

Syntéza Audio signálů: Semestrální práce

Martin Helmich

1. Kompozice – Eleonor Rigby (The Beatles)

V první části semestrální práce jsem se věnoval syntéze skladby "Eleanor Rigby" od skupiny The Beatles. K tomu jsem využil MIDI soubor, který mi poskytl pan profesor Roman Čmejla. Pro zpracování jsem použil nástroj MIDI toolbox 2.3, jenž byl součástí cvičení a vychází z verze od Kena Schutteho. Tento toolbox jsem si následně upravil tak, aby podporoval funkce mé syntézy.

Poskytnutá skladba se skládá ze 4 nástrojů – housle a basa (obojí ve variantě legato a staccato). V MIDI souboru jsou stopy pro staccato označeny jako pizzicato (MIDI ID 46), avšak s ohledem na to, že v originální skladbě je pro tyto stopy použito staccato, jsem implementoval právě tento styl. Pro syntézu byli použity 4 techniky syntézy popsané níže, které jsou řetězené za sebou pro vytvoření co nejvíce realistického dojmu.

Syntézu lze spustit spuštěním skriptu main.m. Syntézy jednotlivých nástrojů jsou pak přístupné ve formě funkcí ve složce „Generators“. Výsledný audio soubor celé skladby je uložen ve formátu m4a ve složce „result“ (BEATLES2.m4a). Originální MIDI soubor je dostupný ve složce „midi“. Složka „private“ obsahuje funkce MIDI toolboxu zmíněného výše.

1.1 Karplus-Strongův algoritmus

Pro co největší realističnost vibrující struny bylo využito Karplus-Strongova algoritmu, který je buzen šumem a pilovou vlnou. Vzhledem k tomu, že je nutné replikovat tah smyčce, tak bylo použito konstantní buzení, na který je aplikován IIR filtr pro vytvoření efektu vibrující struny. Analogicky si to můžeme představit tak, že tah smyčcem vytváří konstantní šumové buzení, které je obohaceno harmonickými složkami struny simulovanou pilovou vlnou. Inspirací pro tuto techniku byla syntéza klarinetu z 11 cvičení a plugin Karpy¹, vytvořený knihovnou Max for Live.

1.2 LPC analýza

Ačkoliv fyzikální modelování Karplus-Strongovým algoritmem vytváří realistické vibrace struny, tak výstup obsahuje až příliš široké frekvenční spektrum, které je potřeba vyfiltrovat. K navrhnutí filtru byla využita LPC analýza reálných nahrávek houslí a basy, jejíž a-koeficienty byli následně použity k filtraci signálu IIR filtrem. Nahrávky byli pořízeny v DAW softwaru Ableton Live 10 za využití balíčku Orchestral Strings². Licence veškerého softwaru vlastním. Nahrávky, které byli využité k LPC analýze, jsou dostupné ve složce Recordings.

LPC analýza byla provedena skriptem analyze_recording_LPC.m. A-koeficienty byli následně uloženy v .mat souborech, které jsou načítány pomocí matlab funkce load při syntéze jednotlivých nástrojů. Pro výběr signálu LPC jsem vždy našel periodu (nebo sled period), který byl následně analyzován pomocí funkce lpc v matlabu. Pro replikaci analýzy je třeba odkomentovat konkrétní sekce kódu, návod je dostupný v komentářích.

¹ <https://piumaxforlive.gumroad.com/l/karpy>

² <https://www.ableton.com/en/packs/orchestral-strings/>

Vzhledem k tomu, že (zejména) housle mají na různých frekvencích jinou barvu, tak byla LPC analýza provedená pro 4 oktávy – od G3 (196 Hz) až do G6 (1568 Hz). Při syntéze je pak použitý filtr, který je nejbližší potřebné frekvenci. Pro basu byla použita pouze jedna oktáva, jelikož má nižší frekvenční rozsah, ve kterém se zároveň barva zásadně nemění.

1.3 Ozvěna

Housle staccata byla implementována pomocí krátkého šumového impulzu o délce 70ms, který byl obohacen konvoluční ozvěnou. Impulzní odezva byla použita z balíčku Convolution Reverb³ ze softwaru Ableton Live 10. Následně byla upravena délka a obálka impulzní odezvy pro lepší replikaci reálného nástroje. „Wet“ signál byl smíchán s „dry“ signálem v poměru 3.5:96.5 (Tj. 3.5% je čistá ozvěna a 96.5% je originální signál bez ozvěny).

1.4 Vibrato

Vibrato bylo vytvořeno pomocí frekvenční a amplitudové modulace pro simulaci více realistického zvuku houslí. Vibrato je kontrolováno LFO (low frequency oscillator), jehož frekvence je doporučeno nastavit od 3 do 10 Hz. Lze zvolit i náběh (attack) vibrata, tj. za jak dlouho začne vibrato plně hrát. V nahrávce je vibrato zvoleno velmi slabé, aby ho posluchač sotva poznal, nicméně nahrávka díky němu nabývá větší realističnosti. Vibrato je použito pouze u houslí, nikoliv u basy.

³ <https://www.ableton.com/en/packs/convolution-reverb/>

2. MIDI Wavetable syntezátor

2.1 Úvod

Cílem této části semestrální práce bylo vytvořit real-time syntezátor v prostředí Matlab, který lze ovládat MIDI kontrolérem Komplete Kontrol A25 od značky Native Instruments⁴. Pro zpracování MIDI v reálném čase bylo využito Audio Toolboxu, který obsahuje funkce jako `midireceive()` k přijímání dat z MIDI kontroléru a strukturu `audioDeviceWriter`, která umožňuje zápis a přehrávání bufferu v reálném čase ve smyčce. Pro demonstraci funkcionality se **prosím podívejte na video** `MIDI_SYNTH_DEMO.mp4`, které je uloženo v kořenovém adresáři této části semestrální práce. Video jsem nahrál z toho důvodu, abyste mohli vydat fungující skript bez potřeby připojení syntezátoru.

V rámci práce jsou zpracovávány zprávy z klaviatury (`msg.Type == 'NoteOn'` a `msg.Type == 'NoteOff'`), `PitchBend` (`msg.Type == 'ControlChange'`) a knoflíků (`msg.Type == 'ControlChange'`). Konkrétně je pracováno s knoflíky CC1 a CC14-21. CC21 je vymezen pro vypínání audio streamu, zbytek je pro ovládání parametrů syntezátoru, viz obrázek 1. Každý knoflík CCx vrací hodnoty v rozmezí 0-127, v rámci zpráv z klaviatury jsou přijímány zprávy ohledně noty. Pro analýzu zpráv z MIDI byl vytvořen skript `midi_get_messages.m`.



Obrázek 1 Rozmístění ControlChange knoflíků a PitchBend

Celý syntezátor je možné spustit skriptem `main.m`, který používá funkce pro generování signálů a efektů z příslušných složek. Výstupní audio z běhu syntezátoru lze nahrát zaškrtnutím proměnné `recordAudio = true`; a určení času v proměnné `duration = 10`;. Pokud není připojen midi-syntezátor, tak skript přehraje základní demonstrační signál s aplikovaným flangerem. Vzorkovací frekvence je zvolena na standardních 44.1 kHz a velikost bufferu je zvolena na 256 vzorků.

UPOZORNĚNÍ: Syntezátor byl testován pouze na počítači Apple MacBook Pro s čipem M1 (2020) a operačním systémem Sequoia 15.2. Pokud bude skript spouštěn na jiném počítači, mohou se vyskytnout problémy s kompatibilitou.

2.2 Oscilátory

Generování zvuku je vyřešené pomocí oscilátorů, které jsou definované následující strukturou:

⁴ <https://www.native-instruments.com/en/products/komplete/keyboards/komplete-kontrol-a25-a49-a61/>

```

oscDefaultStruct = struct( ...
    'midiNote', 0, ...
    'phase', 0, ...
    'amplitude', 1, ...
    'current_osc_length', 0, ...
    'releaseFlag', false, ...
    'releaseVec', zeros(round(ADSR(4)*fs), 1));
oscilators = repmat(oscDefaultStruct, numberOfOscilators, 1);

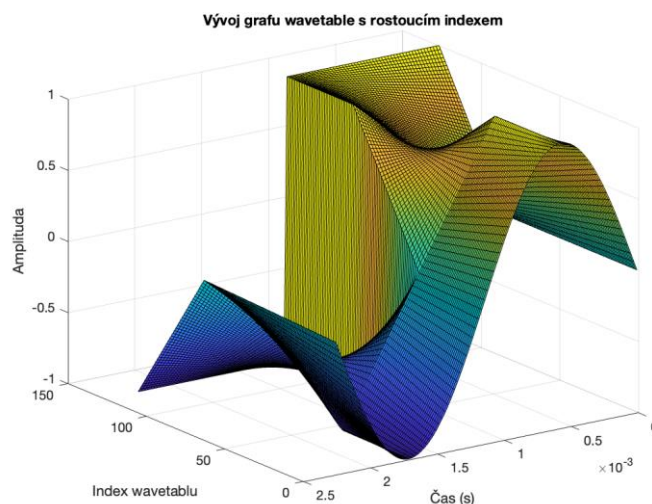
```

Skript definuje dále množství oscilátorů, což je „množství not“, které je syntezátor schopný zahrát najednou. Syntezátor byl testován s počtem oscilátorů nastavený na 7. Při každém stisknutí MIDI klávesy je proskenované pole oscilátorů a pokud je nalezen nějaký neaktivní, tak je mu přiřazena nota a amplituda z MIDI zprávy. Každý oscilátor si zároveň pamatuje svojí fázi, která je potřebná pro udržení kontinuity mezi buffery. Další proměnné jsou udržované pro správnou funkcionalitu ADSR obálky, viz kapitola 2.5.

Zvuk oscilátorů je definovanými 2 režimy, kde každý z nich odpovídá jinému druhu syntézy. Defaultně je spuštěný v režimu tabulkové syntézy (viz kapitola 2.3), nicméně po vytažení pitchbend knoflíku nahoru je přepnuto do režimu aditivní syntézy, pomocí které je implementován jeden nástroj – lesní roh (viz kapitola 2.4). Z režimu lesního rohu je možné přepnout syntezátor zpět do režimu wavetable vytažením pitchbend knoflíku dolů.

2.3 Wavetable syntéza

Jako primární zdroj zvuku pro oscilátory byla zvolena tabulková (wavetable) syntéza, která je ovládána knoflíkem CC1. Tabulka byla vytvořena tak, aby nabízela plynulé prolnutí mezi sinusovou, trojúhelníkovou, pilovou a čtvercovou vlnou, viz obrázek 2. Tabulka obsahuje 127 vln, které jsou postupnou lineární interpolací 4 zmíněných základních vln. Skript na vytvoření této tabulky včetně interaktivního 3D grafu z obrázku 2 lze nalézt pod jménem `create_basic_wavetable.m`.



Obrázek 2 Vývoj vlny s rostoucím indexem tabulky

Tabulka je uložena v soboru `basic_wavetable.mat` a je k ní přístupováno vždy při tvorbě oscilátoru ve funkci `generateWavetable.m`. Tato funkce implementuje tabulkovou syntézu lineární interpolací na základě momentálního indexu tabulky, který je daný proměnnou `wavetableIndex` ve skriptu `main.m` a je měněn knoflíkem CC1. Tento princip umožňuje uživateli kontinuálně volit množství harmonických složek syntezátoru, který odpovídá jeho požadovanému zvuku.

2.4 Aditivní syntéza

Jako další nástroj pro kontrolér byla zvolena aditivní syntéza lesního rohu. Aditivní syntéza byla vytvořena pomocí skriptu `additive_synth_analysis.m`, který vychází ze skriptů funkcí probíraných na cvičení 3 (konkrétně funkce `analyze.m` byla přebrána přímo ze cvičení). Nahrávka lesního rohu byla přebrána ze stránek předmětu⁵. Funkce, která implementuje aditivní syntézu v reálném čase se jmenuje `generateFrenchHorn`.

Na tento zvuk lze v MIDI kontroléru přepnout pomocí vytažení knoflíku „PitchBend“ nahoru, který se nachází nalevo vedle klaviatury. Přepnutím se zároveň nastaví defaultní obálka a vibrato (vytvořené pomocí flangeru s `flanger_mix == 1`) pro co nejvíce realistickou repliku reálného nástroje. Frekvenční rozsah je v rozmezí od E2 do F5, tj. MIDI ID od 40 do 77, tj. pokud uživatel stiskne klávesu mimo rozsah, tak nástroj nebude hrát.

2.5 ADSR obálka

Implementace obálky byla o něco komplexnější, jelikož byla rozdělena na 2 části: Attack-Decay-Sustain a Release. To je z toho důvodu, aby nota byla hrána s hlasitostí definovanou parametrem Sustain do té doby, co uživatel **nepustí** odpovídající klávesu. V takovém případě se oscilátor přepne do režimu Release a syntezátor dohraje notu bez toho, že by uživatel držel jakoukoliv klávesu.

Hodnoty o ADSR obálce jsou uloženy ve proměnné `ADSR = [0.01 0.5 0.3 0.1]`; s tím, že hodnoty attack, decay a release odpovídají časové délce jednotlivých segmentů v sekundách a sustain odpovídá hlasitosti noty, když se dostane do této fáze.

Ve chvíli, co uživatel stiskne klávesu (tj. přijímám midi zprávu `msg.Type == 'NoteOn'`), tak je na notu aplikovaná obálka ADS, tj. attack-decay-sustain. Obálka je aplikovaná funkcí `apply_ADSR` a informace o fázi obálky je uložena jako `'current_osc_length'` ve structu oscilátoru. Tato hodnota je inkrementována o velikost bufferu vždy na konci jeho zpracování. Jakmile je `'current_osc_length'` větší než délka ADS obálky, tak je hlasitost noty nastavená na úroveň sustain, tj. `ADSR(3)`.

Ve chvíli, co uživatel pustí klávesu (tj. přijímám midi zprávu `msg.Type == 'NoteOff'`), tak je oscilátor převeden do režimu Release označením `'releaseFlag' = true` ve structu oscilátoru. V tuto chvíli je zároveň vytvořena release obálka, která vychází z poslední hodnoty ADS obálky. To je z toho důvodu, aby bylo zajištěné spadnutí z aktuální hodnoty, když uživatel nedržel klávesu dost dlouho na to, aby doběhla ADS obálka do konce. Vektor obálky je pak z tohoto důvodu uložen přímo ve structu oscilátoru v proměnné `'releaseVec'`. Jakmile release obálka doběhne do konce, tak je oscilátor vypnut a je volný pro zahrání další noty.

Parametry ADSR obálky je možné měnit za běhu syntezátorů pomocí knoflíků CC14 až CC17 (tj. CC14 = attack, CC15 = decay, CC16 = sustain a CC17 = release).

2.6 Flanger

Na výstup lze aplikovat efekt flangeru. Efekt je aplikovaný na všechny oscilátory najednou. Implementace flangeru obsahuje 3 parametry, které jsou ovládané knoflíky CC18 až CC20 následujícím způsobem:

- CC18: Flanger mix – kontroluje poměr mezi originálním a flanger signálem, tj.
 - `flanger_mix = 0.0` odpovídá plně originálnímu signálu
 - `flanger_mix = 0.5` odpovídá půl na půl originálnímu a flanger signálu
 - `flanger_mix = 1.0` odpovídá plně flanger signálu
- CC19: Flanger max delay – kontroluje maximální zpoždění při flangeru. Defaultně je maximální hodnota nastavená na 8ms, nicméně pokud velikost bufferu odpovídá

⁵ <https://sami.fel.cvut.cz/syn/>

kratšímu času než 8ms (například 32), tak je nastavený na délku bufferu. To je z toho důvodu, že pro tvorbu flangeru si pamatují pouze jeden buffer dozadu. V budoucím vývoji lze tuto funkcionalitu rozšířit a pamatovat si i vzdálenější hodnoty.

- CC20: Flanger LFO frekvence: Frekvence low-frequency-oscilátoru, standardně nastavená v mezích od 0 do 4 Hz.