SASP Homework 4

May 17, 2022

The topic of this homework is Wave Digital Filter (WDF) modeling. You will be required to model a three-way crossover network in the Wave Digital (WD) domain starting from a reference analog circuit.

Electroacoustic transducers are often characterized by three drivers; a woofer, a midrange speaker and a tweeter. A crossover network is employed to filter the input audio signal such that low-frequency sounds are diverted to the woofer, mid-frequency sounds to the midrange speaker and high-frequency sounds to the tweeter. Let us consider the three-way crossover network shown in Fig. 1.

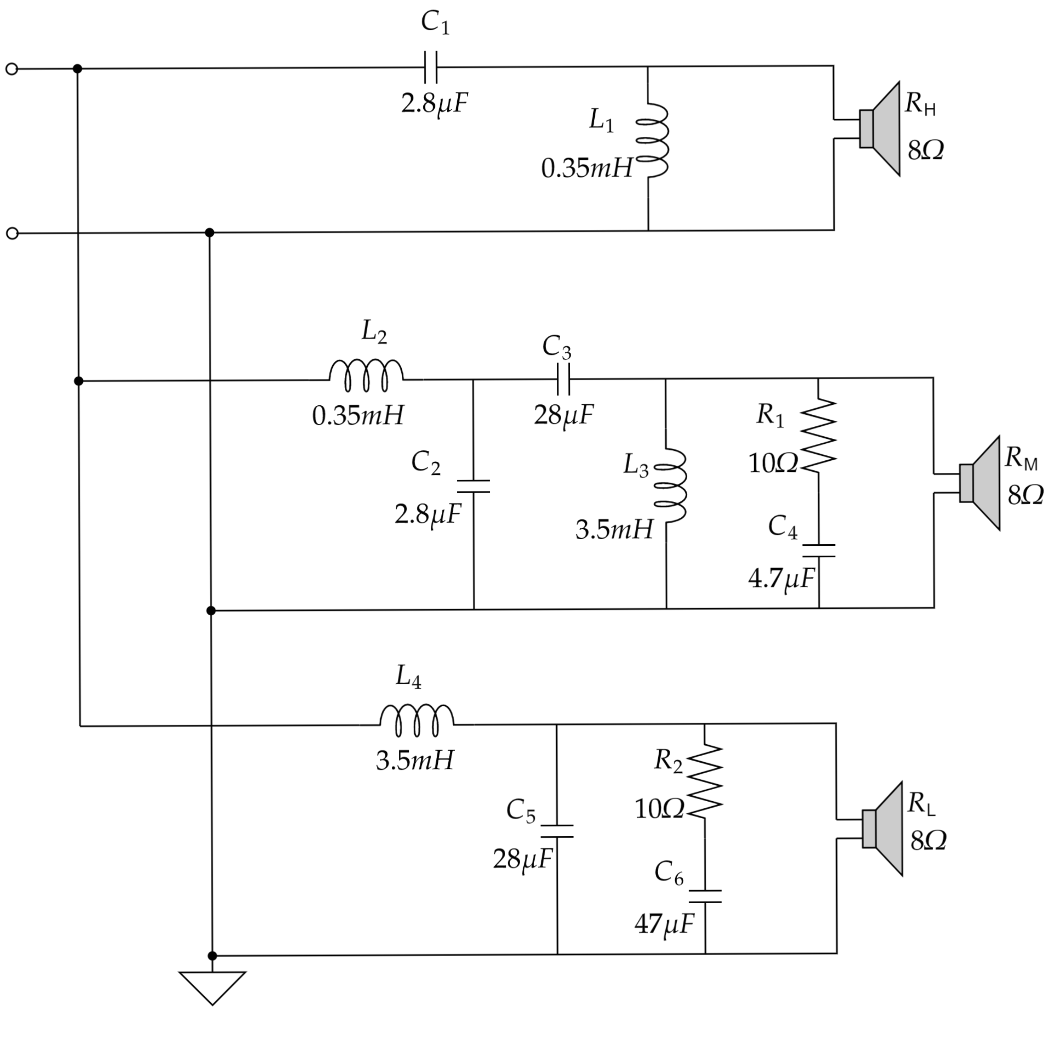

Figure 1: Crossover network.

The upper subcircuit is a high-pass filter connected to the tweeter speaker, which is modeled as a resistor with resistance $R_{\mathrm{H}} = 8\ \Omega$. The mid subcircuit is a band-pass filter connected to the midrange speaker, which is modeled as a resistor with resistance $R_{\mathrm{M}} = 8\ \Omega$. Finally, the subcircuit at the bottom is a low-pass filter connected to the woofer speaker, which is modeled as a resistor with resistance $R_\text{L} = 8$ Ω. All the elements of the circuit are linear.

In this homework you will be asked to:

- design a WDF implementation of the crossover network, providing a WDF scheme and indications on how to set the free parameters of the WDF in such a way that it can be implemented in an explicit fashion (i.e., without using iterative solvers);
- develop a MATLAB implementation of the designed WDF using the trapezoidal discretization method (bilinear transformation) for approximating continuous-time derivatives;
- answer a short list of questions.

You will be guided through this process step by step in the following sections.

1 Reference Circuit

The analog circuit that we will consider as a reference for designing the WD structure is obtained by augmenting the circuit in Fig. 1 with an *ideal voltage source* V_in at the input port. The result is reported in Fig. 2 which shows the reference circuit implemented in LTspice.

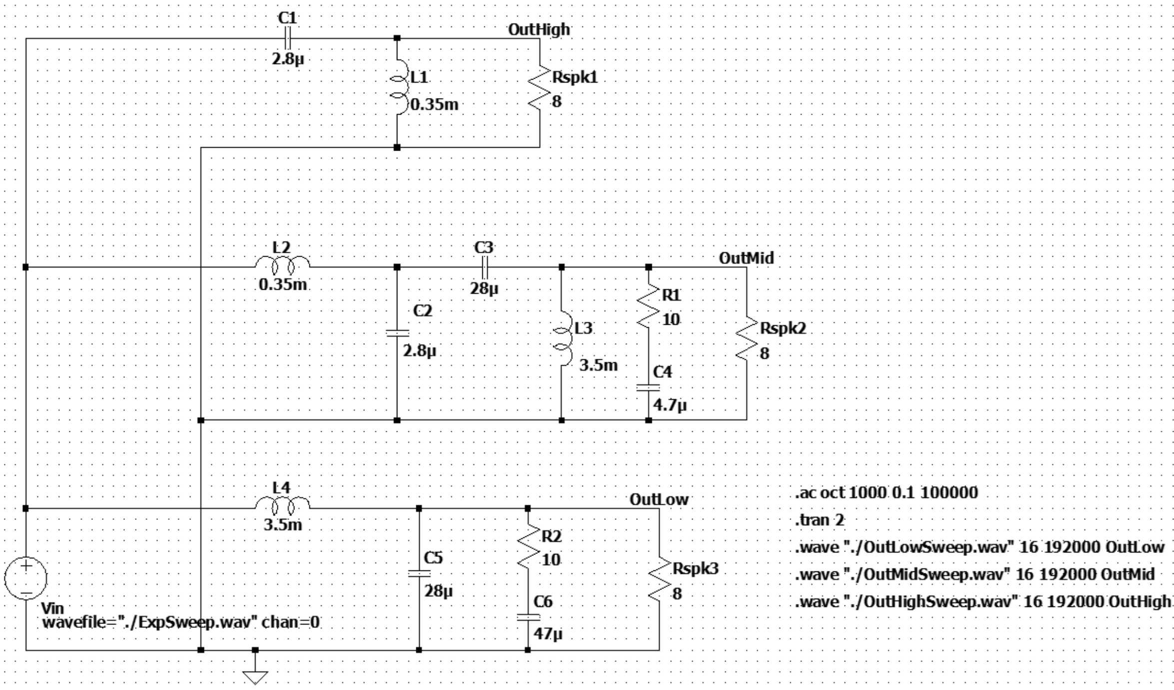

Figure 2: Reference circuit.

1.1 LTspice Implementation of the Reference Circuit (optional)

You are not expected or required to download LTspice and to install it in order to complete this homework and achieve the maximum grade. The input signal as well as the output signals of the reference circuit will be provided to you such that you can use them as a ground truth when you implement the required WDF. However, LTspice could help you to analyze the behavior of the reference circuit and search for a smart solution in designing the corresponding WDF structure. If you think that it can be useful, you can download LTspice for free at the following link `https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html`.

1.2 Input Signal

The model of the ideal voltage source element is not characterized by any internal series resistance, hence its constitutive equation in the continuous-time domain is the following

$$v(t) = V_{\text{in}}(t) \ , \tag{1}$$

where $v(t)$ is the port voltage of the ideal voltage source element and $V_{\text{in}}(t)$ is the input audio signal.

The input signal $V_{\text{in}}(t)$ is an exponential swept sine defined as [1]

$$V_{\text{in}}(t) = \sin\left(\frac{2\pi f_1 T_{\text{end}}}{\ln(f_2/f_1)}\left(\exp\left(\frac{t}{T_{\text{end}}}\ln(f_2/f_1)\right) - 1\right)\right) \ . \tag{2}$$

The exponential sweep in eq. (2) starts from frequency f_1, ends at frequency f_2 and lasts T_{end} seconds. We synthesize the signal in eq. (2) using the built-in MATLAB function `sweeptone`. In particular, the input signal is generated using the following command

```
Vin = sweeptone(1.99, 0.01, Fs, 'SweepFrequencyRange', [5 15000]);       (3)
```

The signal `Vin` is an exponential sweep starting from an initial frequency $f_1 = 5$ Hz, ending at a final frequency $f_2 = 15$ kHz and lasting $T_{\text{end}} = 1.99$ seconds. The exponential sweep is followed by 0.01 seconds of silence. The parameter `Fs` is the sampling frequency.

The obtained input signal is then written into an audio file called `ExpSweep.wav` using the MATLAB command

```
audiowrite('ExpSweep.wav',Vin,Fs);       (4)
```

and fed to the LTspice simulation as an input signal, as it is evident from Fig. 2. The file `ExpSweep.wav` is synthesized with a sampling frequency equal to 192 kHz.

1.3 Output Signals

The three output signals of the reference circuit are the voltage across resistor R_{spk1}, the voltage across resistor R_{spk2} and the voltage across resistor R_{spk3}, which, in the continuous-time domain, are referred to as $V_{\text{outLow}}(t)$, $V_{\text{outMid}}(t)$ and $V_{\text{outHigh}}(t)$, respectively. It is worth noticing that these three signals are also equal to the electric potentials (in volts) at the nodes of the reference circuit `OutLow`, `OutMid` and `OutHigh`, respectively, in accordance to Fig. 2. Such signals are then saved in the three audio files called `outlowsweep.wav`, `outmidsweep.wav` and `outhighsweep.wav`, respectively.

As we will provide you such output signals, you will be able to import them in MATLAB using the commands

```
[OutLowSpice,FsLTSpice]=audioread('outlowsweep.wav');
[OutMidSpice,FsLTSpice]=audioread('outmidsweep.wav');
[OutHighSpice,FsLTSpice]=audioread('outhighsweep.wav');
```

where the variable `FsLTSpice` is the sampling frequency of the audio files (i.e., 192 kHz). In this way you will be able to compare each output signal of the implemented WDF with a ground-truth.

2 Port-wise Definition of Wave Variables

The port-wise definition of wave variables considered in this homework is the same definition of voltage waves we have already seen in class. Therefore, Kirchhoff variables in the discrete-time domain at one port of an element are expressed in terms of wave variables as

$$v[k] = \frac{a[k] + b[k]}{2} \ , \quad i[k] = \frac{a[k] - b[k]}{2Z[k]} \ , \tag{5}$$

where $v[k]$ is the port voltage, $i[k]$ is the port current, $a[k]$ is the wave incident to the element, $b[k]$ is the wave reflected by the element and $Z[k]$ is a scalar free parameter different from zero.

3 WD Model of the Ideal Voltage Source

The WD scattering relation of an ideal voltage source is obtained by substituting the definition of port voltage (5) in terms of wave variables into the constitutive equation (1) and then solving for $b[k]$. Hence the discrete-time WD model of the ideal voltage source is the following

$$b[k] = 2V_{\text{in}}[k] - a[k] \ . \tag{6}$$

It is important to notice that an ideal voltage source with scattering relation (6) cannot be adapted; this means that there is no value of the free parameter $Z[k]$ that allows us to eliminate the instantaneous dependence between $b[k]$ and $a[k]$.

4 WDF Design

There are many possible ways of representing the reference circuit of Fig. 2 as a WDF structure, all leading to equally accurate results.

You are asked to choose one valid WDF representation of the reference circuit and draw the corresponding WDF scheme. For your convenience, you are encouraged to choose the WDF representation that, according to you, minimizes computational complexity (hint - notice that, since an ideal voltage source is connected to the input port of the crossover network circuit, the low-pass filter subcircuit, the band-pass filter subcircuit and the high-pass filter subcircuit can be analyzed/implemented separately without loss of accuracy, by connecting the same ideal voltage source to the three input ports of the three different subcircuits that are all in parallel). However, the only constraint on the required WD structure is that it must be computable in a fully explicit fashion; this means that no iterative solvers can be used to implement it in the WD domain.

You can propose a WDF based on one connection tree or more separated connection trees (e.g., one per subcircuit).

Connection trees will be characterized by

- a root (hint - the root should be the ideal voltage source since it cannot be adapted);
- one or more nodes (WD topological junctions called adaptors);
- many leaves (linear one-port elements that can be adapted).

As far as nodes of connection trees are concerned, you can use

- N-port series adaptors with $N \geq 3$.
  Remember that one port of a series adaptor, e.g. port 1, can be made reflection-free by setting

$$Z_1 = \sum_{n=2}^{N} Z_n$$

- N-port parallel adaptors with $N \geq 3$.
  Remember that one port of a parallel adaptor, e.g. port 1, can be made reflection-free by setting

$$Z_1 = \frac{1}{\sum_{n=2}^{N} Z_n^{-1}}$$

- N-port arbitrary topological junctions (adaptors) with $N \geq 3$.
  In this general case the adaptation condition for making, e.g., port 1 reflection-free can be found using the symbolic environment in MATLAB. As an example, here follows a MATLAB-like pseudocode for finding the value of Z_1 that makes a generic 5-port topological junction reflection free at port 1

```
syms Z1 Z2 Z3 Z4 Z5 real

Z=diag([Z1,Z2,Z3,Z4,Z5]);

...
```

```
S = eye(5) - 2*Z*B'*inv(B*Z*B')*B;
Z1 = solve(S(1,1)==0,Z1)
```

where `S` is the scattering matrix of the topological junction and `B` is the corresponding fundamental-loop matrix that maps the subset of independent port currents to all port currents. Matrix `B` depends on the specific topological junction you are considering, as we have seen in class.

In summary you are asked to:

1. Draw the proposed WDF scheme including T-shaped stubs at all ports of WD elements and WD junctions that are adapted (reflection-free ports). Please deliver a figure made with a drawing software of your choice or a photo of an orderly and legible drawing you made on a sheet of paper (please remember that, in case we won't understand something that you have drawn, we will be forced to count it as an error).

2. Assign numbers or names to all ports of topological junctions (adaptors).

3. Write down in symbolic form how to set each free parameter of the WDF.
   As an example, if an adapted resistor with resistance R_1 is connected to a port whose free parameter is called Z_4, you will write $Z_4 = R_1$.
   As another example if a 3-port series adaptor is characterized by free parameters Z_1, Z_2, Z_3 and port 1 is made reflection-free, you will write $Z_1 = Z_2 + Z_3$.

5 WDF Implementation

You are asked to implement the WDF you designed according to the instructions presented in the previous section. You will do this by completing the MATLAB script `HW4CrossoverNetWDF.m` provided by us.

The WD models that you can use for implementing the one-port elements of the reference circuit (apart from the already discussed ideal voltage source) are resumed in Table 1 for your convenience.

Table 1: Wave mappings of common WD linear one-port elements.

Constitutive Eq.	**Wave Mapping**	**Adaptation Condition**
$v(t) = V_{\mathrm{g}}(t) + R_{\mathrm{g}}i(t)$	$b[k] = V_{\mathrm{g}}[k]$	$Z[k] = R_{\mathrm{g}}$
$v(t) = Ri(t)$	$b[k] = 0$	$Z[k] = R$
$i(t) = C\frac{dv(t)}{dt}$	$b[k] = a[k-1]$	$Z[k] = \frac{T_{\mathrm{s}}}{2C}$
$v(t) = L\frac{di(t)}{dt}$	$b[k] = -a[k-1]$	$Z[k] = \frac{2L}{T_{\mathrm{s}}}$

As a check of the correctness of your implementation you are invited to compare the output signals of the WDF structure (i.e., the voltage across R_{H}, the voltage across R_{M} and the voltage across R_{L}) with the ground-truth audio signals `outlowsweep.wav`, `outmidsweep.wav` and `outhighsweep.wav`, described in Subsection 1.3. You are also required to plot the three error signals defined as the difference between the three ground-truth audio signals and the output signals of the WDF.

In the script `HW4CrossoverNetWDF.m`, the input signal `ExpSweep.wav` has already been imported and saved in the vector variable `Vin`, the ground-truth output signals have already been imported, the parameters of the circuit have already been set and the figures plotting the output signals and the error signals are already set up. The only missing parts concern the actual implementation of the WDF.

Once the WDF implementation is completed, you are asked to run the script `HW4CrossoverNetWDF.m` varying the sampling frequency and save the obtained plots representing the error signals. Please, run the script testing the following three sampling frequencies:

- $F_{\mathrm{s}} = F_{\mathrm{LTspice}}/4$,
- $F_{\mathrm{s}} = F_{\mathrm{LTspice}}/3$,

- $F_{\mathrm{s}} = F_{\mathrm{LTspice}}/2$,

where $F_{\mathrm{LTspice}} = 192$ kHz is the sampling frequency at which the ground-truth audio files have been synthesized by LTspice.

Questions

Please provide a concise and to-the-point answer to the following questions:

1. Please carefully observe the error between the ground truth signals and the output signals of the WDF referred to the three subcircuits. Do you notice any difference between the error referred to the output of the low-pass filter, the error referred to the output of the band-pass filter and the error referred the output of the high-pass filter in terms of amplitude peaks? If so, please describe. Knowing that the larger the magnitude error, the less accurate the WDF output signal, which of the three WDF output signals is the least accurate one? Please explain why.

2. How does increasing the sampling frequency $F_s = 1/T_s$ affect the error? Please explain why.

3. Consider a slight deviation from the reference circuit of Fig. 1, where a single diode is added in parallel with the tweeter resistor R_{H}. How does this change affect the computability of the WD structure? Also, if we replace the ideal voltage source with a resistive voltage source, how does this affect the WDF implementation (with both the diode and the resistive voltage source)?

4. According to Table (1), the WDF that you implemented uses the trapezoidal rule for approximating the time derivatives present in the constitutive equations of capacitors and inductors in the discrete-time domain. Such an approximation can be described in the frequency domain by considering the following mapping between the Laplace frequency $s = j\omega$ and a bilinear form involving the complex variable z of the Z-transform domain

$$j\omega \leftarrow \frac{2}{T_{\mathrm{s}}} \frac{1 - z^{-1}}{1 + z^{-1}} \,. \tag{7}$$

   Let us now assume that you want to use a different approximation of time derivatives based on the *Backward Euler Method*. In this case, the approximation is characterized by the following mapping

$$j\omega \leftarrow \frac{1 - z^{-1}}{T_{\mathrm{s}}} \,. \tag{8}$$

   Focusing just on inductors, how would you modify their WD models such that they are based on the Backward Euler Method instead of the trapezoidal rule? Describe in the report all the mathematical steps needed to derive such models. In particular:

   - start by considering the constitutive equation of an inductor in the continuous-time domain;
   - express the approximation based on the Backward Euler Method of such constitutive equation in the discrete-time domain;
   - derive the corresponding (*non-adapted*) scattering relation in the WD domain;
   - derive the *adapted* scattering relation along with the corresponding adaptation condition on the free parameter $Z[k]$;
   - finally, fill Table 2.

Table 2: WD Inductor Model based on Backward Euler Method.

Constitutive Eq.	**Wave Mapping in Case of Adaptation**	**Adaptation Condition**
$v(t) = L\frac{di(t)}{dt}$		

Files to be delivered

You are required to deliver the following files:

1. A short **report** in pdf format including
   - the picture of the WDF scheme following the indications discussed in Section 4
   - a description of the WDF scheme in words, in case you believe that your scheme requires some further explanation
   - the setting of free parameters due to adaptation conditions (please define the free parameters at each port coherently to the WDF scheme and specify how to set each free parameter using symbolic expressions as indicated in Section 4)
   - the subplots of the error signals requested in Section 5; three subplots per sampling frequency (one referred to the output of the low-pass filter, one referred to the output of the band-pass filter, one referred to the output of the high-pass filter), for a total of 9 subplots
   - the answers to the questions
2. The folder containing the completed MATLAB script named `HW4CrossoverNetWDF.m` and the four audio files `ExpSweep.wav`, `outlowsweep.wav`, `outmidsweep.wav` and `outhighsweep.wav` such that we can directly execute the script in that folder. Please, **deliver just one MATLAB script and avoid auxiliary files such as MATLAB functions**.

Remember to write your names both in the report and at the beginning of the MATLAB script as a comment. Put both the report and the folder with the MATLAB script in another folder called 'Surname_HW4' (where 'Surname' is your surname) in case you do the homework individually or called 'Surname1_Surname2_HW4' (where your surnames 'Surname1' and 'Surname2' are in alphabetical order) in case you do the homework in groups of 2. Finally, compress the folder in a zip file.

References

[1] A. Farina. "Advancements in Impulse Response Measurements by Sine Sweeps". In: *Proc. 122nd Audio Engineering Society (AES) Convention.* Vienna, Austria, May 2007.