SASP Homework 3

April 28, 2023

The topic of this homework is the implementation of a computationally efficient Leslie speaker emulation presented in [2]. The Leslie speaker is a famous amplifier and speaker for electronic instruments. It is based on a rotating baffle chamber ("drum") in front of the bass speaker and a rotating system of horns in front of the treble driver, which are responsible for the characteristic sound of this device. The musician controls the rotation speed of the drum and horns via an external switch or pedal that alternates between a slow and fast speed setting called *chorale* and *tremolo*. As for how it works, the audio signal is amplified and then sent to a crossover network that splits it into two separate frequency bands (bass and treble), and each output is sent to a speaker. A single woofer is used for bass and a single compression driver is used for the treble. The sound radiated from the speakers is isolated in an enclosure, except for a series of outlets leading to the rotating horns or drum. An electric motor drives both the horns and the drum with the same power, although the horn rotates slightly faster than the drum due to its lighter weight. The rotation of the drum and horn is perceived by the listener as frequency and amplitude modulation.

An efficient digital model for the Leslie loudspeaker is described in [2] and is based on the scheme shown in Fig. 1. The digital structure takes as input a discrete-time audio signal $x[n]$ sampled at a sampling frequency F_s and outputs a discrete-time audio signal $y[n]$. First, the signal passes through two separate filters that emulate the crossover network. The filters are characterized by the same cutoff frequency and divide the signal into two frequency bands, yielding two separate signal paths (bass and treble). The rotation of the drum and horns is modeled separately for each of the two frequency bands and each one comprises two blocks, i.e., a frequency modulation block and an amplitude modulation block. For each signal path, the modulation signals (Modulator 1 and Modulator 2 in fig. 1) are shared between the frequency and amplitude modulation blocks. The frequency modulation effect is implemented with a spectral delay filter (SDF), i.e., a chain of N all-pass filters modulated by an external signal. The outputs of the SDFs are then amplitude modulated with the same modulation signal. The two audio paths are then summed to form the output $y[n]$.

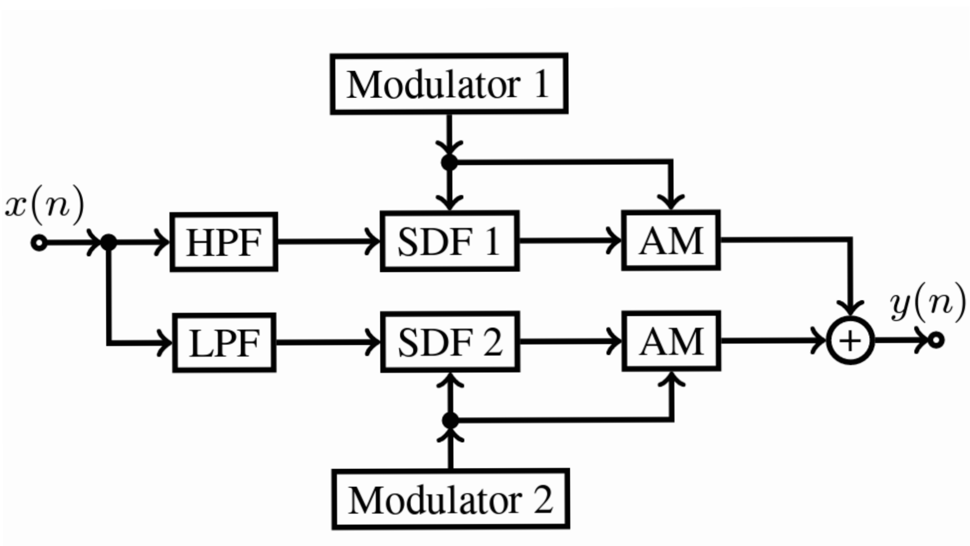

Figure 1: Block diagram of the Leslie speaker emulation.

In this homework you will be asked to:

- implement the digital audio effect scheme of Fig. 1;
- test it using the two possible configurations and compare the results with ground-truth signals provided by us;
- answer a short list of questions.

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

1 Implementation of the Digital Audio Effect Structure

You are asked to implement the digital structure of Fig. 1 using a MATLAB function. In order to facilitate this task, we provide you a commented draft of such a function called `leslie.m` with some missing code. The signature of the function is the following

```
function [y] = leslie(x, Fs, freq)
```

The vector variable `x` is the input audio signal sampled with sampling rate `Fs`, while the vector variable `y` is the returned output audio signal. To implement this effect as in an actual real-time audio plug-in, the processing of the discrete-time input signal should be done sample by sample. This means that all processing blocks in Fig. 1 are implemented in a single processing loop, with each cycle producing a single output sample.

Following the approach of [2], the crossover network is modeled through a couple of 4th-order Butterworth filters, one is a high-pass filter and the other is a low-pass filter. Both filters share the same cutoff frequency $f_c = 800$ Hz. For this homework, it is not necessary to calculate the filter coefficients explicitly, since it is allowed to use native MATLAB functions. Given the filter coefficients, the filter can be implemented in the discrete-time domain using its difference equation and the preferred implementation (Direct Form I, Direct Form II, ...).

According to the scheme in Fig. 1, the rotating drum and horns are modeled using SDF and AM blocks, and are modulated by a signal of type

$$m(n) = M_s m(n) + M_b, \tag{1}$$

where M_s is a scaling factor, M_b is a constant, and $m(n)$ is a sinusoid whose frequency depends on the speed setting (chorale and tremolo). The bass and treble modulators differ in terms of scale factor, bias term, and frequency. In the implementation considered, the bass modulator oscillates at 2 Hz in chorale mode and at 6 Hz in tremolo mode. A small difference in frequency is used to model the slight difference in rotational speed between the horns and the drum. In particular, the treble modulator oscillates 0.1 Hz faster than the bass modulator at both speeds.

As for the SDFs, their global transfer function is

$$H(z) = \left(\frac{z^{-1} + a_1}{1 + a_1 z^{-1}}\right)^N \tag{2}$$

where a_1 is the filter coefficient subjected to modulation by an external signal, and N is the number of cascaded all-pass filters. To facilitate the implementation of each SDF, we propose to use the following difference equation to implement an Nth order SDF [1]

$$y(n) = \sum_{i=0}^{N} \binom{N}{i} m^i(n)[x(n-(N-i)) - y(n-i)]. \tag{3}$$

where $y(n-i) = 0$ when $i = 0$. Also, the parameter characterizing the SDFs and their modulators are reported in the table below.

The amplitude modulation block is charachterized by the following input-output relation

$$y(n) = [1 + \alpha m(n)]x(n), \tag{4}$$

where α is set to 0.9 for both the AM blocks.

Parameter	Treble	Bass
SDF length N	4	3
Modulator Scaler M_s	0.2	0.04
Modulator Bias M_b	-0.75	-0.92

2 Results Validation

Once you have completed the `leslie.m` function according to the suggestions provided in the previous section, you are asked to test it by applying it to the provided input audio signal `HammondRef.wav` at both rotation speeds (i.e., chorale and tremolo).

We provide you with the script `mainHW4.m` in which rotation speeds are already set. You just need to set the string variable `mod_speed` equal to one of the following strings: 'chorale', 'tremolo',

It is then computed the Mean Squared Error (MSE) between the output signals returned by the `leslie.m` function you implemented and the ground-truth audio signals (one per rotation speed) contained in the `Leslie_ref` folder. The MSE is finally printed to screen.

You are asked to compute the MSE for each rotation speed and annotate their values. Please note that the MSE is a measure of accuracy of your result (the lower the MSE, the more your results are accurate); if your implementation of the considered effect is correct you should have that MSE $< 5 \times 10^{-10}$.

Questions

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

1. In this homework, you implemented two SDFs to perform frequency modulation. The subtle frequency modulation achieved by this technique is similar to a vibrato effect. Please name another approach to implementing this effect and describe its computational scheme.
2. When you implemented the two SDFs, the modulator signals were predetermined. Is it possible to use any modulator signal for the chain of all-pass filters without affecting the stability of the filter? If the stability of the filter is an issue, please specify a valid condition for the modulator signal to ensure stability.
3. Let us consider the first all-pass filter in the chain of an SDF. What can we say about its group delay? For a pediodic signal with two spectral components $u(n) = A_1 \sin(\omega_1 n/F_s) + A_2 \sin(\omega_2 n/F_s)$, with $\omega_1 \neq \omega_2$, as input signal to the all-pass filter stage, is the delay of both frequency components equal? Under what condition?
4. Suppose we implement a fractional delay using the Lagrange method. For a delay $D = 3.3$ samples, how does the filter order N affect the magnitude response and phase delay? Can we keep increasing the filter order to get better magnitude and phase delay responses? And why? Justify your answer providing useful MATLAB plots of the magnitude and phase delay responses.

Files to be delivered

You are required to deliver the following files:

1. A short **report** in pdf format including
   - The two mean squared errors (for both "chorale" and "tremolo" modes).
   - The answers to the questions.
2. The **whole folder** containing the MATLAB files provided by us with the completed `leslie.m` function.

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 files 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] J. Kleimola et al. "Sound Synthesis Using an Allpass Filter Chain with Audio-Rate Coefficient Modulation". In: *Proc. of the 12th Int. Conf. on Digital Audio Effects (DAFx-09)* (Sept. 2009).

[2] J. Pekonen, T. Philajamaki, and V. Valimaki. "Computationally Efficient Hammond Organ Synthesis". In: *Proc. of the 14th Int. Conf. on Digital Audio Effects (DAFx-11)* (Sept. 2011).