Sound Analysis, Synthesis and Processing

Third Homework

Report
Digital Audio Effects

Students

Alberto Doimo - 10865196
Paolo Ostan - 10868276

POLITECNICO
MILANO 1863

Introduction

a) Leslie Simulation

The request for this homework was to implement a digital audio effect and in particular a Leslie speaker emulation. The digital model is based on the scheme proposed in figure 1.

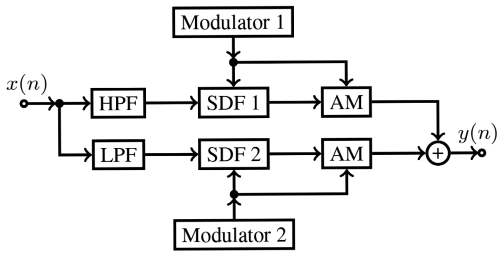

Figure 1: Block diagram of the Leslie speaker emulation.

The implementation was developed in MATLAB, as asked, and follows the requested steps of the precompiled script. In particular the `Leslie.m` file presents the following processing steps:

1. Design of the crossover network using two 4th order Butterworth filters, a *low-pass frequency* and a *high-pass frequency* filter with cutoff frequency *fc=800 Hz.*
2. Computation of the filter outputs for the N samples exploiting the predefined *in* and *state* buffers, arranged as circular arrays in order to optimize the number of writings and readings and simulate a deployable real-time system.
3. Spectral delay filters implementation to emulate the work of bass and treble units, implemented exploiting circular array as for the crossover.
4. Amplitude modulation block implementation.

In the `mainHW3.m`:

1. Modulation speed selection.
2. Audio file load.
3. Leslie speaker simulation by `Leslie` function.
4. Playback of the output signal.
5. Mean Squared Error (MSE) computation between the output signal and the reference one (`Leslie_ref`).

The MSE results are here displayed both for *tremolo* and *chorale* velocities:

- *Chorale* MSE: `3.113e-10`
- *Tremolo* MSE: `3.10989e-10`

Both results are, as asked, lower than 5×10^{-10}

Q1) Vibrato effect techniques

After the implementation of the Leslie speaker which produces a vibrato-like effect, whose results are shown in point *(a)*, we discuss now other methods to implement a vibrato effect. As presented in [1], a standard and generic implementation for modulation effects such as *chorus, flanger, vibrato, doubler, echo* is presented in figure 2. To achieve the desired effect, the designer can tune the different parameters of the system such as *Blend, Feedforward and Feedback.* In the specific, in order to obtain a vibrato effect the parameters assume the values listed in table 1.
Rearranging the system according to the parameters

Modulation system Parameters		
Blend	**Feedforward**	**Feedback**
0.0	1.0	0.0

Table 1: Modulation system parameters

assigned we obtain the simpler scheme represented in figure 3, As matter of fact, as discussed in [3] the vibrato effect can be interpeted as a varying time delay. In order to implement it we need a delay line and a low-frequency oscillator to drive the delay time parameter.

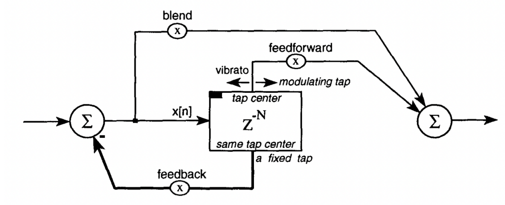

Figure 2: Modulations block scheme

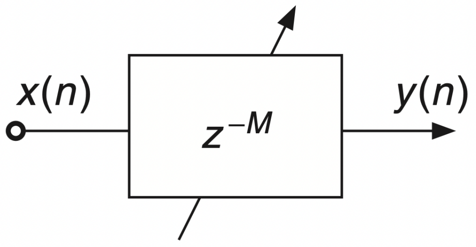

Figure 3: Vibrato block diagram

As explained in both [3] and there are many ways to implement variable delay-line such as:

- Least squares design
- windowing methods
- GLS method
- Lagrange interpolation

- Allpass Interpolation

These different implementation techniques are characterized by some pros and cons which make them suitable in various cases.
The methods enumbered have the aim of minimizing the error calculated between real solution performed through different types of filters and the ideal solution, whose impulse response is represented by a single impulse for $n = D$, when D is an integer.

$$h_{id} = \delta(n - D) \tag{1}$$

In the case in which D is not an integer $h_{id}(n)$ is an represented by a *sinc* function centered in D infinitely long and non-causal.

$$h_{id} = sinc(n - D) \tag{2}$$

The first four methods offer different approximations of the sinc function using *FIR filters* coefficients, which allow to obtain a filter which is guaranteed to be stable, and allows more linear responses but needs the coefficients of the filter to be recomputed every time we need to vary delay time. The fifth method provides an implementation of the fractional through the use of an *IIR allpass filter*, which allows a more fast computation of the real-time varying parameter but is less linear in the magnitude and phase response, and needs proper parameters tuning in order to be stable.

Q2) Spectral Delay Filter Stability

The Spectral Delay Filter is generally implemented as a cascade of N first order allpass filters with transfer function:

$$H(z) = \frac{a_1 + z^{-1}}{1 + a_1 z^{-1}} \quad a_1 \in [-1, 1] \tag{3}$$

So the cascade will be:

$$H_N(z) = \left(\frac{a_1 + z^{-1}}{1 + a_1 z^{-1}}\right)^N \tag{4}$$

The coefficent a_1 is a function of time and varies with the modulator $m'(n)$ signal as:

$$a_1(n) = m'(n) \tag{5}$$

where the modulator can assume any shape.
In the Leslie emulation the modulator $m'(n)$ is driven by a sinusoidal oscillator $m(n)$, modified with the parametrs M_s and M_b as:

$$m'(n) = M_s m(n) + M_b \tag{6}$$

These parameters have the following values:
In general, in order to ensure the filter stability the following condition must be respected for a_1:

$$|a_1| < 1 \tag{7}$$

Modulator Parameters

Parameters	**Treble**	**Bass**
Modulator Scaler Ms	0.2	0.04
Modulator Bias Mb	-0.75	-0.92

Table 2: Modulation Parameters

So for the modulator it must be:

$$|a_1(n)| = |m'(n)| < 1 \tag{8}$$

This condition directly derives from the stability of an LTI causal digital filter which must have (in the z-plane):

1. all the poles inside the unit circle
2. the unit circle inside the ROC

In the Figure 4 there is a representation of a cascade of 4 first-order all pass filters, with $a_1 = 0.95$, which is the biggest value that results from the modulator (6) with the values of table 2. Table 3 shows te maximum and minimum values obtained with 6 for both *bass* and *treble* velocities.

Values of the modulator

Velocities	**max value**	**min value**
Bass	-0.88	-0.96
Treble	-0.55	-0.95

Table 3: Maximum and minimum values of the modulator

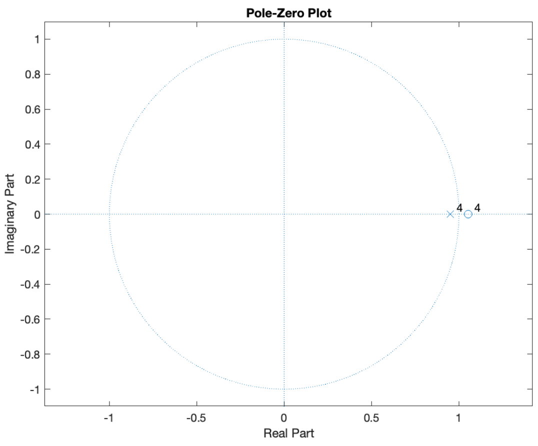

Figure 4: Pole-zero plot

Q3) Group-delay conditions

The group delay for a first order all pass filter is defined as:

$$\tau_g(\omega) = -\frac{\partial \angle H(\omega)}{\partial \omega} = \frac{1 - a_1^2}{1 + 2a_1 cos(\omega) + a_1^2} \tag{9}$$

Evaluating the formulation of the *group delay* we can notice how, for positive values of a_1 the filter results to be more linear around low frequencies, and viceversa, for negative values of a_1 the filter results to be more linear around high frequencies. Another evaluation made considering the filter formulation is the importance of the choice of the correct order for the filter in order to obtain a *Group delay* quasi-linear along the whole spectrum. In the specific case we noticed how, for values of a_1 that tend to zero the filter frequency response obtained becomes an ideal delay $\delta(n-D)$, with delay value D equal to the order of the filter. In the specific considering a *first-order allpass* filter we obtain:

$$H_{AP}(z) = \frac{a_1 + z^{-1}}{1 + a_1 z^{-1}} => z^{-1}\; for\; a_1 => 0 \tag{10}$$

Therefore, as validated by the first graph presented in figure 5, we can notice how, it is convenient to choose the order of the filter as

$$N = round(D) \tag{11}$$

And then tune the value of a_1 according to the desired delay value to obtain. One possible implementation of a *first-order* allpass filter is the *Thiran Allpass filters*. Where the phase is assumed to be:

$$arg\{H(\omega)\} \approx -\frac{sin(\omega_0)}{a_1 + cos(\omega_0)} + \frac{a_1 sin(\omega_0)}{1 + a_1 cos(\omega_0)} \tag{12}$$

$$= -\omega_0 \frac{1 - a_1}{1 + a_1} \tag{13}$$

$$a_1 = \frac{1 - D}{1 + D} \tag{14}$$

The *Group Delay and Phase Delay* are now computed for a range of different values which can be assume by a_1 mantaining the filter stable and shown in figure 5.

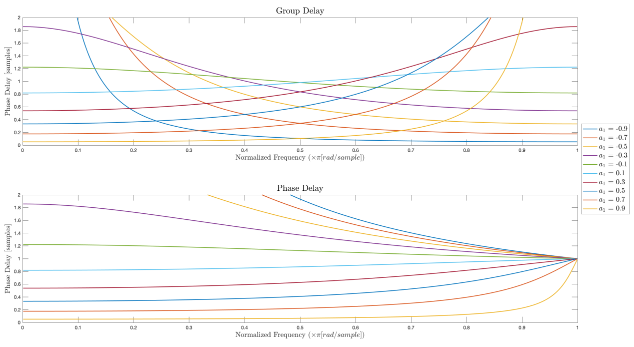

Figure 5: First order allpass filter Group and Phase delay

Considering now a signal:

$$u(n) = A_1 sin(\frac{\omega_1 n}{Fs}) + A_2 sin(\frac{\omega_2 n}{Fs}) \tag{15}$$

We evaluate the signal convolved with a first-order allpass filters, as discussed in previous point, and consider the delay that the two frequency components assume due to the filter operation. In order to evaluate it we first consider the *Phase delay*, which is defined as

$$\tau_p(\omega) = -\frac{\angle H(\omega)}{\omega} \tag{16}$$

allows us to evaluate delay in samples assumed by each frequency component of the signal after passing through the filter. Considering the second plot in figure 5 we can notice how for low values of a_1 the *phase delay* of the system is quasi-linear, while increasing the module of a_1 the delay difference between the signals increases. The *Group delay* instead allows to evaluate the comprehensive delay assumed by the composite signal when processed by the filter.

Q4) Lagrange Interpolation

Considering fractional delay, the implementation developed using Lagrange interpolation aims at making the of the *FIR* filter used to computer the delay, and the sinc function, ideal fractional delay, used as reference maximally flat for a certain order N of derivatives, specified as parameter.

$$\left.\frac{d^l E(\omega)}{d\omega^l}\right|_{\omega=\omega_0} = 0,\; for\; l = 0, ..., N \tag{17}$$

We therefore obtain a problem in $N+1$ linear equations:

$$\sum_{n=0}^{N} n^l h(n) = D^l\; with\; l = 0, ..., N \tag{18}$$

Which can be solved in matrix form as:

$$Vh = v \tag{19}$$

And then solved using the inverse formulation:

$$h = V^{-1} v \tag{20}$$

Leading to the solving form which allows us to find the impulse response of the desired filter as:

$$h(n) = \prod_{k=0, k\neq n}^{N} \frac{D-k}{n-k}\quad n = 0, ..., N \tag{21}$$

The case of the implementation of a fractional delay with value $D = 3.3$ samples is considered. In the specific we discuss the filter response varying the order N of the filter.
As a matter of fact in figure 6 we present a comparison between different orders of polinomials for the lagrange interpolation. The values presented in the tables are in a range of N that goes from $N = 4$ to $N = 9$. Where $N = 4$ is the lower-bound for the polinomial order as we to respect the condition:

$$N > \lfloor D \rfloor \tag{22}$$

It is noticeable how increasing the order of the polinomial leads to an improvement of the overall response up to a certain value of N, respecting the optimality condition stated by [2]:

$$\frac{N-1}{2} \leq D \geq \frac{N+1}{2} \tag{23}$$

That in the specific case , considering $D = 3.3$ samples, leads to

$$5 \leq N \geq 7 \tag{24}$$

Hence considering $N > 7$ we notice how the response in term of *magnitude* or *Linear or Group Delay* present a worse response than those obtained considering the optimal values for the polinomial order.

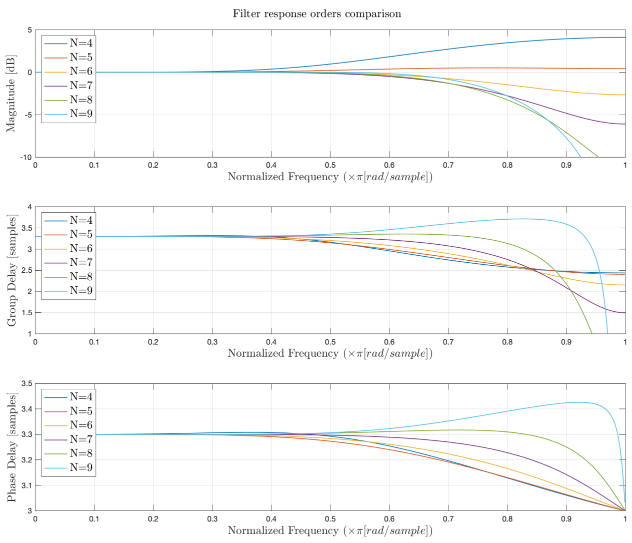

Figure 6: Filter response comparison for different order N

In figure 7 the filters obtained with *Lagrange intepolation* are compared to the ideal sinc centered in $D = 3.3$ samples in order to have a qualitative idea of the different responses at the variation of the filter order.

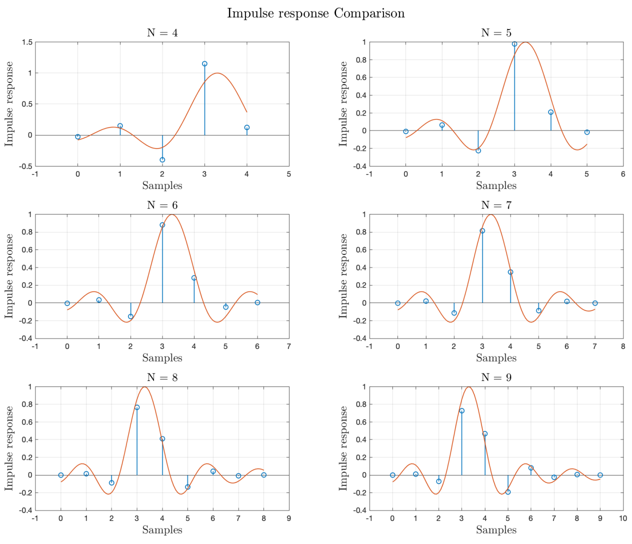

Figure 7: Impulse response of different order filters compared to ideal Sinc

References

[1] J. DATTORRO, *Effect design part 2: Delay-line modulation and chorus*, Journal of Audio Engineering Society, 45 (1997).

[2] A. K. K.RAJALAKSHMI, SWATHI GONDI, *A fractional delay fir filter based on lagrange interpolation of farrow structure*, International Journal of Electronics and Electical Engineering, 2 (2013).

[3] U. ZÖLZER, ed., *DAFX: Digital Audio Effects*, Wiley, second ed., 2011.