

# ISS Projekt 2022/21

Honza Brukner a Honza Černocký, ÚPGM FIT VUT

October 14, 2022

## 1 Úvod

V projektu bude vaším úkolem vytvořit co nejvěrnější syntetické piano jehož zvuková charakteristika bude komprimovaná pouze do několika parametrů. K dispozici máte nahrávky jednotlivých tónů, které bude potřeba analyzovat a následně vygenerujete skladbu na základě dodaného souboru.

Projekt je individuální a je možno řešit v Python-u, Matlab-u, Octave, jazyce C nebo v libovolném jiném programovacím či skriptovacím jazyce. Je možné použít libovolné knihovny. Projekt se nezaměřuje na “krásu programování”, není tedy nutné mít vše úhledně zabalené do okomentovaných funkcí, ošetřené všechny chybové stavy, atd. Důležitý je výsledek. **Kód musí prokazatelně produkovat výsledky obsažené ve Vašem protokolu.**

V zadání se občas vyskytují funkce, či jiné části kódu, jako nápovědy. Tyto ukázky jsou v Python-u s použitím knihovny `numpy` (přesněji `import numpy as np`).

## 2 Vstup

V souboru <https://www.fit.vutbr.cz/study/courses/ISS/public/proj2022-23/klavir.wav> máte k dispozici wav soubor se všemi tóny klavíru<sup>1</sup>. Vzorkovací frekvence je  $F_s = 48$  kHz a nahrávka má jeden kanál. Tóny jsou od C2 (MIDI 24) do C9 (MIDI 108), celkem jich je tedy 85, každý tón má 2 sekundy. Přehled MIDI značení a frekvencí tónů máte v <https://www.fit.vutbr.cz/study/courses/ISS/public/proj2022-23/midi.txt><sup>2</sup>. Dále máte v <https://www.fit.vutbr.cz/study/courses/ISS/public/proj2022-23/personal/xlogin00.txt> MIDI označení tónů, na kterých budete demonstrovat průběžné řešení projektu (`xlogin00` je Váš login, pro studenty FSI je to osobní číslo VUT).

## 3 Odevzdání projektu

bude probíhat do informačního systému VUTIS (studenty FSI prosím o zaslání řešení emailem) ve dvou souborech:

1. `xlogin00.pdf` nebo `xlogin00_e.pdf` (kde “`xlogin00`” je Váš login či číslo studenta) je protokol s řešením.
  - V záhlaví prosím uveďte své jméno, příjmení a login.
  - Pak budou následovat odpovědi na jednotlivé otázky — obrázky, numerické hodnoty, komentáře.
  - U každé otázky uveďte stručný postup - může se jednat o kousek okomentovaného kódu, komentovanou rovnici nebo text. Není nutné kopírovat do protokolu celý zdrojový kód. Není nutné opisovat zadání či teorii, soustřeďte se přímo na řešení. Je-li v zadání “zobrazte”, znamená to, že výsledek chceme vidět v protokolu.
  - Pokud využijete zdroje mimo standardních materiálů (přednášky, cvičení a studijní etapa projektu ISS), prosím uveďte, odkud jste čerpali (např. článek, skripta, dokumentace `numpy`, `Matlab`, `scipy`, ...).
  - Protokol je možné psát v libovolném systému (Latex, MS-Word, Libre Office, ...), můžete jej psát i čitelně rukou, dolepit do něj obrázky a pak oskenovat. Pokud budete tvořit protokol přímo jako Python notebook (doporučeno), prosíme Vás o jeho export do PDF.
  - Protokol může být česky, slovensky nebo anglicky. Budete-li psát anglicky, prosíme, abyste nazvali výsledný soubor `xlogin00_e.pdf`. Za angličtinu v protokolu není žádné zvýhodnění ani penalizace, slouží nám jen pro výběr opravujících.
2. `xlogin00.tar.gz` je komprimovaný archiv obsahující následující adresáře:
  - `/src` - Vaše zdrojové kódy – může se jednat o jeden soubor (např. `reseni.py` nebo `reseni.ipynb`), o více souborů či skriptů nebo o celou adresářovou strukturu.

<sup>1</sup>Autoři zadání děkují Adamovi Černockému za vygenerování souboru s klavírními tóny pomocí software FL Studio.

<sup>2</sup>vypreparováno z [https://www.inspiredacoustics.com/en/MIDI\\_note\\_numbers\\_and\\_center\\_frequencies](https://www.inspiredacoustics.com/en/MIDI_note_numbers_and_center_frequencies), používáme anglické značení tónů.

- /audio - audio soubory – ve formátu WAV, bitová šířka 16 bitů, bez komprese, se zadanou vzorkovací frekvencí.
3. Projekt je **samostatná práce**, proto budou Vaše zdrojové kódy křížově korelovány a v případě silné podobnosti budou vyvozeny příslušné závěry.
  4. Silná korelace s kódy ze studijní etapy projektu, z Python notebooků studijní opory a z přednášek je v pořádku, nemusíte tedy měnit názvy proměnných, přepisovat komentáře, atd.

## 4 Standardní zadání

Úspěšné řešení těchto bodů zadání vede k plnému počtu bodů za projekt, tedy 18ti.

Všechny grafy musí mít popsané osy. Pokud vykreslujete v časové doméně, osa x bude zobrazovat čas v (mili,mikro,...) sekundách, pokud ve spektrální, osa x bude ve frekvencích v Hz.

Pokud není řečeno jinak, spektrum zobrazujte jako log PSD (logaritmus druhé mocniny absolutních hodnot DFT) od 0 Hz do poloviny vzorkovací frekvence. Jelikož signál ustředňujeme, 0. koeficient DFT bude 0 a kazilo by to vykreslování (logaritmus 0 je  $-\infty$ ), můžete tedy pro potřeby vykreslování k PSD přičíst nějakou malou konstantu (třeba  $10^{-5}$ ).

### 4.1 Základy – 2 body

Načtěte všechny signály a vyberte 0.5 s dlouhý úsek ze stabilní části signálu, tedy té, kde zní pouze tón a už není slyšet úder kladívka (doporučujeme přeskočit první 0.25 s signálu). Pokud budete pracovat v Pythonu, můžete načtení provést takto:

---

```
import numpy as np
import soundfile as sf
MIDIFROM = 24
MIDITO = 108
SKIP_SEC = 0.25
HOWMUCH_SEC = 0.5
WHOLETONE_SEC = 2
howmanytones = MIDITO - MIDIFROM + 1
tones = np.arange(MIDIFROM, MIDITO+1)
s, Fs = sf.read('klavir.wav')
N = int(Fs * HOWMUCH_SEC)
Nwholetone = int(Fs * WHOLETONE_SEC)
xall = np.zeros((MIDITO+1, N))    # matrix with all tones - first signals empty,
                                   # but we have plenty of memory ...

samplefrom = int(SKIP_SEC * Fs)
sampleto = samplefrom + N
for tone in tones:
    x = s[samplefrom:sampleto]
    x = x - np.mean(x)            # safer to center ...
    xall[tone,:] = x
    samplefrom += Nwholetone
    sampleto += Nwholetone
```

---

Zobrazte 3 periody vašich tří tónů v ustálené části a spočítejte a vykreslete DFT celého 0.5 s dlouhého úseku. Vaše tóny uložte jako audio/a\_orig.wav, audio/b\_orig.wav, a audio/c\_orig.wav

### 4.2 Určení základní frekvence – 3 body

Spočítejte základní frekvenci všech tónů a srovnejte je s frekvencemi definovanými MIDI. Základní frekvenci můžete spočítat pomocí autokorelace nebo DFT.

Pro vaše tři tóny zobrazte graf s použitou metodou a vyznačte v něm, kde/jak jste našli  $f_0$ , pokud jste použili nějaký výpočet, uveďte jej. Pokud vidíte rozdíly mezi očekávanou (MIDI) a skutečnou frekvencí, komentujte z čeho mohou plynout. Rozladění piana, přesnost metody (jaká je?), ...

Hint: Autokorelace funguje lépe pro tóny s nižší frekvencí, kde má DFT občas tendenci selhávat a naopak. Máte k dispozici MIDI frekvenci, pomocí které si můžete zkontrolovat, jestli jste frekvenci určili správně.

### 4.3 Zpřesnění odhadu základní frekvence $f_0$ – 3 body

Pomocí DTFT (ne DFT!) zpřesněte odhad skutečné základní frekvence všech tónů.

Můžete například vypočítat koeficienty pro frekvence rozložené 100 centů<sup>3</sup> okolo nejbližší MIDI frekvence. Další možnost je použít frekvence v rozpětí  $\pm 2$  koeficientů DFT kolem odhadnutého základního tónu. U některých tónů (hlavně na nižších frekvencích) se stává, že je první koeficient FR nižší než druhý, můžete tedy také vyhledat, na které frekvenci leží druhý koeficient a tu potom podělit 2. Pokud vám příliš překáží laloky sinc, způsobené ostrým obdélníkovým oknem, můžete použít okénkovou funkci, např. `np.hamming`.

Srovnajte takto získaný odhad s původním, napište, kterou možnost používáte, případně jestli je kombinujete a uveďte, jakým způsobem DTFT implementujete.

### 4.4 Reprezentace klavíru – 3 body

Reprezentujte každý tón použitím 10 floating point čísel.

Možný postup: Aproximujte Fourierovu řadu (FR) pomocí DTFT, spočítejte koeficienty na násobcích základní frekvence ( $1f_0 \dots 5f_0$ ), uložte modul a fázi, alternativně můžete spočítat pouze moduly pro prvních 10 násobků a neukládat fáze. Jelikož pracujeme s reálnými signály, opět vyhledejte přesnou polohu koeficientu z intervalu okolo očekávané frekvence.

Zobrazte modul DFT vašich tří tónů, do stejných grafů vyznačte koeficienty (budou to body na křivce), frekvenční osu vykreslete od 0 do  $11 \times$  nejbližší MIDI frekvence v Hz (v protokolu budou 3 grafy, jeden pro každý tón).

### 4.5 Syntéza tónů – 3 body

Syntetizujte signály odpovídající Vaším třem tónům. Použijte stejnou vzorkovací frekvenci jako má původní nahrávka.

Pro syntézu můžete využít FR (uvědomte si, že  $c_{-i} = c_i^*$ ; výsledek je reálný, ale díky nepřesnostem výpočtu bude pořád komplexní číslo, imaginární části se můžete zbavit pomocí `np.real()`) nebo využijte prosté sčítání cosinusovek (jak vyřešíte amplitudu?)

Uložte 1 sekundu vygenerovaného signálu pro každý z vašich tří tónů do souborů `audio/a.wav`, `audio/b.wav`, a `audio/c.wav`.

Srovnajte v grafu 10 period vámi vygenerovaného a originálního signálu na vašich třech tónech. Signály synchronizujte (ručně, korelací, ...) tak, aby začínaly se stejnou fází. Komentujte rozdíly.

### 4.6 Generování hudby – 3 body

V <https://www.fit.vutbr.cz/study/courses/ISS/public/proj2022-23/skladba.txt> máte k dispozici skladbu zpracovanou z MIDI souboru s řádky v následujícím formátu, kde každý řádek odpovídá jednomu stisknutí klávesy:

od [ms]	do [ms]	MIDI	hlasitost
---------	---------	------	-----------

Vygenerujte skladbu. Pozor, nahrávka je polyfonní, zní v ní tedy několik tónů současně. Vytvořte dvě nahrávky, jednu se vzorkovací frekvencí 8000 Hz a druhou s 48000 Hz. Z obou uložte prvních 10 s jako `audio/out_8k.wav` a `audio/out_48k.wav`. Pro generování na 8000 Hz dejte pozor, abyste neporušili vzorkovací teorém - nesmíte generovat harmonické nad  $\frac{F_s}{2}$ .

Pokud si budete chtít vygenerovat nějaké skladby pro potěšení, můžete se podívat zde: [https://github.com/Lemlak/ISS22\\_table\\_generator](https://github.com/Lemlak/ISS22_table_generator).

### 4.7 Spektrogram – 1 bod

Zobrazte spektrogram prvních 10 s vygenerovaného signálu pro obě vzorkovací frekvence, komentujte, co vidíte. Použijte velikost okna 0.03 ms, překryv 0 ms a počet vzorků DFT 2048 pro 44.1 kHz a 512 pro 8 kHz. Můžete využít knihovni funkce `scipy.signal.stft` nebo `scipy.signal.spectrogram`.

<sup>3</sup>[https://cs.wikipedia.org/wiki/Cent\\_\(hudba\)](https://cs.wikipedia.org/wiki/Cent_(hudba))

## 5 Bonusový úkol

Za tyto úkoly nejsou body, ale bude Vás také hrát pocit skutečného porozumění signálům. Autor/ka nejlepšího řešení dostane láhev kvalitního francouzského červeného vína. Nejlepší řešení bude hodnoceno podle kvality zvuku a nápaditosti řešení.

### 5.1 Vylepšení zvuku

Pohrajte si generováním tónů tak, aby zněly co nejlépe.

Možnou úpravou je například práce s ADSR obálkou<sup>4</sup>. Tu můžete vypočítat pro každý tón zvlášť nebo vytvořit jednu univerzální. Výpočet ADSR můžete provést např. ručně, z grafu tónu nebo pomocí odhadu krátkodobé energie.

Popište svůj postup a opět uložte prvních 10 s Vaší nejlepší skladby se vzorkovací frekvencí 48 kHz do `audio/bonus.wav`.

---

<sup>4</sup>[https://en.wikipedia.org/wiki/Envelope\\_\(music\)](https://en.wikipedia.org/wiki/Envelope_(music))