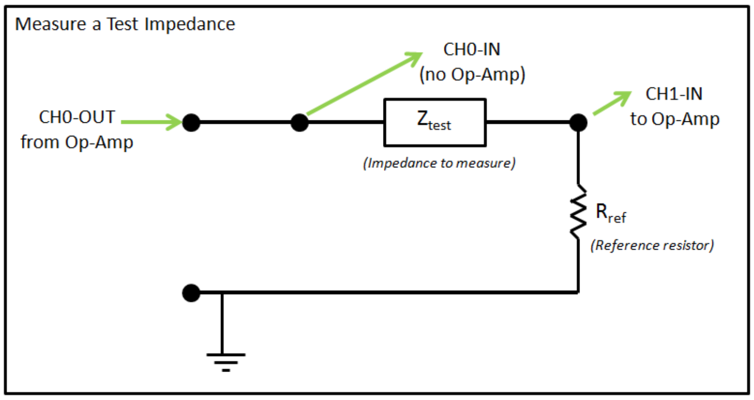


Figure 5: Circuit to measure a test impedance. The reference resistor R_{ref} should have an impedance magnitude similar to the test impedance.

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. To measure an impedance (V/I) we need to know the current, but the MU box measures voltages. Using a voltage divider with a reference resistor R_{ref}, you can calculate the current, and thus 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 reference 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 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 ($Z_{test}/R_{ref} \rightarrow 0$), and or the current is very small ($Z_{test}/R_{ref} \rightarrow \infty$).

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).

2 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.

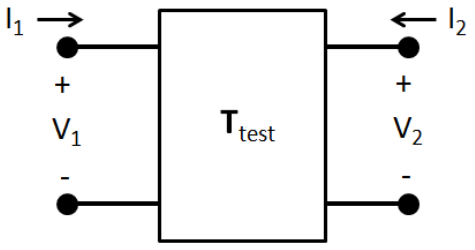
We can measure the input impedance of one of the two ports by placing a resistor in series with it, allowing one to simultaneously measure the current through it, and the voltage across it. When properly measured, one may find the complex input impedance, defined as

$$Z_{in}(f) = \frac{V_{in}(f)}{I_{in}(f)}, \tag{1}$$

where $V_{in}(f)$ is the complex voltage and $I_{in}(f)$ is the complex current across the input terminals of the network we wish to characterize.

2.1 A brief review of 2-port matrices

NOTE: We will use the subscript '1' to represent the left side of an arbitrary 2-port network, and subscript '2' to represent the right side. These subscripts do not correspond to CH0 and CH1 of the breakout box (this depends on your measurement circuit configuration)!!!



(Two-port network to measure, characterized by ABCD matrix T_{test})

Figure 6: Both currents are defined as into the ports. Thus, when we look at the ABCD matrix representation, you will usually see a negative current $-I_2$, which is the current flowing out of port 2, to the right.

The relationship between the two sides of the unknown two-port is given by the transmission matrix

$$T_{test} = \begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{bmatrix} \qquad \begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{bmatrix} \begin{bmatrix} V_2 \\ -I_2 \end{bmatrix} \tag{2}$$

where both currents are define as pointing into the port as in Figure 6 (e.g. I_1 flows to the right, I_2 flows to the left). This transmission matrix is related to the impedance matrix by

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \frac{1}{\mathcal{C}} \begin{bmatrix} \mathcal{A} & \Delta_{\mathbf{T}} \\ 1 & \mathcal{D} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \tag{3}$$

where $\Delta_{\mathbf{T}} = \mathcal{A}\mathcal{D} - \mathcal{B}\mathcal{C}$ is the determinant of the transmission matrix $\mathbf{T}_{test}$.

2.1.1 Reversibility of the Transmission matrix

Reversing the direction of the network, by taking the matrix inverse, and reversing the signs of I_1, I_2, results in

$$\mathbf{T}_{rev} = \begin{bmatrix} V_2 \\ I_2 \end{bmatrix} = \frac{1}{\Delta_{\mathbf{T}}} \begin{bmatrix} \mathcal{D} & \mathcal{B} \\ \mathcal{C} & \mathcal{A} \end{bmatrix} \begin{bmatrix} V_1 \\ -I_1 \end{bmatrix} \tag{4}$$

Thus if $\mathcal{A} = \mathcal{D}$ and $\Delta_{\mathbf{T}} = 1$, the matrix is reversible. The low-pass filter and twin-T networks, analyized below, are examples of reversible networks. Note that if $\Delta_T = -1$, the network *cannot* be reversible (we would require $\mathcal{A} = \mathcal{D} = -\mathcal{A} = -\mathcal{D}$ and $\mathcal{B} = -\mathcal{B}$, $\mathcal{C} = -\mathcal{C}$, so there is no non-trivial solution). A moment's thought should convince you that a loud speaker is not reversible, since $\mathcal{A} \neq \mathcal{D}$ and $\Delta_T = -1$.

2.1.2 Reciprocity of the Transmission matrix

The determinant of every passive transmission matrix is either +1 or -1. If the system is *reciprocal* the $\Delta_{\mathbf{T}} = 1$. If the system is anti-reciprocal then $\Delta_{\mathbf{T}} = -1$. For example, an ideal transform, having a turns ratio of N, has the $\mathbf{T}$ matrix

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} N & 0 \\ 0 & 1/N \end{bmatrix} \begin{bmatrix} V_2 \\ -I_2 \end{bmatrix}.$$

The ideal transformer is *reciprocal* since the determinant is 1. Also a transformer is not *reversible* ($\mathcal{A} \neq \mathcal{D}$), and it is *passive* and *loss-less*.

A gyrator having $T_o = lB_o$ has a $\mathbf{T}$ matrix

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} 0 & T_o \\ 1/T_o & 0 \end{bmatrix} \begin{bmatrix} V_2 \\ -I_2 \end{bmatrix}.$$

Thus a gyrator is anti-reciprocal since the determinant is -1. Any circuit containing a single gyrator will be anti-reciprocal. It is also *passive* and *loss-less*.[5]

2.2 Measuring 2-port parameters

You will measure two networks provided to you, a **low-pass filter** and a **twin-T** (notch filter) network. Here are some tips:

- You have the ability to measure **voltages** directly, using the MU box
- If you wish to measure a **current**, you will need a reference resistor of some sort (i.e. by measuring the voltage across a known resistor, you can calculate the current). This can be like your circuit in Fig. 5, or you can measure the voltage across a known resistor inside the 2-port you are trying to characterize
- To find impedance and ABCD matrix values, often you will want to **open-circuit** ($I = 0$) or **short-circuit** ($V = 0$) across the output (or input) terminals of the 2-port. Consider the matrix equations given in the previous section: each matrix equation gives 2 equations, each in terms of 3 out of the 4 quantities V_1,V_2,I_1,and I_2. By short- or open-circuiting one of the ports, you can generate relationships between 2 of the 4 quantities, and the matrix elements.

---

[5] Magnetic-flux diffusion effects introduce losses and nonlinear effects (Kim and Allen, 2013).

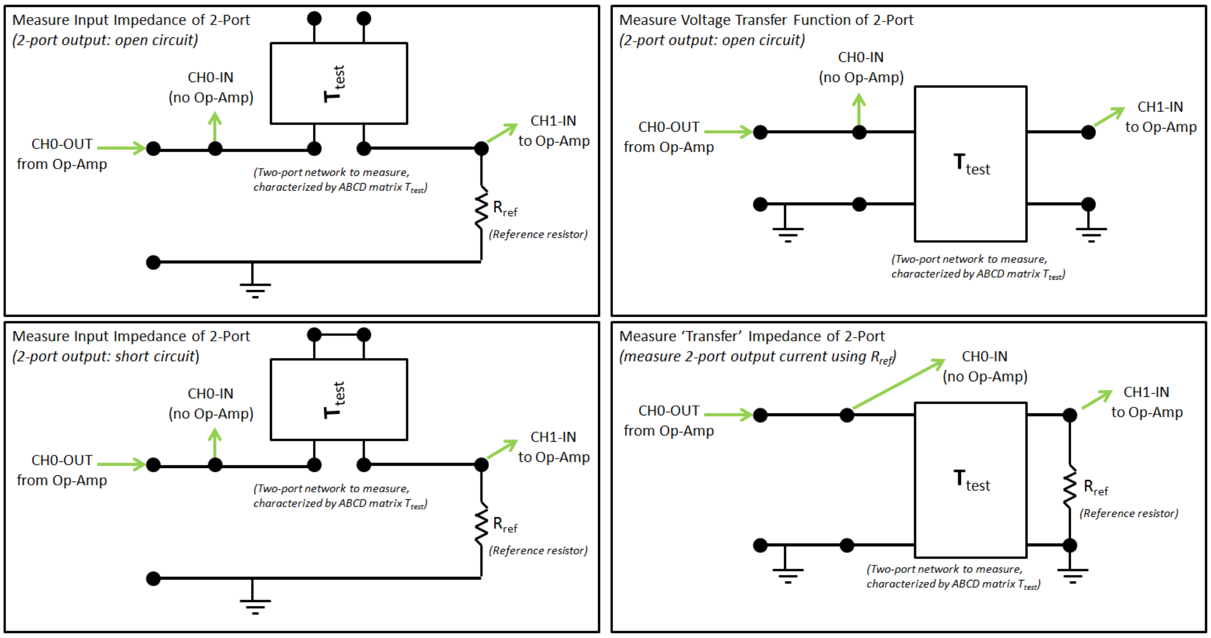


Figure 7: Circuits to measure the input impedance and transfer function of a 2-port network. When using a reference resistor R_{ref}, remember it should have an impedance magnitude similar to the test impedance.

Figure 7 shows some test circuits you can use to find the elements of the ABCD matrix, or the impedance matrix (z_{11}, z_{12}, z_{21}, z_{22}) (*NOTE: Once you know all elements of one matrix, you can solve for the other.*). There are other ways you can measure all four elements, this is only one option. Considering Figure 7:

- **Top-Left**: (Input impedance = V_1/I_1) Measure V_1 and I_1 with $I_2 = 0$ (port 2 open-circuited). From Equation 3 this gives
$$\left.\frac{V_1}{I_1}\right|_{I_2=0} = z_{11} = \frac{A}{C}$$

- **Bottom-Left**: (Input impedance = V_1/I_1) Measure V_1 and I_1 with $V_2 = 0$ (port 2 short-circuited). From Equation 4 this gives
$$\left.\frac{V_1}{I_1}\right|_{I_2=0} = \frac{B}{D}$$
If we take the inverse of the Z matrix in Eq. 3, we can also show this quantity is equal to Δ_Z/z_{22}.

- **Top-Right**: (Voltage transfer function = V_2/V_1) Measure V_1 and V_2 with $I_2 = 0$ (port 2 open-circuited). From Equation 2 this gives the voltage transfer function
$$\left.\frac{V_2}{V_1}\right|_{I_2=0} = \frac{1}{A}$$

- **Bottom-Right**: (Transfer impedance = V_1/I_2 (or V_2/I_1))

Method 1: Measure V_1, V_2, and I_2 as in the Figure. Using a reference resistor, we find

$$-I_2 = V_2 R_{ref}$$

From Equation 2 we have

$$V_1 = AV_2 - BI_2$$

Thus, putting these equations together we have

$$\left.\frac{V_1}{I_2}\right|_{V_2=-I_2/R_{ref}} = -\frac{A}{R_{ref}} - B$$

Method 2: Short circuit the output terminals of the 2-port to measure

$$\left.\frac{V_1}{I_2}\right|_{V_2=0} = -B$$

To get I_2 in this case, you must use a reference resistor that is internal to the network (this might not be possible for every type of 2-port; you can't do it for the Twin-T). For example, in the case of the low-pass filter, the resistor R_2 is directly in series with the terminal on the right-hand side. By measuring the voltage across this resistor (e.g. this resistor becomes R_{ref} in your calculation of I_2, since it is followed by a short to ground), you can calculate V_1/I_2.

Thus, we have 4 equations in 4 unknowns (A, B, C, and D), so we can solve for the impedance and transmission matrices.

NOTE: You can also get all 4 elements by measuring the input impedance and voltage transfer function (Fig. 7 top-left and top-right) of the network, then swapping the ports and making the same 2 measurements. Can you figure out how to do this?

NOTE: The networks we are measuring in the lab have some symmetry - you can use known symmetry to decrease the number of measurements you have to make. Can you figure out how to do this?

3 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!!!