IMDEA

Semester 3

TRANSDUCERS MEASUREMENT

Work 1:
Loudspeaker Parameters Measurement

Antonin Novak & Pierrick Lotton

This experiment involves measuring and analyzing the linear parameters of electrodynamic loudspeakers and their frequency dependency. It is closely related to the Transducers Measurements lecture (CM). For more detailed theory, refer to [1].

For a thorough understanding, we strongly recommend you write your own code. However, if needed, you can refer to the Python codes developed during the lectures available on the *umtice* platform.

Equipment

- Data Acquisition Module NI USB-4431
- Audio amplifier
- Displacement sensor Panasonic HG-C1030-P (sensitivity = 2 mm/V)
- Z-box circuit for current and voltage measurement
- Electrodynamic loudspeaker

Preparations

- Prepare the measurement bench as depicted in Fig. 1.
- Position the displacement sensor approximately 5 cm in front of the speaker, with the laser targeted at the center of the loudspeaker membrane.
- Adjust the distance between the laser and the displacement sensor until the sensor displays a value close to 0.
- Connect the USB-4431 device to the computer. Connect the output `AO_0` of the USB-4431 device to the amplifier. Connect the measured quantities U_{HP} to the input `AI_0`, U_R to `AI_1`, and the displacement $x(t)$ (output of the displacement sensor) to `AI_2`.
- Warning! The displacement sensor introduces a significant signal delay of approximately 1 ms, which may require compensation.
- Warning! The displacement sensor has a limited frequency range (DC to 500 Hz).

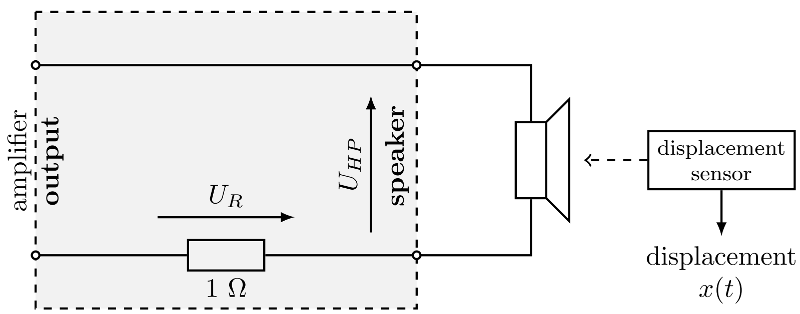

Fig. 1: Block diagram of the experimental setup for the loudspeaker measurement.

1 Measurement

Open and run the Python file `TP01_01_measurement.py`. This file contains a basic script that generates a swept-sine signal, sends it to the output `AO_0` of the USB-4431 device, acquires the measured signals, calculates the spectra of all the quantities, and prepares the spectral variables `U`, `I`, and `X`.

You would need to specify the name of the USB-4431 device recognized by your operating system. On the System Tray (the menu with small icons on the bottom right-hand corner of the Windows taskbar), find the icon , click on it, and note the device number (e.g., Dev4). Modify the line with the device name in the Python file as follows:

```
""" Parameters """
Dev = 'Dev4'  # name of the NI device
```

This script should perform the measurement. You should be able to hear the loudspeaker playing the swept sine, and it will also save the data to `results/meas_data.npz` file and plot the results.

2 Parameters Fitting

Open the Python file `TP01_02_fitting.py`. This file **loads your previously saved measurement results.** Run the file to verify that you can see the measurement results.

2.1 Electrical part

To estimate the parameters from the measured data, use the following equation

$$U(\omega) = R_e + j\omega L_e I(\omega) + B\ell V(\omega) \tag{1}$$

which can be written in matrix form as

$$\mathbf{U} = \mathbf{A} \begin{bmatrix} R_e \\ L_e \\ B\ell \end{bmatrix}, \tag{2}$$

where the matrix $\mathbf{A}$ is given by

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & j\omega\mathbf{I} & \mathbf{V} \end{bmatrix}. \tag{3}$$

To solve this equation numerically in Python, use the following code which will give you a `solution` variable that contains the unknown parameters R_e, L_e, and $B\ell$.

```
# Create matrix A
A = np.array([I, 1j*omega*I, V]).T

# Solve for the parameters [Re, Le, Bl] using least squares method
solution = np.linalg.lstsq(A, U, rcond=0)[0]
```

As the loudspeaker model may not be perfect and the spectral values U, I, and V are complex, you may obtain a complex-valued solution. To address this, separate the equation into its real and imaginary parts and perform the fitting separately:

```python
# Solve for the parameters using real part only
solution_re = np.linalg.lstsq(np.real(A), np.real(U), rcond=0)[0]
```

- Compare the solution `solution_re` with an imaginary solution and comment on the results.
- Are the estimated parameters R_e, L_e, and B_ℓ reasonable?

Even with the separation of real and imaginary parts, the results may not be accurate if you don't limit the frequency range of your data. To ensure better accuracy, limit the frequency range by specifying a minimum (f_{min}) and maximum (f_{max}) frequency (see the code below):

```python
# Define the frequency range
f_min = 20
f_max = 500

# mask the frequency range
mask = (f_axis > f_min) & (f_axis < f_max)

# Solve for frequency limited values
solution_re = np.linalg.lstsq(np.real(A[mask,:]), np.real(U[mask]),
    rcond=None)[0]
```

- Are the estimated parameters R_e, L_e, and B_ℓ reasonable?
- Change the frequency `f_max` to 1000 and than to 300 to observe its influence on the estimated parameters.

2.2 Mechanical part

Repeat the previous step for the second loudspeaker equation

$$B\ell I(\omega) = M_{ms} j\omega V(\omega) + R_{ms} V(\omega) + K_{ms} \frac{V(\omega)}{j\omega}, \tag{4}$$

and find the parameters M_{ms}, R_{ms}, and K_{ms}.

- Are the estimated parameters M_{ms}, R_{ms}, and K_{ms} reasonable?
- If not, check that:
  - you have compensated for the delay of the displacement sensor,
  - and the frequency range is limited around the resonance frequency of the loudspeaker.

2.3 Final model

Once you have estimated all the parameters, you can compare the measured input impedance with the one based on the model. Plot both amplitude and phase of loudspeaker input impedance.

3 Advanced Method

Use the Python file `TP01_03_advanced.py` for this part. Copy the lines from the previous file where you defined the velocity `V`, the angular frequency `omega`, and where you compensated for the delay of the displacement sensor.

3.1 Electrical Impedance and $B\ell$ factor

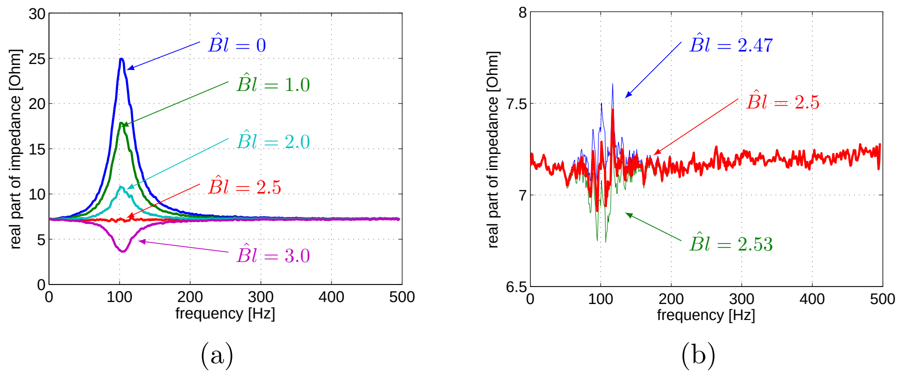

Fig. 2: An example of the $B\ell$ estimation and separation of electrical and mechanical impedance. The basic principle of the method was explained during the lecture. a) rough estimation of $B\ell$, b) $B\ell$-refinement.

Estimate the following quantities from the measured data :

- Force factor $B\ell$ (the principle of $B\ell$ estimation and impedance separation is illustrated in Figure 2). To do this, vary $B\ell$ and plot the real and/or imaginary part of

$$Z_e(f) = \frac{U(f)}{I(f)} - Bl\frac{V(f)}{I(f)}. \tag{5}$$

  Try to remove the delay compensation to see the effect of the delay on the results.

- Apparent resistance $R_e(f)$ (function of frequency up to 20 kHz).

$$R_e(f) = \text{real}\left\{Z_e(f)\right\}, \tag{6}$$

- Apparent inductance $L_e(f)$ (function of frequency up to 20 kHz)

$$L_e(f) = \frac{\text{imag}\left\{Z_e(f)\right\}}{\omega}, \tag{7}$$

  Note that if the results are too noisy at high frequencies

3.2 Electrical impedance model

Fit the Leach, R2L2, and R3L3 models on the measured electrical impedance and compare the model with the measurement. Use the class `Electrical_Impedance` that you can import as follows:

```
from functions.Electrical_Impedance import Electrical_Impedance
```

Inspect the file `functions/Electrical_Impedance.py` and its documentation to understand how to use it. Note that for the estimation, you would need to limit the frequency axis (approximately from 20 Hz to 20 kHz) to exclude useless data (outside the bounds of the swept-sine). You can use the same approach with `mask` as in the previous section.

3.3 Mechanical Impedance

Estimate the following quantities using the mechanical impedance $Z_{ms}(f)$, which is derived from the known force factor $B\ell$, current I, and velocity V. Note that the frequency range is limited to DC - 500 Hz due to the displacement sensor's frequency limits and the piston motion condition.

- Apparent mechanical resistance $R_{ms}(f)$,

  $$R_{ms}(f) = \text{real}\left\{Z_{ms}(f)\right\}, \qquad (8)$$

  and provide an explanation for the variations.

- Mass M_{ms} and K_{ms} using a fit.

3.4 Final model

Once you have estimated all the parameters, you can compare the measured impedance with the one based on the model.

- For mass M_{ms} and stiffness K_{ms} take the values from the fit, for R_{ms} take a value near the resonance frequency.
- For the electrical part use:
  1. Leach model.
  2. R2L2 model.
  3. Simple Thiele-Small model with R_e and L_e parameters. For R_e, use the value estimated from very low frequencies. For L_e, experiment with **two values** found on the graph of apparent inductance: one near the resonance frequency and another near high frequencies.

Plot both the amplitude and phase of the measured and modeled loudspeaker impedance, and then conclude which of the models fits the best.

4 Report Demands

Write a report (approximately 3 to 4 pages) and focus on:

- Describing the measurement setup.
- Discussing the limitations of both tested methods.
- Providing an explanation for the frequency variations observed in $R_e(f)$ and $L_e(f)$ [2, 3, 4].
- Providing an explanation for the frequency variations observed in $R_{ms}(f)$ [5, 6, 7].
- Provide a comment on the choice of the model for the electrical part.

References

[1] Novak, Antonin, "Measurement of loudspeaker parameters : A pedagogical approach.", *In International Congress on Acoustics 2019*, Aachen, Germany, pp. 2-–9.

[2] Vanderkooy, John, "A model of loudspeaker driver impedance incorporating eddy currents in the pole structure.", *J. Audio Eng. Soc.*, vol. 37, no. 3, pp. 119–128, 1989.

[3] Leach Jr, W. Marshall, "Loudspeaker voice-coil inductance losses: circuit models, parameter estimation, and effect on frequency response.," *J. Audio Eng. Soc.*, vol. 50, no. 6, pp. 442–450, 2002.

[4] Thorborg, Knud, and Andrew D. Unruh, "Electrical equivalent circuit model for dynamic moving-coil transducers incorporating a semi-inductor.", *J. Audio Eng. Soc.*, vol. 56, no. 9, pp. 696–709, 2008.

[5] Knudsen, M. H., and Jensen, J. G., "Low-Frequency Loudspeaker Models That Include Suspension Creep.", *J. Audio Eng. Soc.*, vol. 4, no. 1-2, pp. 3—18, 1993.

[6] Thorborg, K., Tinggaard, C., Agerkvist, F. T., and Futtrup, C., "Frequency dependence of damping and compliance in loudspeaker suspensions.", *J. Audio Eng. Soc.*, vol. 58, no. 6, pp. 472-–486, 2010.

[7] Novak, A., "Modeling viscoelastic properties of loudspeaker suspensions using fractional derivatives.", *J. Audio Eng. Soc.*, vol. 64, no. 1–2, pp. 35-–44, 2016.