

ČVUT FEL

Zpráva k semestrální práci

A2B31SMS

Jan Vimr
2017/2018

1. Postup

Zadáním semestrální práce byla syntéza libovolného hudebního nástroje pro skladbu: „Let čmeláka“ – Nikolaj Rimskij Korsakov, dále odevzdat dvě durové hudební stupnice daného nástroje a libovolná vlastní realizace audio syntézy v prostředí MATLAB. Jelikož zadaná skladba „Let čmeláka“ má velmi rychlé tempo, a tudíž všechny tóny jsou velmi krátké, hledal jsem pro svoji tvorbu nástroj, který byl pro takovou skladbu vhodný a nakonec jsem se rozhodl pro syntézu xylofonu.

A) „Let čmeláka“

První fází mojí práce byla analýza několika vzorků signálu xylofonu a to jak v časové tak spektrální oblasti. Při analýze časového průběhu zvuku xylofonu jsem dospěl k závěru, že nejlepším řešením obálky signálu bude klesající exponenciála. Analýzou spektra signálu jsem získal hodnoty nejvýraznějších spektrálních čar. Dále jsem zjistil, že spektrální složení signálu se v čase mění a proto jsem původně uvažoval o použití tabulkové syntézy. Po několika pokusech jsem ale od této metody ustoupil a pokračoval místo toho v pokusech s aditivní syntézou.

Další fází už bylo samotné napsání programu. Použil jsem tedy MIDI Toolbox a začal upravovat funkci „synth.m“. Pro větší přehlednost jsem pro jednotlivé nástroje vytvořil zvláštní funkce, které jsou zavolány z funkce „synth.m“. Finální podoba funkce „synth.m“:

```
if channel~=10;
    if synthtype == 14
        y=xylophone(freq,dur,amp,Fs);
    elseif synthtype == 74
        y=flute(freq,dur,amp,Fs);
    elseif synthtype<64
        tau = 0.1; % doznivani
        y = amp.*exp(-t/tau).*sin(2*pi*freq*t);
    else
        y = amp.*square(2*pi*freq*t);
    end
end
```

Realizace samotné funkce „xylophone.m“ vypadá takto:

```
function y=xylophone(freq,dur,amp,Fs)
% XYLOFON
% freq ... zakladni harmonicka frekvence
% dur ... delka trvani tonu
% amp ... hlasitost tonu
% Fs ... vzorkovaci frekvence

tau = .06; % parametr exponencialni obalky
dur = .8; % doba trvani tonu
t = 0:1/Fs:dur-1/Fs; % casova osa
```

```

o = exp(-t./tau); % obalka signalu
ampl = [.4 .04 .3 .08 .04 .09 .05]; % amplitudy jednotlivych harmonickych
slozek
x = o.*[ampl*sin(2*pi*[1 3 7 11.8 13.7 18.5 21.7]'*freq*t)]; % aditivni
synteza signalu
y = amp.*x; % hlasitost signalu

```

Jelikož většina celé syntézy probíhá na jednom řádku, zde příkládám ještě podrobnější rozepsání této části programu do jednotlivých úkonů:

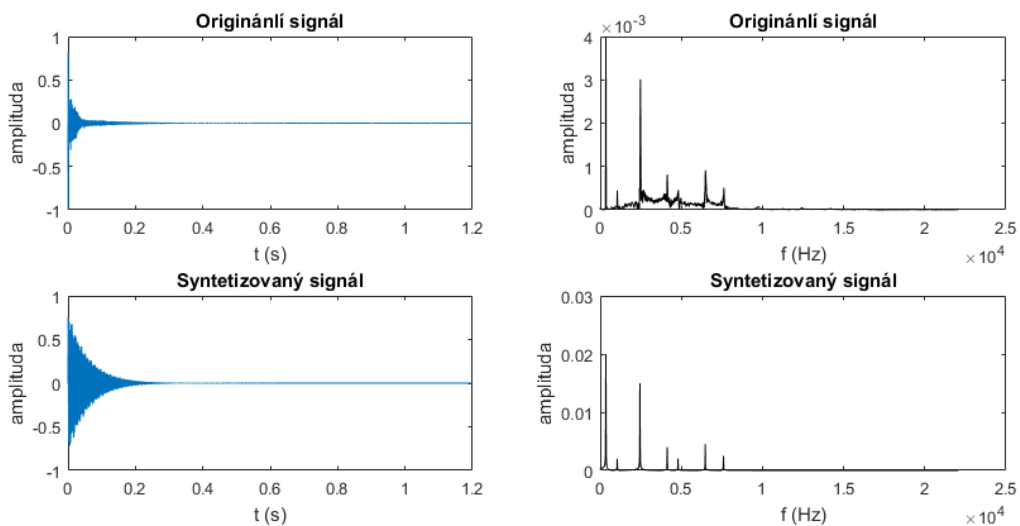
```

% aditivni synteza signalu
ampl = [.4 .04 .3 .08 .04 .09 .05]; % amplitudy jednotlivych harmonickych
slozek
n = [1 3 7 11.8 13.7 18.5 21.7]'; % sloupcovy vektor obsahujici nasobky
harmonickych slozek
s = sin(2*pi*n*freq*t); % matice, kde pocet radku = length(n), pocet
sloupcu = length(t)
a = ampl*s; % nasobeni vektor(1,7)* matice(7,length(t)) -> radky se
poscitaji s vahou danou vektorem ampl
x = o.*a; % vynasobeni obalkou

```

Ve funkci jsem nakonec nahradil parametr dur, tedy délku trvání tónu, konstantou, jelikož mi přišlo, že to více odpovídá skutečnému xylofonu, u kterého nelze délku tónu ovlivnit.

Porovnání skutečného a syntetizovaného zvuku xylofonu:



Pomocí MIDI Toolboxu jsem pak už jen vyrobil skladbu „Let čmeláka“.

B) Dvě oktávy durové stupnice

Pro výrobu stupnice jsem napsal skript „stupnice.m“, který k syntéze využívá funkci „synth.m“ a výsledný signál vyexportuje ve formátu .m4a. Kód skriptu „stupnice.m“:

```
fs = 48000; %vzorkovaci frekvence
noty = [40 42 44 45 47 49 51 52 54 56 57 59 61 63 64]; % jednotlivé klavesy
doba = .8;
freq = 440*2.^((noty-49)/12); % vypočet frekvence z čísla klavesy

y=[];
for k = 1:length(noty)
    x=synth(freq(k),doba,1,fs,14); % syntéza jednotlivých tonů
    y=[y x]; % zápis všech tonů za sebe
end

y=y./max(abs(y)); % normování signálu
audiowrite('stupnice_xyl.m4a',y,fs); % zápis do formátu m4a
```

C) Vlastní tvorba

Jelikož mě vytváření xylofonu bavilo, ale už jsem neměl další nápady, jak jej dále vylepšovat, rozhodl jsem se pro tvorbu jiného nástroje. Nakonec jsem zvolil flétnu. Postup byl velmi podobný jako u xylofonu. Zanalyzoval jsem několik vzorků zvuku flétny. Po prohlédnutí spektra bylo jasné, že nejlepší volbou bude aditivní syntéza, jelikož spektrum obsahovalo pouze několik spektrálních čar. Obálku jsem realizoval ze dvou částí. První měla tvar lichoběžníku a jejím úkolem bylo vytvořit náběh (attack) a uvolnění (release). Druhá část byla tvořena funkcí sinus, ta má za úkol simulovat kolísání amplitudy signálu – tremolo. Tato část obálky dobře dokreslí zvuk flétny, nesmí ale být příliš výrazná, jelikož potom zvuk začne připomínat spíše smyčcové nástroje. Na základě poslechu jsem oproti zkoumanému signálu ponechal pouze liché harmonické složky, zvuk potom působil více jako dechový nástroj.

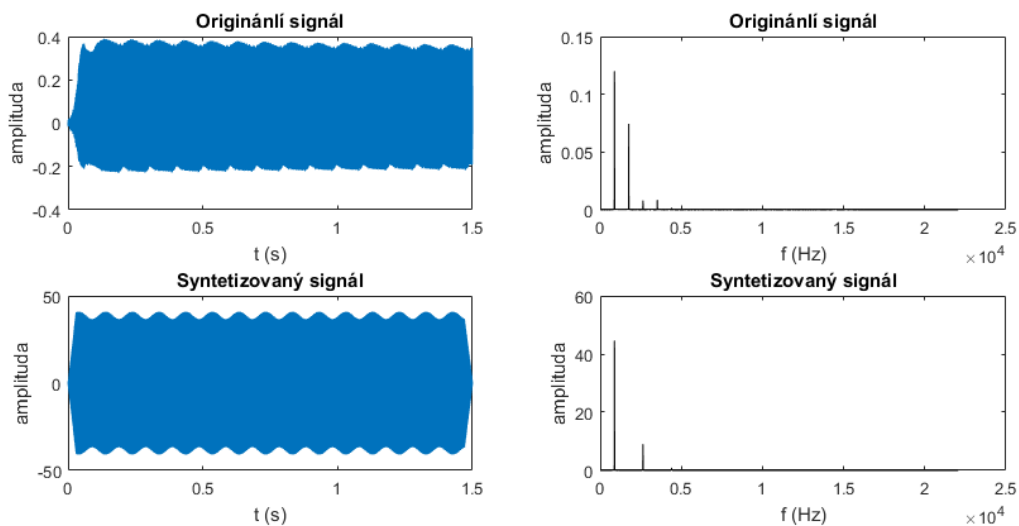
```
function y = flute(freq,doba,amp,fs)
% FLÉTNA

t = 0:1/fs:doba-1/fs; % časová osa

%obálka signálu
p=.1; % parametr doby nabehu a poklesu
o1=interp1(doba.*[0 p 1-p 1],[0 1 1 0],t); %vytvorí se lichobezník tvořící první část obálky
o2=46+2.5*sin(2*pi*10.1.*t-1); % sinusový průběh tvoří druhou část obálky
o=o1.*o2; % spojení do finální podoby obálky

ampl = [.99 .25 .02]; % amplitudy jednotlivých harmonických složek
x=o.*(ampl*sin(2*pi*[1 3 5]'*freq*t)); % aditivní syntéza signálu
y=amp.*x; % hlasitost signálu
```

Porovnání skutečného a syntetizovaného zvuku flétny:



Obdobným způsobem jako u xylofonu jsem vytvořil jednu oktávu durové stupnice a pomocí MIDI Toolboxu jsem nechal vygenerovat skladbu „Dolly Suite, Op. 56 - Berceuse“ - Gabriel Fauré.

2. Závěr

Při semestrální práci jsem zdokonalil svoje dovednosti v prostředí MATLAB a rozšířil si znalosti v oblasti syntézy audio signálů. S dosaženými výsledky nejsem zcela spokojen. Xylofon hraje dobře na vyšších frekvencích, slušně na středních ale na nižších frekvencích už je to horší. Zejména se to projeví při skladbě „Let čmeláka“, která má velký frekvenční rozsah. Syntéza flétny se myslím podařila lépe, ale to částečně i proto, že skladby pro flétnu neobsahují nízké frekvence.

3. Zdroje souborů

Vzorky xylofonu: <https://soundpacks.com/free-sound-packs/xylophone-samples-pack/>

Vzorky flétny: <https://soundpacks.com/free-sound-packs/flute-sound-kit/>

MIDI soubor pro vlastní tvorbu: <http://www4.osk.3web.ne.jp/~kasumitu/eng.htm>

4. Příložené soubory

Složka „vimrjan_2017“ obsahuje kompletní MIDI Toolbox s upravenou funkcí „synth.m“, přidanými funkcemi „xylophone.m“, „flute.m“ a skriptem „stupnice.m“. Zdrojové MIDI soubory jsou ve složce „midi“. Výstupní soubory jsou ve složce „result“.

Složky „Xylophone“ a „Flue“ obsahují skripty „Zobrazeni.m“, který přehraje přiložené vzorky zvuku nástroje a zobrazí jejich časový průběh a spektrum. A skripty „Xyl_adi.m“ (resp. „Flu_adi.m“) které slouží pro porovnání originálních a syntetizovaných zvuků.