Synth Challenge 2019 - Report

Adam Kubíček

1. Synth Challenge

Synth Challenge is the sound and musical instrument synthesis competetion using software envirometn Matlab. It is based at Czech Technical University in Prague, Faculty of Electrical engineering. This work is divided into 3 parts:

- mandatory composition,
- musical scale,
- arbitrary composition.

2. Mandatory composition

Main task in this part of the work was to syntetise musical instruments for a purpose of playing a *.mid file which was imported into the Matlab. I have chosen Popelka.mid wich contains 5 instruments.

- 01 – Piano
- 46 – Pizzicato Strings
- 49 – String Ensemble
- 71 – Bassoon
- 74 – Flute

I have basically used 2 methods of musical instrument synthesis. Additive synthesis is mainly used because it fitted well in case of piano, string ensemble, flute and bassoon synthesis. And because it is pretty intuitive.

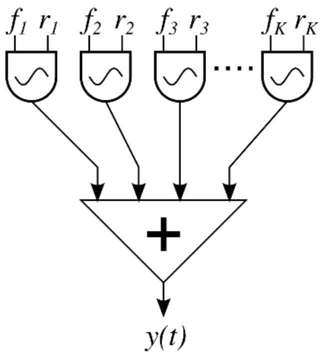
Additive synthesis is a sound synthesis technique that creates timbre by adding sine waves together.

The timbre of musical instruments can be considered in the light of Fourier theory to consist of multiple harmonic or inharmonic partials or overtones. Each partial is a sine wave of different frequency and amplitude that swells and decays over time due to modulation from an ADSR envelope or low frequency oscillator. In this work was used the ADSR envelope.

But for the purpose of bassoon synthesis it was used the Karplus-Strong algorythm because I wanted to use different method of synthesis and this case was an oportunity to implement such method.

2.1 Piano, String Ensemble, Basoon, Flute

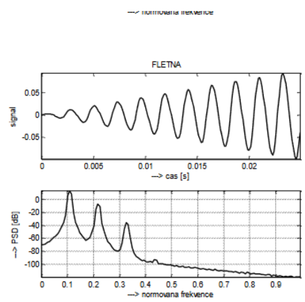
These musical instruments were synthetised using Additive Synthesis. Basic task of this method is to find specific harmonics of the instrument and to create accurate ADSR envelope.



IMG 1 - Schematic diagram of additive synthesis. The inputs to the oscillators are frequencies f_k and amplitudes r_k .

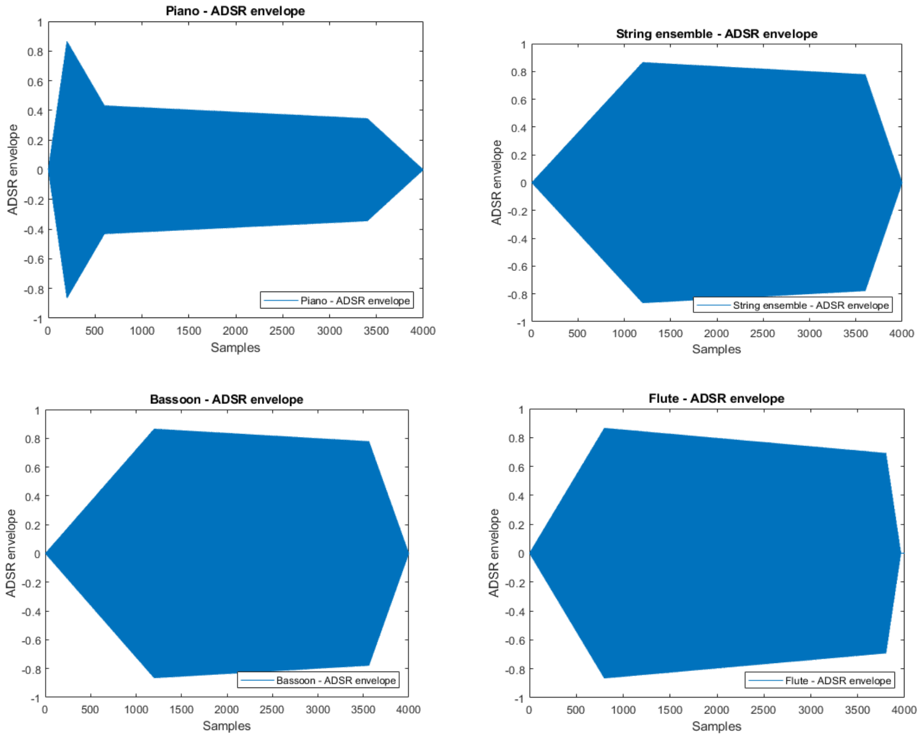
The harmonics of specific instument were found in lectures and on the internet (forums mainly).

Poř.harmonické	**1.**	**2.**	**3.**	**4.**	**5.**	**6.**	**7.**	**8.**	**9.**	**10.**	**11.**
Trubka	0,17	0,63	0,57	0,98	0,56	0,68	0,02	0,05	-	-	-
Harmonika	8,60	0,45	3,40	0,50	0,42	0,13	0,13	0,16	0,04	0,35	0,02
Flétna	2,54	0,25	0,01	-	-	-	-	-	-	-	-
Klarinet	1,00	0,00	0,75	0,00	0,50	0,00	0,14	0,50	0,00	0,12	0,17
Hoboj	0,02	0,20	1,00	0,37	0,36	0,46	0,10	0,06	0,03	0,02	-
Piano	0,32	0,20	0,08	0,07	0,06	-	-	-	-	-	-
Housle	0,39	0,30	0,17	0,01	0,11	-	-	-	-	-	-
Hlas	0,43	0,08	0,01	-	-	-	-	-	-	-	-



IMG 2 – This picture shows examples of harmonics of individual instruments

ADSR envelopes (Attack-Decay-Sustain-Release) are shown bellow (in order left-right: piano, string ensemble, bassoon, flute).



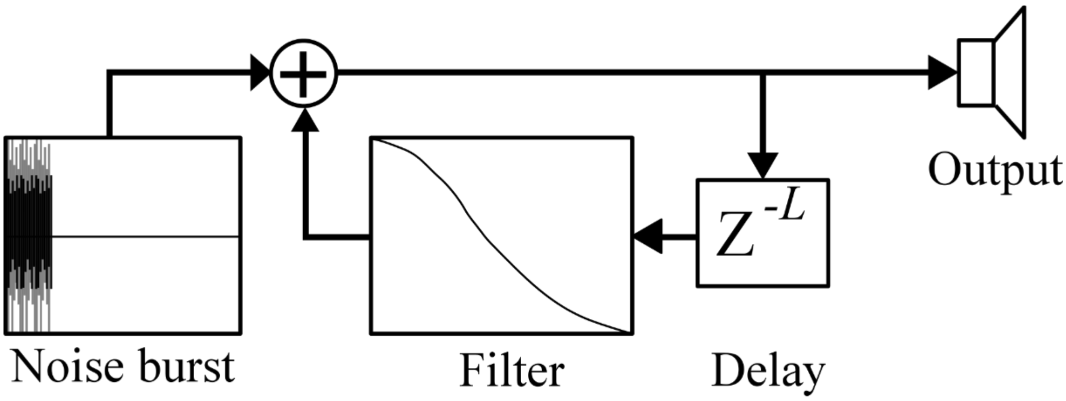
Each instrument has to have typical sound color. Due to that the modulated wave wasn't always the sine wave.

In case of Bassoon it was used the square wave, because it has richer low frequencies and it is typically used for bass-based instruments.

In case of String ensemble it was used the sawtooth wave because it has rougher and more distorted sound than just a sine wave. To create some kind of vibrato effect to the string sound was used Tremolo effect. It basically is sinusoidal AM modulation. Parameters of Tremolo effect are: Fc – vibrato frequency [Hz], and alpha – amplitude of the sine wave (power of the modulation). In this work I used Fc = 5, alpha=0.2.

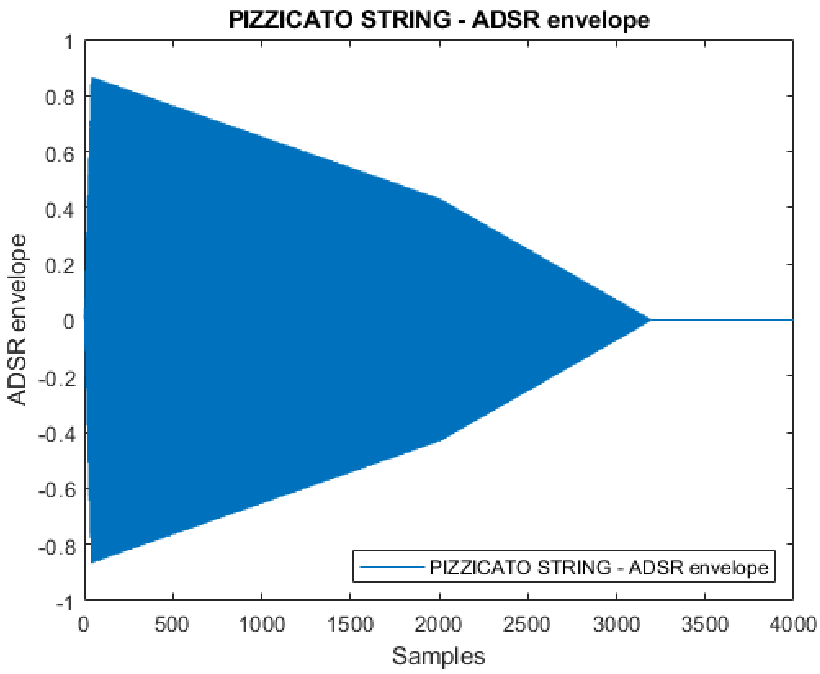
2.2 Pizzicato Strings

Pizzicato strings were synthetise with Karplus-Strong algorythm. Karplus–Strong string synthesis is a method of physical modelling synthesis that loops a short waveform through a filtered delay line to simulate the sound of a hammered or plucked string or some types of percussion.



1. A short excitation waveform (of length L samples) is generated. In the algorithm. I used a burst of white noise.
2. This excitation is output and simultaneously fed back into a delay line L samples long. (In this work: L = round(Fs/freq-0.5, where Fs = sample frequency, freq is frequency of the note)
3. The output of the delay line is fed through a filter. The gain of the filter must be less than 1 at all frequencies, to maintain a stable positive feedback loop.
4. The filtered output is simultaneously mixed back into the output and fed back into the delay line.

This sound was modulated by the ADSR envelope with fast attack for plucked strings shown bellow.



3. Musical scale

In this task C major scale of three octaves was created. Firstly I have created the vector of frequencies that I needed for three octaves of C major scale. Then I played this scale with instruments created in chapter 2. The outpu is in folder *results* called stupnice.m4a.

4. Arbitrary composition

I have chosen to create an isntrument that will have some reverberation. For that case I created banjo using additive synthesis with exponential decrease of amplitude.

Convolutionary Reverb is easily created if we know the impulse response of the room we want to simulate. For that case I downloaded free impulse responses from web http://www.voxengo.com/impulses/. Impulse response *Nice Drum Room.wav* was chosen to create our simulated room with reverberation.

The output of convolution between input singnal (x(t)) and impulse response (h(t)) is our reverbered signal.

$$y(t) = \ x(t) * h(t)$$

But the impulse response is too short to create such filtrations. The solution is to implement FFT.

$$Y(k) = X(k).H(k)$$

Due to that we will have multiplication between fourier image of input signal (X(k)) and frequency response of the room (H(k)).

After that we have to transform from frequency domain to the time domain. To do that we will use inverse fourier transform.

$$real(iFFT\big(Y(k)\big)) = y(t)$$

For the purpose of demosntration I downloaded free midi file of country song that is played with this reverbed banjo. The output is located in folder results under the name of *country.m4a*.

5. References

1. *Julius O. Smith III. "Additive Synthesis (Early Sinusoidal Modeling)". Retrieved 14 January 2012.*
2. *Schroeder, M, R..: „Colorless“ Artifical Reverberation*
3. *Karplus, Kevin; Strong, Alex (1983). "Digital Synthesis of Plucked String and Drum Timbres". Computer Music Journal. MIT Press. 7 (2): 43–55.*
4. *Lectures of http://sami.fel.cvut.cz/syn/*