

SYNTH CHALLENGE 2016

Zadání:

- a) Syntéza libovolného sólového hudebního nástroje pro některou z následujících skladeb:
 - J.S.Bach "Dobře temperovaný klavír - preludium a fuga BWV 846"
 - A.Dvořák "Humoreska"
- b) Dvě oktávy durové hudební stupnice pro vybraný sólový hudební nástroj z předchozího bodu
- c) Libovolná vlastní realizace audio syntézy v MATLABu (možnost i nehudbních zvuků)

Cíl:

- a) Syntéza zvuku čisté elektrické kytary pro skladbu „Humoreska“
- b) Syntéza dvou oktáv stupnice C dur pomocí nástroje z předchozího bodu
- c) Syntéza zvuku zkreslené elektrické kytary pro skladbu „Black Dog“

Postup:

Syntézu elektrické kytary jsem prováděl pomocí dvou technik. Aditivní syntézou deseti harmonických jsem vytvořil čistý tón. Následně jsem použil Karplus-Strongův algoritmus pro vytvoření kovového zvuku strun. Výstup těchto dvou syntéz jsem následně sečetl a vynásobil obálkou. Prvním krokem bylo zjištění vlastností tónu, který vytváří elektrická kytara. Nahrál jsem jeden tón (A 220Hz) přes zvukovou kartu přímo do programu Cubase. Tento zvuk jsem následně načetl do Matlabu a provedl frekvenční analýzu pomocí příkazu `fftshift(fft(in))`. Touto analýzou jsem si ověřil, že se zvuk skládá z několika harmonických, které jsou lineární kombinací 220Hz. Pomocí funkce `ginput` jsem ze zobrazené frekvenční charakteristiky načetl amplitudy jednotlivých harmonických, které jsem napevno napsal do proměnné `amplitude`. Následně jsem vytvořil vektor deseti frekvencí a pomocí `for` cyklu a funkcí `sin` jsem tyto harmonické sečetl dohromady. V dalším kroku jsem vytvořil obálku tónu. Tato obálka se skládá ze 4 částí. Lineární attack, konstantní sustain, exponenciální decay a logaritmický release. Tato obálka slouží k útlumu signálu jako u reálné kytary a také k zamezení praskání při prudkých změnách amplitudy signálu. V předposledním kroku jsem využil Karplus-Strongova algoritmu z přednášky k vytvoření kovového zvuku strun. Tento zvuk jsem ale výrazně ztlumil, aby výsledný tón připomínal spíše elektrickou než akustickou kytaru. Nakonec jsem oba algoritmy sečetl.

K tvorbě dvou oktáv durové stupnice pomocí tohoto zvuku jsem vytvořil proměnnou obsahující 14 frekvencí tónů této stupnice. Celý algoritmus pro syntézu tónu kytary jsem vložil do `for` cyklu. V průběhu cyklu se nejprve načte frekvence tónu, následně se tón vygeneruje, uloží se na příslušnou pozici do výstupní proměnné a nakonec se výstupní proměnná zapíše do souboru ve formátu wave.

Syntézu zkreslené elektrické kytary jsem prováděl pomocí aditivní syntézy podobně jako v prvních úkolech. Tón vytvořený touto syntézou jsem následně vynásobil stejnou obálkou. V tomto případě jsem ale nevyužil Karplus-Strongova algoritmu. Zkreslení kytary jsem provedl pomocí jednoduchého kódu nalezeného na stránkách mathworks.com.

Zdroje:

- <http://sami.fel.cvut.cz/sms/SMS12.pdf>
- <https://www.mathworks.com/matlabcentral/fileexchange/6639-guitar-distortion-effect/content/gdist.m>

```
% Syntéza zvuku čisté elektrické kytary
% Souctova syntéza - vytvoření zvuku kytary
clear all, close all, clc
freq = 220; %testovací parametr - frekvence tónu
dur = 1; %testovací parametr - trvání tónu
amp = 1; %testovací parametr - amplituda tónu
Fs = 44100; %vzorkovací frekvence
amplitudy = amp*[1 0.1732 0.0409 0.05565 0.0273 0.0118 0.0164 0.0058 0.0047
0.0018]; %amplitudy harmonických
frekvence = freq*[1:10]; %frekvence harmonických
t = 0:1/Fs:dur-1/Fs; %časová osa
y = amplitudy(1)*(sin(2*pi*frekvence(1)*t)+cos(2*pi*frekvence(1)));
%vytvorení první harmonické
for k = 2:10 % for cyklus pro přidání dalších 9 harmonických
    y = y + amplitudy(k)*(sin(2*pi*frekvence(k)*t)+cos(2*pi*frekvence(k)));
end
% Generování obálky ADSR [a b c d]
a = linspace(0, 1, 500); %attack v podobě lineárního zesilování (také
zabrání "praskání" na začátku tónu)
b = ones(1,500); %sustain
c = exp(-t(end-length(a)-length(b)-1000)/2); %exponenciální decay
d = (c(end)*(ones(1,1000)))./logspace(0, 1, 1000); %logaritmický release
(zabrání "praskání" na konci tónu)
y = y.*[a b c d]; %násobení tónu obálkou
out = y;
% Karplus-Strong (pouze pro přidání "kovového" zvuku strun)
% zdroj: http://sami.fel.cvut.cz/sms/SMS12.pdf , strana 14
g = 0.5;
z = [];
q=zeros(1,floor(dur*Fs));
N=round(Fs/freq);
q(1:N)=randn(1,N);
p=[1 zeros(1,N-1) -g -g]; r=[1];
u=filter(r,p,q);
z=[z u];
% Výstup
y = out + z/100; %sečtení aditivní syntézy a Karplus-Strongova algoritmu
```

```

% Stupnice C dur - C4-B5(261.63 - 987.77Hz)
clear all, close all, clc
for i = 1:14 %for cyklus pro nařtení frekvencí tónů stupnice
    % freq - vektor řtrnácti frekvencí (pro dvř oktávy stupnice)
    freq = [130.81 146.83 164.81 174.61 196.00 220 246.94 261.63 293.66
329.63 349.23 392.00 440.00 493.88]*2;
    dur = 1; %testovací parametr - trvání tónu
    amp = 1; %testovací parametr - amplituda tónu
    Fs = 44100; %vzorkovací frekvence
    % Souctova syntéza - vytvoření zvuku kytary
    amplitudy = amp*[1 0.1732 0.0409 0.05565 0.0273 0.0118 0.0164 0.0058
0.0047 0.0018]; %amplitudy harmonických
    frekvence = freq(i)*[1:10]; %frekvence harmonických
    t = 0:1/Fs:dur-1/Fs; %řasová osa
    y = amplitudy(1)*(sin(2*pi*frekvence(1)*t)+cos(2*pi*frekvence(1)));
%vytvoreřní první harmonické
    for k = 2:10 % for cyklus pro přidání dalších 9 harmonických
        y = y +
amplitudy(k)*(sin(2*pi*frekvence(k)*t)+cos(2*pi*frekvence(k)));
    end
    % Generování obřlky ADSR [a b c d]
    a = t(1:500)*88; %attack v podobř lineřrního zesilování (takř zabrřní
"praskřní" na zařřtku tónu)
    b = ones(1,500); %sustain
    c = exp(-t(1:end-length(a)-length(b)-500)/2); %exponenciřlní decay
    d = (c(end)*(ones(1,500)))./logspace(0, 2, 500); %logaritmicřký release
(takř zabrřní "praskřní" na konci tónu)
    y = y.*[a b c d];%řnřsobení tónu obřlkou
    % Karplus-Strong (pouze pro přidřní "kovovřho" zvuku strun)
    % zdroj: http://sami.fel.cvut.cz/sms/SMS12.pdf , strana 14
    g = 0.5;
    z = [];
    q=zeros(1,floor(dur*Fs));
    N=round(Fs/freq(i));
    q(1:N)=randn(1,N);
    p=[1 zeros(1,N-1) -g -g]; r=[1];
    u=filter(r,p,q);
    z=[z u];
    y = y + z/100;

    out(1+(i-1)*(dur*Fs):(i)*(dur*Fs)) = y; %vytvoreřní vřstupnřho vektoru s
postupnř jdoucřmi tony
end
% Vřstup
y = out;

```

```
% Syntéza zvuku zkreslené elektrické kytary
% Souctová synteza
clear all, close all, clc
freq = 220; %testovací parametr - frekvence tónu
dur = 2; %testovací parametr - trvání tónu
amp = 1; %testovací parametr - amplituda tónu
Fs = 44100; %Vzorkovací frekvence
amplitudy = amp*[1 0.1732 0.0409 0.05565 0.0273 0.0118 0.0164 0.0058 0.0047
0.0018]; %amplitudy harmonických
frekvence = freq*[1:10]; %frekvence harmonických
t = 0:1/Fs:dur-1/Fs; %časová osa
y = amplitudy(1)*(sin(2*pi*frekvence(1)*t)); %vytvoření první harmonické
for k = 2:10 % for cyklus pro přidání dalších 9 harmonických
    y = y + amplitudy(k)*(sin(2*pi*frekvence(k)*t));
end
% Generování obálky ADSR [a b c d]
a = linspace(0, 1, 500); %attack v podobě lineárního zesilování (také
zabrání "praskání" na začátku tónu)
b = ones(1,500); %sustain
c = exp(-t(1:end-length(a)-length(b)-500)/2); %exponenciální decay
d = (c(end)*(ones(1,500)))./logspace(0, 2, 500); %logaritmický release
(zabrání "praskání" na konci tónu)
y = y.*[a b c d]; %násobení tónu obálkou
%Zkreslení zvuku
%Zdroj: https://www.mathworks.com/matlabcentral/fileexchange/6639-guitar-distortion-effect/content/gdist.m
k = 2*0.9/(1-0.9);
y = (1+k)*(y)./(1+k*abs(y)); %Zkreslený tón
```