

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

GENERÁTOR HARMONICKÝCH ZVUKŮ A AKORDŮ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN BEDNÁŘ



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNologiÍ**
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

GENERÁTOR HARMONICKÝCH ZVUKŮ A AKORDŮ

GENERATOR OF HARMONIC SOUNDS AND CHORDS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN BEDNÁŘ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JIŘÍ SCHIMMEL, Ph.D.

BRNO 2013



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Jan Bednář

ID: 136501

Ročník: 3

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Generátor harmonických zvuků a akordů

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a pomocí technologie VST realizujte generátor zvukového signálu typu VST Instrument s vlastním grafickým rozhraním, který bude pomocí prvků grafického rozhraní nebo připojené MIDI klaviatury generovat základní tón a vyšší harmonické pro několik not současně. Tyto noty budou zadávány pomocí tóniny a typu akordu. Jednotlivé harmonické i jednotlivé noty v akordu budou mít nastavitelnou úroveň. Porovnejte několik algoritmů generování harmonických signálů z hlediska jejich výpočetní náročnosti a zkreslení.

DOPORUČENÁ LITERATURA:

- [1] BOULANGER, R. The Audio Programming Book. The MIT Press, 2010. ISBN 978-0262014465.
- [2] GEIST, B. Akustika, jevy a souvislosti v hudební teorii a praxi. Nakladatelství MUZIKUS, 2005. ISBN 80-86253-31-7
- [3] VST Plug-Ins Software Developer Kit 3.0.1. Steinberg Media Technologies GmbH. 2008.

Termín zadání: 11.2.2013

Termín odevzdání: 5.6.2013

Vedoucí práce: Ing. Jiří Schimmel, Ph.D.

Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Hlavní cíl této bakalářské práce je vytvoření VST zásuvného modulu (plug-inu), který bude schopný generovat zvuk pro několik tónů současně. V této práci je popsána stručná teorie hudební akustiky, která se zaměřuje na základní vztahy mezi tóny, harmonickými složkami a jejich vlivem na zvukovou barvu. Dále je popsán způsob fungování temperované soustavy ladění, vztahy mezi jednotlivými intervaly a způsob vytváření akordů různých tónin na základě jednotné půltónové masky akordu. Zmíněny jsou základy technologie VST, prostředky pro programování pomocí VST SDK a základní zprávy technologie MIDI potřebné pro její implementaci do VST plug-inu.

KLÍČOVÁ SLOVA

generátor, tón, akord, VST, zvuk, harmonická složka

ABSTRACT

Major aim of this bachelor's thesis is creation of VST plug-in module, that is able to generate sound for more tones in the same time. This thesis describes a brief theory of music acoustics which aims to show basic relations between tones, their higher harmonics and impact on the sound color. The function of temperament system of tuning, relations between intervals and way of making chords is described. Next things mentioned in this thesis are basics of VST technology, how to programme with VST SDK and basic MIDI messages for implementation in VST plug-in.

KEYWORDS

generator, tone, chord, VST, sound, harmonics

BEDNÁŘ, Jan *Generátor harmonických zvuků a akordů*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 37 s. Vedoucí práce byl Ing. Jiří Schimmel, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Generátor harmonických zvuků a akordů“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Jiřímu Schimmelovi, Ph.D. za odborné vedení, konzultace a podnětné návrhy k práci.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Výzkum popsáný v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....
(podpis autora)

OBSAH

Úvod	8
1 Hudební teorie	9
1.1 Tón a alikvoty	9
1.2 Hladina akustického tlaku	10
1.3 Lidský sluch	10
1.4 Intervaly	11
1.5 Temperovaná soustava ladění	12
1.6 Akordy	12
2 VST - Virtual Studio Technology	16
2.1 Implementace VST	16
2.2 Hostitelské aplikace	17
2.3 Zvuková karta a ovladače ASIO	17
3 MIDI	18
3.1 MIDI zprávy	18
3.2 Využití	19
4 Vývoj kódu	21
4.1 Postup vývoje	21
4.2 Popis funkcí	23
4.3 Generování zvuku	23
4.4 Volba parametrů	25
4.5 Spuštění a ovládání aplikace	25
5 Analýza generovaného signálu	29
6 Závěr	34
Literatura	35
Seznam symbolů, veličin a zkratk	36
A Obsah přiloženého CD	37

ÚVOD

Základním úkolem této práce je vytvoření funkčního generátoru jednotlivých tónů, které budou mít měnitelnou hlasitost svých harmonických složek a tyto tóny budou v souzvuku tvořit akord. Takovýto program je vhodný na demonstraci základních oblastí hudební akustiky a přispívá k jejich pochopení zejména díky tomu, že uživatel každou provedenou změnu okamžitě uslyší. Druhým úkolem je implementace velmi často používaného rozhraní pro hudební nástroje MIDI, díky kterému je generátor možné ovládat pomocí klaviatury elektronického hudebního nástroje. Samotný kód je vytvořen v jazyce C++ technologií VST firmy Steinberg.

Práce se v úvodu věnuje základům hudební akustiky, vztahy mezi tóny a laděním, aby bylo následně možné přesněji popsat matematické vztahy implementované v kódu. Další část je věnovaná principům technologie VST a práce se základní šablonou pro její vývoj. Ve třetí části naleznete informace o MIDI a základních, pro tento program potřebných, MIDI zprávách. V závěru jsou popsány nejdůležitější funkce kódu pro generování zvuku. Poslední část je věnována analýze signálů generovaných zásuvným modulem, grafům, porovnáním přesnosti a výčtu nedostatků tohoto generátoru.

Na vztahy tónů, ladění a akordy není v této práci brán pohled z oblasti hudební, ale spíše se jen zaměří na velmi základní odvození, nutné pro definování akordu ze základního tónu, tak aby bylo možno vygenerovat libovolný akord v libovolné stupnici i bez znalosti obsáhlé hudební vědy. Z tohoto poté vychází i samostatná programová část řešení, která je k tomu uzpůsobena objektovým návrhem. Práce se věnuje jen nejnutnějším funkcím pro pochopení funkce programu, pro zjištění více informací o programování aplikací pomocí VST SDK (software development kit – sada nástrojů pro vývoj programu) doporučuji prostudovat zdroj [13].

1 HUDEBNÍ TEORIE

Pro seznámení s pojmy a pochopení důležitých informací nutných pro porozumění výslednému kódu je v prvních kapitolách této práce uvedena teorie z oblasti hudební akustiky, ladění a základů hudebních pravidel při tvorbě akordů.

1.1 Tón a alikvoty

Základním prvkem jakékoliv hudby či hudebního signálu je tón, který je charakterizován periodickým dějem, pro který platí vztah

$$f(t) = f(t + T), \quad (1.1)$$

kde t je čas a T perioda. Pokud si vezmeme nejjednodušší periodický děj, tedy harmonický průběh, můžeme hovořit o čistém tónu, jehož vztah vypadá následovně

$$f(t) = A_1 \sin(2\pi ft + \phi_1), \quad (1.2)$$

ve kterém A_1 a ϕ_1 jsou konstanty a f je kmitočet $f = \frac{1}{T}$. V hudbě se ale tento tón nevyskytuje často, jelikož vytvořit tón pouze o základní frekvenci umí pouze pár nástrojů nebo např. tónový generátor. Všechny složitější tóny (periodické, neharmonické) lze rozložit pomocí Fourierovy analýzy na řadu vyšších harmonických složek podle vztahu

$$f(t) = \sum_{n=1}^{\omega} A_n \sin(n2\pi ft + \phi_n), \quad (1.3)$$

kde n je kladné celé číslo. Pro $n = 1$ dostaneme výraz (1.2), který je prvním harmonickým kmitočtem, také nazývaný jako základní tón. Pro $n = 2, 3, 4, \dots$ obdržíme vztah pro vyšší harmonické neboli alikvotní kmitočty, zkráceně alikvoty. Poměr amplitud A_1, A_2, A_3, \dots těchto alikvot udává zásadní parametr tónu, což je barva tónu. Pokud signál obsahuje více alikvot, zdá se být pro posluchače ostřejší, lesklejší. Pokud alikvoty potlačíme, změní se i barva tónu. Obsah vyšších harmonických, barva tónu, je tedy jedním z důležitých kritérií pro rozlišení jednotlivých zdrojů tónů, nicméně není kritériem jediným. [5]

Jak je ze vztahu (1.2) patrné, v rovnici se vyskytuje i posun počáteční fáze ϕ . Ohmův-Helmholtzův psychoakustický zákon říká, že lidské ucho neregistruje rozdíl ve velikosti počáteční fáze, takže 2 signály o stejné frekvenci a amplitudách s posunem fáze první harmonické a vyšších harmonických budou lidskému uchu znít stejně. Dnes už víme, že to není zcela pravda, pro zjednodušení a zpracování zvuku se s ním ale stále počítá. [12, 4]

Dalšími faktory pro rozeznání zdroje zvuku jsou nejen alikvoty ale také časový průběh dynamiky, tvar přechodových dějů a obsah neharmonických kmitočtů (šumy,

hluky). [5] V této práci se ale dynamikou a přechodovými ději zabývat nebudu, neboť cílem je především generování stálých tónů, u nichž je možné měnit jejich hlasitost a hlasitost jejich alikvotů.

1.2 Hladina akustického tlaku

Abychom mohli definovat hlasitost, musíme první definovat samotný akustický tlak a hladinu akustického tlaku. „Akustickým tlakem se nazývá rozdíl mezi okamžitou velikostí celkového tlaku p_C v daném bodě zvukového pole za přítomnosti zvuku a statickou hodnotou atmosférického (barometrického) tlaku p_{00} . Je to tedy střídavá složka tlaku, která je superponovaná k barometrickému tlaku za přítomnosti zvuku. Z toho je patrné, že v každém bodě zvukového pole se hodnota celkového tlaku, který je součtem barometrického a akustického tlaku, bude měnit v čase.“ Značí se p a jeho jednotkou je Pascal (Pa). [12]

Hladina akustického tlaku L_P je poté

$$L_P = 10 \log \frac{p^2}{p_0^2} = 20 \log \frac{p}{p_0}, \quad (1.4)$$

jenž je vztažena k hodnotě $p_0 = 2 \times 10^{-5}$ Pa a jeho jednotka je decibel [dB]. Anglicky se označuje jako *Sound Pressure Level*, zkratkou SPL v dolním indexu nebo závorce, aby se zdůraznilo, že se jedná právě o údaj vztažený k hodnotě 2×10^{-5} Pa a označuje se např. dB_{SPL} , $\text{dB}_{(\text{SPL})}$. V našem případě číslcového výpočtu se ale budeme setkávat s úrovní signálu, která je vztažena k maximální hodnotě, kterou je systém schopen zpracovat, v technologii VST je to hodnota 1. Jednotka je dBFS (Full Scale, plný rozsah). Nejhlasitější zvuk bude mít tedy hladinu akustického tlaku o hodnotě $L_P = 0$ dBFS, všechny ostatní signály budou vyjádřeny se zápornou hodnotou. Pokud dojde k překročení hladiny 0 dBFS, vznikne nelineární zkreslení, což je vznik vyšších harmonických složek signálu které signál původně neobsahoval. [12]

Hlasitost zvuku je podle ČSN 01 1600 „vlastnost sluchového vnímání, podle které lze uspořádat zvuky na stupnici v rozsahu od tichých po hlasité.“

1.3 Lidský sluch

Základní rozsah slyšení je udáván od 20 Hz až po 20 kHz, ale pro každého jedince se může lišit, zejména s věkem klesá horní hranice slyšení. Pro lidské ucho je hlasitost zvukového signálu závislá na frekvenci tohoto signálu. Ucho je nejcitlivější na kmitočty okolo 3 kHz a tato citlivost postupně klesá na obě strany spektra. Signál o frekvenci 1 kHz a úrovni 10 dB se může zdát stejně hlasitý jako signál o frekvenci 50 Hz a úrovni 50 dB. Další vlastností lidského ucha je tzv. maskování zvuků. Pokud

mají 2 signály blízkou frekvenci lidské ucho uslyší pouze jeden tón, pokud bude hlasitost jednoho signálu výrazně vyšší. Může také vzniknout případ, kdy lidské ucho uslyší tón i frekvenci, která bude zhruba průměrem těchto dvou frekvencí a úroveň tohoto tónu se bude periodicky měnit. [5, 12]

1.4 Intervaly

Základní hudební tóny leží v rozsahu 16 - 4200 Hz, nicméně se nepoužívají všechny kmitočty spojitě, ale tóny jsou vybírány podle určitých pravidel. V dnešní evropské hudbě se používají tyto tři soustavy:

- Pythagorejská (kvintová),
- přirozená (terciová),
- temperovaná (dvanáctitónová).

Vzdálenost dvou tónů v každé soustavě se nazývá *interval* a je dán poměrem vyššího a nižšího kmitočtu [5]

$$I = \frac{f_2}{f_1}. \quad (1.5)$$

Základní názvy intervalů se označují latinskými číslovkami, jejich přehled je uveden v tab. 1.1:

Tab. 1.1: Základní názvy intervalů [7]

1	Dva tóny stejného stupně tvoří	<i>prímu</i> ,
2	od I. k II. stupni se nazývá	<i>sekunda</i> ,
3	od I. k III. stupni se nazývá	<i>tercie</i> ,
4	od I. k IV. stupni se nazývá	<i>kvarta</i> ,
5	od I. k V. stupni se nazývá	<i>kvinta</i> ,
6	od I. k VI. stupni se nazývá	<i>sexta</i> ,
7	od I. k VII. stupni se nazývá	<i>septima</i> ,
8	od I. k VIII. stupni se nazývá	<i>oktáva</i> ,

Nejdůležitějším intervalem je oktáva (8), což je dvojnásobná frekvence, tedy druhá harmonická složka. Pokud je tento interval zahrán zároveň, nemusí být správně rozeznán nezkušeným posluchačem, pokud se ale zahraje postupně, je rozeznán snadno. Pokud je vzdálenost dvou intervalů větší jak 8, leží v jiných oktávách. Oktáv se v hudbě používá zhruba 8. Slyšitelný rozsah je 10 oktáv. Jednotlivé tóny v jedné oktávě mají pojmenování *c, d, e, f, g, a, h, c*. [5]

U intervalů ale můžeme kromě základního názvu určovat i název jakosti intervalů, aby bylo možné zjistit, kolik půltónů přesně tento interval obsahuje. Podle jakosti dělíme intervaly na

- základní
čisté (1, 4, 5, 8),
velké a malé (2, 3, 6, 7),
- odvozené - vznikají zvětšováním a zmenšováním intervalů základních:
zvětšené (vzniknou z čistých a velkých)
zmenšené (vzniknou z čistých a malých)

1.5 Temperovaná soustava ladění

Temperovaná soustava ladění byla zavedena zejména kvůli vytvoření nástrojů, které mají neměnné ladění, tak aby bylo možné hrát skladby v libovolné stupnici. [5]

Od pythagorejské a přirozené se liší zejména tím, že definuje interval, jehož násobek určuje oktávu. Interval čisté oktávy (dvojnásobný kmitočet) je rozdělen na 12 stejných temperovaných půltónů tak, že pro jakékoliv dva sousední půltóny platí vztah

$$f_{n+1} = q f_n, \quad (1.6)$$

ve kterém je f_{n+1} je kmitočet nejbližšího vyššího půltónu k půltónu f_n a q je kmitočtový interval temperovaného půltónu. Pokud víme, že oktáva je dvojnásobný kmitočet a jednu oktávu dělíme na 12 půltónů, dostaneme vztah [6]

$$q = \sqrt[12]{2}. \quad (1.7)$$

Ladění této temperované stupnice je odvozeno od mezinárodní úmluvou stanoveného *komorního tónu*. Podle normy ISO R16 je to komorní tón a^1 s kmitočtem $f = 440$ Hz s maximální odchylkou $\pm 0,5$ Hz. V tab. 1.2 jsou uvedeny tóny a jejich kmitočty všech oktáv temperovaného ladění.

1.6 Akordy

Akord je definován jako „souzvuk nejméně tří různých tónů.“ U akordů rozeznáváme dva tvary:

- základní tvary,
- obraty.

Základní tvary akordů jsou v klasické hudbě složeny z tercií. Základním tvarem akordů je:

- Kvintakord - souzvuk tří tónů v terciových vzdálenostech, krajní tóny tvoří kvintu.
- Septakord - souzvuk čtyř tónů v terciových vzdálenostech, krajní tóny tvoří septimu.

- Nónový akord - souzvuk pěti tónů v terciových vzdálenostech, krajní tóny tvoří nónu.

Obraty vznikají přeskupením tónů v akordu, např. přenesením spodního tónu o oktavu nahoru nebo opačně. [7]

V momentě kdy tyto znalosti propojíme s tóninou, můžeme začít vytvářet akordy různých stupnic. Pro příklad uvedeme základní tóninu C dur, která obsahuje tóny *c, d, e, f, g, a, h, c*. Víme, že kvintakord je tvořen třemi tóny v terciových vzdálenostech. Podle tab. 1.1 zjistíme, že tercie je interval od I. ke III. tónu, v našem případě je první tón *c* a třetí tón *e*. Terciová vzdálenost od tónu *e* je tón *g*. Ze znalosti kvintakordu a tónů v tónině C dur můžeme vytvořit kvintakord C dur složený z tónů *c, e, g*. Septakord by pak přidal další tercii, tón *h*. Ze stupnice C dur můžeme odvodit, že durové stupnice jsou tvořeny tónovou posloupností

$$1 - 1 - \frac{1}{2} - 1 - 1 - 1 - \frac{1}{2},$$

tedy s půltónem mezi III. - IV. a VII. - VIII. stupněm. Se znalostí této posloupnosti můžeme jednoduše odvodit ostatní durové stupnice, označované velkým písmenem. (C, G, D, A, E, H, Fis a Cis dur.) [7] Pokud změníme tóninu např. na G dur, za dodržení této posloupnosti, musí mít tato tónina v předznamenání tón *fis*. Tóny této stupnice poté jsou *g, a, h, c, d, e, fis, g*. Kvintakord poté bude tvořen tóny *g, h, d*. [7, 2]

Pro naše účely a generování akordů nebudeme nutně potřebovat znalost všech tónin, nicméně si vystačíme vždy s jednou, ze které si dovedíme pomocí počtu půltónů od základního tónu různé typy akordů, které jsou uvedeny v tab. 1.3. Durový kvintakord jsme si již odvodili, v tónině C dur jej tvoří tóny *c, e, g*. Celá řada půltónů je uvedena v prvním sloupci tab. 1.2, jsou to tóny *c, cis, d, dis, e, f, fis, g, gis, a, ais, h, c*. Pokud si je očíslovujeme od tónu *c = 0*, budou tvořit akord C dur (*c, e, g*) půltóny 0 - 4 - 7. Pokud k tomuto kvintakordu přidáme ještě další tercii, vznikne nám septakord. Septakordů může být více typů. Pokud použijeme tercii malou (3 půltóny), (vzdálenost od základního tónu je malá septima) dostaneme dominantní septakord s tóny *c, e, g, ais*, tedy půltóny 0 - 4 - 7 - 10. Použitím velké tercie (4 půltóny) dostaneme z kvintakordu C dur velký septakord C dur s tóny *c, e, g, h* a půltóny 0 - 4 - 7 - 11. [7]

Pro příklad mollových stupnic použijeme stupnici a moll aiolskou, bez předznamenání, která obsahuje pouze tóny *a, h, c, d, e, f, g, a*. Posloupnost aiolské mollové stupnice je tedy

$$1 - \frac{1}{2} - 1 - 1 - \frac{1}{2} - 1 - 1,$$

a při vytváření kvintakordu zjistíme, že první tercie *a - c* obsahuje pouze $1\frac{1}{2}$ tónu, což jsou 3 půltóny a podle názvosloví jakosti intervalů hovoříme o tercii malé. Druhá

tercie od tónu c je tón e , což je tercie velká, 4 půltóny. Celý akord a moll poté obsahuje tóny a , c , e , zapsáno pomocí počtů půltónů 0 - 3 - 7. [7]

Pokud si dohromady spojíme, že v temperované soustavě ladění je mezi všemi sousedními půltóny přesně daný poměr q , podle rovnice (1.7), stačí nám znát pouze jakými půltóny se který akord tvoří a budeme schopni jej generovat na libovolném tónu neboli kmitočtu. Do teorie harmonie poté nemusíme dále zabíhat a pro generování akordů v libovolné stupnici nám bude stačit si stupnici zvolit a přiřadit počty půltónů neboli půltónovou masku akordu. Příklady těchto akordů jsou zobrazeny v tab. 1.3. [7]

Tab. 1.2: Kmitočty tónů temperovaného ladění [6, 2]

	Oktáva							
	sub- C ₂	kontra C ₁	velká C	malá c	jednoč. c ¹	dvouč. c ²	tříč. c ³	čtyřč. c ⁴
tón	f [Hz]							
c = his	16,35	32,70	65,41	130,81	261,62	523,25	1046,50	2093,00
cis = des	17,32	34,64	69,29	138,59	277,18	554,37	1108,73	2217,46
d	18,35	36,71	73,42	146,84	293,67	587,33	1174,66	2349,31
dis = es	19,45	38,89	77,78	155,56	311,13	622,25	1244,51	2489,01
e	20,60	41,20	82,40	164,81	329,63	659,25	1318,51	2637,02
f	21,83	43,65	87,32	174,61	349,23	698,46	1396,91	2793,82
fis = ges	23,13	46,52	92,50	185,00	369,99	739,98	1479,98	2959,95
g	24,50	49,00	98,00	196,00	392,00	783,99	1567,98	3135,95
gis = as	25,96	51,92	103,83	207,65	415,30	830,61	1661,22	3322,43
a	27,50	55,00	110,00	220,00	440,00	880,00	1760,00	3520,00
ais = hes	29,14	58,27	116,54	233,08	466,16	932,33	1864,65	3729,30
h = ces	30,87	61,74	123,47	246,94	493,88	987,77	1957,53	3951,06

Tab. 1.3: Odvozené počty půltónu pro vytvoření základních akordů. [7]

základ akordu	půltónová maska akordu	příklad značení	název
kvintakord dur	0 - 4 - 7	C	kvintakord C dur
kvintakord moll	0 - 3 - 7	Cmi	kvintakord c moll
kvintakord dur	0 - 4 - 7 - 10	C ⁷	dominantní septakord (přidána malá septima)
kvintakord dur	0 - 4 - 7 - 11	C ^{7maj}	velký septakord (přidána velká septima)
kvintakord moll	0 - 3 - 7 - 10	Cmi ⁷	malý mollový septakord
kvintakord moll	0 - 3 - 7 - 11	Cmi ^{7maj}	velký mollový septakord

2 VST - VIRTUAL STUDIO TECHNOLOGY

Tato technologie slouží jako rozhraní pro integraci softwarových hudebních nástrojů a efektů s programy na úpravu a nahrávání hudby. V dnešním světě je masivně podporována většinou profesionálních i amatérských programů na úpravu a vytváření hudby, kde se používají jako nástroje, nové efekty nebo simulace tradičního vybavení používaného v nahrávacím studiu a fungují na bázi Digital signal processing – číslicové zpracování signálu (dsp). Tato technologie se používá formou zásuvných modulů – plug-inů, které pracují v hostitelské aplikaci. Tou může být libovolná aplikace, která umí s technologií VST pracovat, nejčastěji jsou to ale tzv. Digital Audio Workstation – digitální hudební pracovní stanice (DAW). (Např. programy Cubase a Nuendo firmy Steinberg,¹ nebo Pro Tools firmy Avid.²) [15]

Technologie VST je určena pouze pro zpracování signálů uvnitř hostitelské aplikace a nestará se o vstupní a výstupní operace nebo o práci se zdrojem zvukových signálů, toto tedy musí zajistit hostitelská aplikace. Velmi často je tak spolu s technologií VST implementována v aplikacích i technologie Audio streaming input/output (ASIO), která se stará o vstup a výstup se zvukovým hardwarem. [10]

2.1 Implementace VST

Pro vývoj samotného VST plug-inu musíme pracovat s tzv. VST SDK (Software development kit – balíček pro vývoj programu), které je poskytnuté firmou Steinberg zdarma a jako open source (ve formě zdrojových kódů), stačí se na internetové stránce bezplatně zaregistrovat a souhlasit s licenčními podmínkami.³ Tento vývojový balíček je závislý na použité platformě a vývojovém prostředí, takže grafické funkce knihovny VST GUI (Graphic user interface - grafického uživatelského rozhraní) jsou vázány na použitý operační systém, datové typy (64 a 32 bit) a binární formáty knihoven jsou závislé na typu procesoru. Vývojové prostředí má vliv na konvence volání tzv. callback funkcí a vstupní body knihoven plug-in modulu. [10]

VST SDK mají více typů, v dnešní době je nejnovější verze 3.5.2, starší poté 2.4 nebo 2.3. Nejnovější verze ale přepracovala celou svou strukturu a její programový model je složitější, podpora v hostitelských, zejména freewarových, aplikacích není ještě implementována a proto budeme používat verzi 2.4. Tato verze již umožňuje rozlišení 64bit, také je zaručena zpětná kompatibilita, pokud je kód sepsán správně. [10, 16]

¹URL: <http://www.steinberg.net>

²URL: <http://www.avid.com/us/products/pro-tools-software>

³URL: <http://www.steinberg.net/en/company/developer.html>

I když je oficiálním vývojovým prostředím pro VST SDK verze 2.4 prostředí Microsoft Visual C++ 8 pro platformu Windows a Apple XCode 2.2 (a novější) na platformě Mac, jsou zdrojové kódy VST SDK upraveny tak, aby bylo možné podle definovaných symbolů vývojových prostředí, které se starají o podmíněnou definici datových typů, volacích konverzí a grafických funkcí, sestavit výsledný program s různými kompilátory. Pro platformu Windows jsou to kompilátory prostředí Microsoft Visual C++ a Borland C++ Builder, na platformě Mac OSX kompilátor prostředí Apple XCode a Metrowerks Code Warrior. [10, 16]

2.2 Hostitelské aplikace

Již vytvořený a hotový program by měl být kompatibilní se všemi aplikacemi podporujícími technologii VST, pro vývoj a testování aplikace je dobré zvolit program, který umožňuje program přeložit bez optimalizací a s možností krokování programu. Toto je ale většinou problém, protože aplikace jsou chráněny proti crackerům a spuštění v ladícím režimu vývojového prostředí nedovolují. [10]

Ne všechny aplikace také podporují zobrazení grafického rozhraní nebo opačně, pokud plug-in modul GUI nemá, neumožní uživateli změnit nastavení. Dobré je si především na začátku ujasnit, kolik a jakých vstupních a výstupních kanálů bude VST plug-in upravovat. Dalším důležitým parametrem je, zda-li je v aplikaci implementována podpora MIDI protokolu a také jakým způsobem (některé aplikace jsou schopné MIDI zprávy nejen přijímat a provádět ale také je rovnou vysílat), nebo jestli je možné výstup z aplikace nějak zaznamenávat. Neméně důležitým parametrem je i s jakými ovladači zvukové karty je program schopen pracovat.

2.3 Zvuková karta a ovladače ASIO

VST technologie, tedy Virtual Studio Technology, je i ve svém názvu primárně určena pro práci s hudbou způsobem, který měl napodobovat práci v hudebním studiu a byl zamýšlen pro práci se zvukovou kartou, která se postará o vstupní a výstupní operace se signálem. Všechny společnosti, které vytvářejí zvukové karty určené na práci s počítačovou hudbou dodávají ke svým výrobkům ASIO ovladače. Tyto zvukové karty mají velký cenový i kvalitativní rozptyl, podle počtu vstupních a výstupních kanálů, kvality převodníků a dalších funkcí se pohybují ceny nejlevnějších zařízení v řádů tisíců korun, nejdražší zařízení mohou stát desítky až stovky tisíc korun, pro jednoduchý vývoj aplikace se však tato investice dá obejít díky ovladači **asio4all**,⁴ který je zapouzdřením ovladače WDM v technologii ASIO. [10]

⁴Dostupné z URL: <http://asio4all.com/>

3 MIDI

MIDI (Musical Instrument Digital Interface - digitální rozhraní pro hudební nástroje) má několik významů, je zároveň normou, protokolem a hardwarovým rozhraním. Je to komunikační standard, který umožňuje společnou komunikaci hudebního nástroje a počítače. Vznikl v roce 1983 z protokolu UMI (Universal Musical Instrument – univerzální hudební nástroj) na kterém spolupracovala firma Roland Corp. a Sekvential Circuits, poté co se k vývoji přidali společnosti Yamaha a Oberheim. Postupně se tento protokol vyvíjel a v hudební praxi je stále více používán i přes svou značnou zastaralost a z toho plynoucí omezení. [3]

MIDI je jazyk, který popisuje hudbu v binární podobě a každé binární slovo představuje událost, která se stala v průběhu hudebního výkonu. Tento jazyk vznikl pro klávesové nástroje a proto mnoho prvků z těchto kořenů vychází, nejčastější zprávy se tedy týkají zejména toho, jestli hudební nástroj právě hraje nebo nehraje nějakou notu. [3]

3.1 MIDI zprávy

Obsah MIDI zprávy je jednoduše definovaný, jako první je vždy vyslán prvek **status byte** za kterým může následovat jeden nebo více **datových bajtů**. Jak je ukázáno na obr. 3.1, stavový bajt je od datového rozlišen pomocí nejvýznamnějšího bitu (MSB - most significant bit), pokud je tento bit nastaven na jedničku, jedná se o stavový bajt, pokud je nastaven na nulu, je to datový bajt. [3, 11]

Status byte

1	T	T	T	n	n	n	n
---	---	---	---	---	---	---	---

Data byte

0	v	v	v	v	v	v	v
---	---	---	---	---	---	---	---

identifikátory:

T - typ zprávy

n - MIDI kanál / typ systémových dat

v - hodnota dat

Obr. 3.1: Struktura MIDI zprávy, [11]

Další rozdělení je podle identifikátoru typu zprávy (viz orb. 3.1) na:

- Kanálové, přijímané pouze zařízeními, které jsou nastaveny na jeden ze 16 kanálů:

Channel Voice – základní zprávy, používají je pro popis hudební skladby, nejčastěji jsou to zprávy Note on a Note off.

Channel mode – popisují zařízení způsob, jak přijímat a posílat informace, např. vypnutí všech zvuků.

- Systémové, přijímané všemi zařízeními:

System Common – používají sekvencery¹ pro nastavení ladění, výběru a umístění skladby.

System Real-Time – synchronizační příkazy.

System Exclusive – informace o nastavení daného nástroje, právě používaný zvuk nebo program. [3, 11]

3.2 Využití

Pro tuto práci je nutné rozeznat pouze základní MIDI zprávy typu **Channel Voice**. Aby bylo možné implementovat MIDI protokol do VST pluginu, je potřeba zejména rozlišit přicházející data, kde se zaměřím jen na dvě nejdůležitější zprávy **Note On** a **Note Off**.

MIDI zpráva Note On

Při stisku klávesy na MIDI klaviatuře se pošle zpráva obsahující dvě informace: číslo zahrané noty a její sílu, tzv. velocity. Tyto informace se přenáší v datových bajtech, máme tedy 7 bitů pro data (viz obr. 3.1, což odpovídá rozmezím hodnot 0 až 127.) První datový bajt určuje výšku noty od hodnoty 0 - nota C₂ v sub-subkontra oktávě až po notu g⁶ v šestičárkované oktávě, které odpovídá hodnota 127. Druhý datový bajt přenáší informaci o síle zahrané noty - velocity. Pokud je hodnota velocity 0 tak zpráva dostává stejný význam jako MIDI zpráva note off. [3, 11]

MIDI zpráva Note Off

Tato zpráva se posílá vždy při uvolnění hrané klávesy. Většina zařízení ale používá zprávu Note On, jen s úrovní velocity 0, některé klaviatury však mají funkci Release Velocity Sensing, která sleduje rychlost uvolnění klávesy, což má vliv na doznívání zahrané noty, proto se využívá zpráva Note Off s nenulově nastavenou úrovní velocity. Ukázka posílaných zpráv je v tabulce 3.1, kde je vidět rozdíly posílaných dat pro zmíněné MIDI zprávy.[3, 11]

¹Zařízení pro záznam a přehrávání MIDI zpráv

Tab. 3.1: Popis MIDI zpráv Note On a Off [3]

Popis akce	stisknuta klávesa	nota c^1	úroveň velocity
MIDI zpráva	Note On, kanál 3	číslo noty 60	100
Binární podoba	1001 0010	0011 1100	0110 0100
Popis akce	klávesa puštěna	nota c^1	úroveň velocity
MIDI zpráva	Note Off, kanál 3	číslo noty 60	0
Binární podoba	1000 0010	0011 1100	0000 0000

4 VÝVOJ KÓDU

Pro řešení tohoto zadání byla zvolena šablona VST SDK verze 2.4 a vývojové prostředí Microsoft Visual C++ s verzí projektu 8. Tato šablona poskytuje pro lepší komfort kromě standardních konfigurací *Debug* a *Release* i konfiguraci *Debug with GTPlayer EDU*, která sestaví plug-in modul přímo v adresáři této hostitelské aplikace a po spuštění aplikace je možné plug-in rovnou načíst. [13]

Šablona již předem definuje velké množství parametrů, které mohou nebo v některých případech nemusí být použity. Definice základních parametrů plug-in modulu se nachází v souboru `vst_temp_defs.h`, (kromě údajů o verzi a identifikátoru lze nastavit i použití či nepoužití grafického editoru a jeho parametry, počet vstupních a výstupních kanálů a pod.) [13]

Soubor `vst_temp.cpp` je určen pro implementaci výsledného algoritmu a v této práci je celý obslužný kód implementován v tomto souboru. Nachází se zde jak definice jména programu, výrobce, tak i definice používaných konstant. Dále pole struktur `unsortedParams`, která definuje reálné hodnoty a vlastnosti chování algoritmu, kde je každá položka určena jedním záznamem obsahujícím pojmenováním, základní, maximální a minimální hodnotou, jménem a popisem parametru, typem zobrazení a parametrem pro nastavení pouze pro čtení. Zde tyto soubory a strukturu uvádím zejména proto, že budou klíčové pro popis chování programu v následující kapitole. [13]

4.1 Postup vývoje

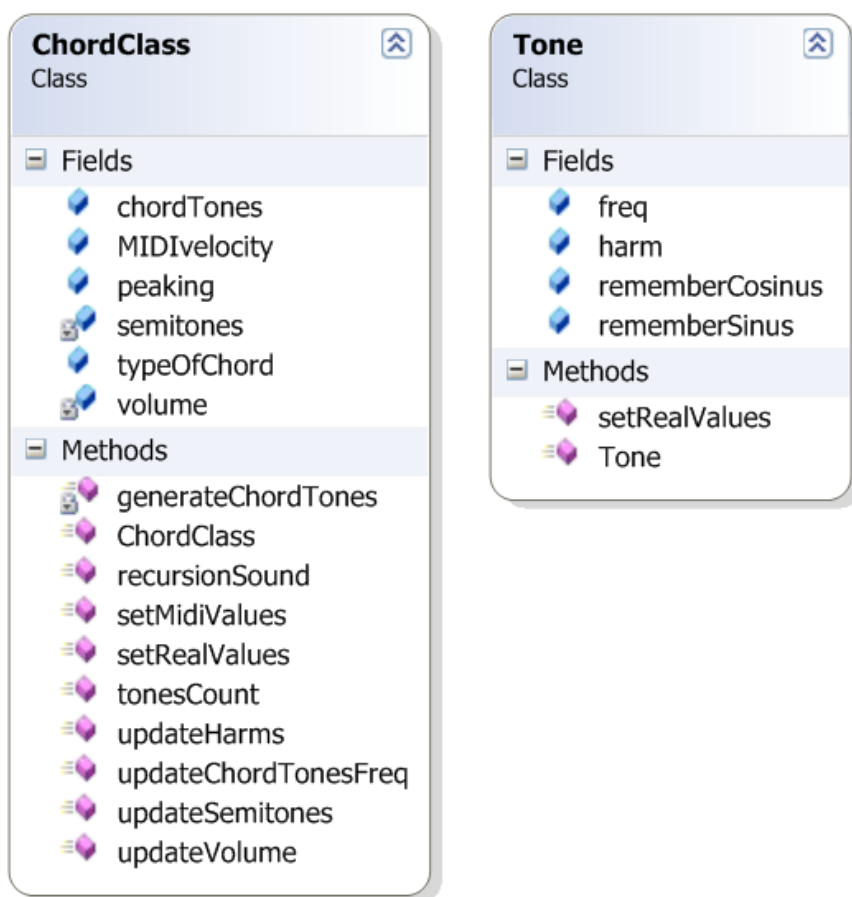
Jak je v kapitole 1.1 ze vztahu (1.2) vidět, tón je definován třemi parametry. Jsou to frekvence f , amplituda A_1 a fázový posun ϕ_1 . U jeho vyšších harmonických složek se podle vztahu (1.3) mění jen amplitudy A_n a fáze ϕ_n . Fázovým posunem se ale zabývat již nebudu, jak je uvedeno v kapitole 1.1, změna fáze není pro lidské ucho slyšitelná. V programu tedy budou stačit dvě proměnné, které budou definovat jeden tón. Je to proměnná *frekvence* a pole *amplitud*. Pro vyšší harmonické se nemusí použít pro ukládání frekvence pole, protože frekvence vyšších harmonických složek jsou jen násobky frekvence základní, jak je vidět ze vztahu (1.2). Třída představující *tón* o n -harmonických si vystačí se dvěma proměnnými: jednou *frekvencí* a *n-amplitudami*.

Akord je v kap. 1.6 definován jakou souzvuk tří a více tónů. V této kapitole je také řečeno, že pro vytvoření akordu libovolné tóniny stačí znát pouze tzv. půltónovou masku akordu, která jednoznačně definuje typ akordu. (Ukázky základních akordů jsou uvedeny v tab. 1.3.) V tomto kroku, tedy po zvolení tóniny a výběru typu

akordu, už nic nebrání v tom požadovaný akord vytvořit. Třída *akordu* si tedy vystačí s proměnnými *tóny* a *typem akordu*.

Výsledný diagram tříd je zobrazen na obr. 4.1, kde je vidět, že program obsahuje navíc další proměnné. Každý tón si uchovává kromě frekvence *freq* a síly harmonických složek *harm* i informace pro rekurzivní generátor, *rememberCosinus*, *rememberSinus*, který popíši později.

Třída *ChordClass* obsahuje pole tónů *chordTones*. Proměnná určující typ akordu, *typeOfChord*, je výčtového typu, který definuje vlastní názvy akordů, aby se v programu na tyto typy nemuselo odkazovat pomocí čísel, ale názvů. Půltónová maska akordu, podobná jako v tab. 1.3, je uložena v proměnné *semitones*. Proměnná *volume* slouží pro nastavení celkové hlasitosti. Poslední proměnnou je *MIDIvelocity*, která slouží k uchovávání aktuální hlasitosti hraného tónu, pokud využíváme tzv. MIDI mód.



Obr. 4.1: Diagram tříd VST plug-inu

4.2 Popis funkcí

Při spuštění aplikace se jako první musí provést základní nastavení hodnot jednoho *Akordu*. Neustále se vychází z myšlenky tvorby akordu ze základního tónu a akordové masky. Aby aplikace mohla v pořádku fungovat, potřebuje definovat základní hodnoty třídy akordu. Tato nastavovací funkce se provede pouze při spuštění aplikace, ale jsou díky ní nastaveny všechny parametry. Tento proces probíhá následovně:

1. frekvence základního tónu se spolu s hodnotami amplitud a fází všech tónů pošlou funkci, která se stará o prvotní nastavení proměnných v akordu.
2. Tato funkce nastaví hodnoty *frekvence* a *amplitudy* prvnímu tónu z *chordTones* a zavolá další funkci, která má na starost vygenerování *frekvencí* pro ostatní tóny v akordu.
3. Generování *frekvencí* pro ostatní tóny čerpá ze vztahu (1.6), vztah pro velikost frekvence vyššího tónu odvozeného pomocí půltónové masky je poté

$$f_n = f_1 q^C, \quad (4.1)$$

kde f_1 je *frekvence základního tónu*, $q = \sqrt[12]{2}$ a C je proměnná určující počet půltónů nutných k vytvoření správného tónu z půltónové masky. Po vypočítání této *frekvence* dosadí tato funkce i *amplitudy* pro ostatní tóny.

V tomto momentě je stanoven *typ akordu*, jeho *základní tón* i správně vypočítané *frekvence* ostatních tónů v akordu, stejně tak jako jejich *amplitudy*, takže lze přejít k hlavnímu bodu této práce, převedení těchto údajů na zvuk.

4.3 Generování zvuku

Jak jsme dříve zmínil v kapitole 2, VST plug-in se nestará o výstupní operace a pouze svým algoritmem upravuje, v tomto případě vytváří zvuk. Stačí tedy posílat data na výstup programu a o zpracování se postarají příslušné ovladače. Jako první jsem zkusil použít funkci, která vznikne drobnou úpravou vztahu (1.3),

$$f(t) = \sum_{n=1}^{\omega} X A_n \cos(n2\pi ft + \phi_n), \quad (4.2)$$

kde X je proměnná, která zajistí, aby nebylo možné přesáhnout maximální možnou velikost hlasitosti (hodnota 0 dBFS) a mění se s počtem tónů v akordu, což zaručí, že při generování různého počtu tónů v akordu budeme mít vždy stejnou výstupní hlasitost.

Tento algoritmus není dostatečně rychlý na to, aby zvládl generovat více tónů zároveň, podle použitého výpočetního zařízení bylo možné generovat 3 až 5 tónů, ale při velmi vysokém vytížení procesoru. Hlavním problémem bylo vysoké množství

násobení a výpočetně náročná funkce $\cos(t)$. Proto jsem zkusil použít rekursivního generování, které obsahuje podstatně méně operací. Algoritmus 1 vychází z rovnice, která je převzata ze zdroje [14]

$$\cos'(w \times x) = -w \sin(w \times x) \quad (4.3)$$

$$\sin'(w \times x) = w \cos(w \times x), \quad (4.4)$$

která je velmi jednoduchá a využívá k výpočtu derivaci funkcí $\sin(t)$ a $\cos(t)$. Tento algoritmus zůstává stabilní i ve chvíli, kdy měníme velikost proměnných, je rychlý a největší výhodou je ta, že počáteční hodnota proměnné *cosinus* se nemusí rovnat 1, ale může být libovolně velká, generované hodnoty poté budou v rozsahu \pm zvoleného čísla, např. od hodnoty -100 do $+100$ při zvolení *cosinus* = 100.

Algoritmus 1: Algoritmus rekursivního generování

```

1: Definice proměnných
2: length = velikost výstupního pole
3: sinus = 0
4: cosinus = 1
5: frekvence =  $2.0 * \pi / \textit{length}$ 
6: Rekursivní výpočet
7: for i = 1 to length - 1 do
8:   cosinus - = sinus * frekvence
9:   sinus + = cosinus * frekvence
10: end for

```

Finální algoritmus se ovšem komplikuje, neboť je takto potřeba generovat hodnoty pro všechny tóny a jejich vyšší harmonické složky. Je potřeba si v paměti uchovávat všechny finální stavy proměnných *sinus* a *cosinus* pro každou harmonickou složku všech tónů, aby bylo možné v generaci pokračovat od stejné hodnoty a nevznikaly tak chyby.¹ Úroveň zkreslení rekursivního generátoru uvedu v kapitole 5. Rozdíl těchto dvou typů generace je ale veliký, na počítači se stejnou hardwarovou a softwarovou konfigurací bylo vytížení procesoru původně přes 86 % při generování pěti tónů, po implementaci nového algoritmu kleslo vytížení pod 10 %, což dovoluje razantně snížit velikost výstupní vyrovnávací paměti ASIO ovladače a odezva pro hráče je tak o mnoho lepší.

¹Kompletní zdrojový kód pro nahlédnutí je přiložen jako příloha na CD.

4.4 Volba parametrů

Počet tónů byl zvolen na maximální hodnotu 5, tedy základní tón a 4 další tóny. Toto dostačuje na generování klasických i atypických akordů, tento počet tónů platí také pro MIDI režim.

Počet alikvot byl omezen na hodnotu 5, tedy 5 harmonických složek. Klasické elektronické hudební nástroje generují 8 až 12 harmonických složek, pro názornou demonstraci vlivu alikvot na barvu zvuku bude ale tato hodnota dostatečná. [9]

Frekvence základního tónu je možné měnit od hodnoty 130,5 Hz po hodnotu 1048 Hz, což je podle tab. 1.2 zhruba rozsah od tónu c až po tón c³.

Amplituda každé složky je nastavitelná v rozmezí 0 až 100 %, kdy hodnota 100 % odpovídá hodnotě těsně pod 0 dBFS, tak aby nedošlo k přebuzení.

Celkové zesílení tónů se nastavuje pomocí logaritmické stupnice a je cejchováno v dB, jeho rozmezí je od -60 dB do 0 dB.

4.5 Spuštění a ovládání aplikace

Konečný program je sestaven jako soubor dynamické knihovny .dll a je nutné jej spustit v hostitelské aplikaci. V následující podkapitole 4.5.1 je popsán správný postup spuštění, aby nedošlo k problémům s hostitelskou aplikací, špatné inicializaci zásuvného VST modulu nebo nefunkčnosti externí MIDI klaviatury.

4.5.1 Spuštění a základní nastavení

Některé hostitelské aplikace mají složku pro nalezení VST plug-in modulů definovanou nebo je nutné umístit knihovnu přímo do složky programu.² Toto je nutné provést ještě před spuštěním programu. Po spuštění si program najde vložené VST plug-in moduly a my s nimi můžeme pracovat. [10]

Následný postup spuštění a ovládání programu bude popsán pro program Tobybear MiniHost, který je volně dostupný na oficiálních stránkách.³

- Po spuštění aplikace **MiniHost.exe** je nejprve nutné správně nastavit ASIO ovladač. Je potřeba otevřít menu „ASIO“, dále označit položku „Driver“ a vybrat vhodný ASIO ovladač. (Což může být např. *ASIO DirectX Full Duplex* nebo *asio4all*.) Další položka v menu „ASIO“ je „Show Control Panel“, která

²Program Cubase má ve své základní složce podsložku s názvem *VSTPlugins*, GT Player zase podsložku s názvem *Plugins*

³MiniHost stahujte z URL: http://www.kvraudio.com/product/minihost_by_tobybear

zobrazí panel pro nastavení ASIO ovladače. Zde lze nastavit velikost vyrovnávací paměti neboli dopravní zpoždění – latency, většinou pomocí počtu vzorků (velikosti vyrovnávací paměti) nebo času. Důležité je si uvědomit, že malá velikost vyrovnávací paměti – bufferu, může způsobovat výpadky přehrávaného zvuku, pokud by počítač nestihl provádět výpočty dostatečně rychle, opačně velmi vysoké hodnoty zvyšují i dopravní zpoždění, což je nepřijatelné pro pohodlnou interakci hráče s klaviaturou. Finální nastavení bude záležet na výpočetním výkonu počítače a typu použitého ovladače.

- Pro využití externí MIDI klaviatury, je potřeba ji připojit k počítači již před spuštěním programu a poté v menu „MIDI“ → „Configure MIDI Ports“ vybrat správné „MIDI“ in zařízení.
- Po nastavení ovladače se musí určit cesta k souboru samotného plug-in modulu v menu „VST“ → „Load VST Plugin“. Otevře se klasický prohlížeč souborů, ve kterém se vybere požadovaná knihovna .dll VST zásuvného modulu.
- Objeví se obrazovka GUI pro ovládání parametrů programu, jejíž podoba je znázorněna na obr. 4.2, ve které se nachází sedm táhel, které ovládají parametry programu, tři tlačítka a dvě kontextové menu.

Bypass – slouží k vyřazení programu ze signálové cesty.

Note – slouží k nastavení frekvence základního tónu.

Harm – nastavuje amplitudu (hlasitost) jednotlivých tónů v akordu a jejich vyšších harmonických složek.

Volume – nastavuje zesílení všech tónů, při překročení maximální výstupní úrovně se automaticky stáhne tak, aby nedošlo k přebuzení.

Peaking – po zapnutí přestane hlídat velikost výstupní úrovně.

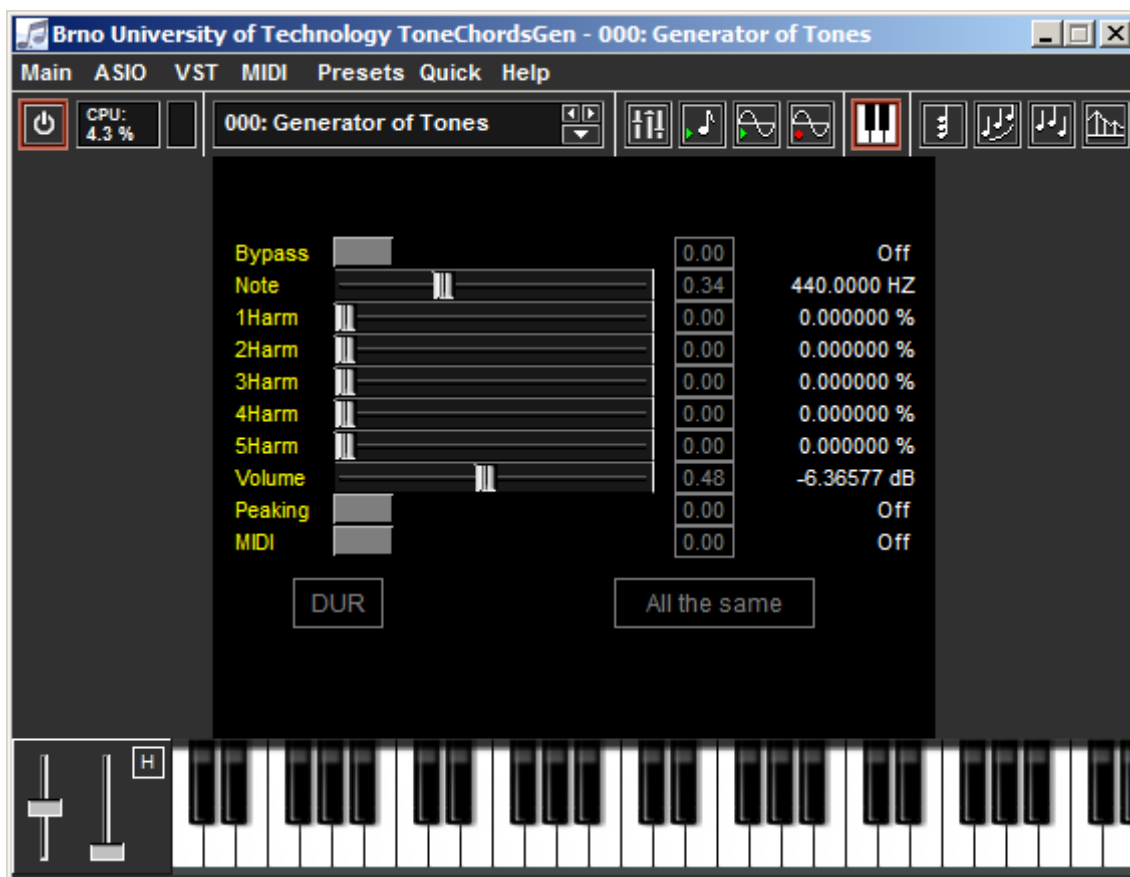
MIDI – přepíná mezi módem pro generaci akordů a módem generování podle hraných tónů na MIDI klaviatuře.

Akordy – vlevo dole - umožňují provést přehledný výběr pro zvolení typu generovaného akordu. (V MIDI nemá toto menu žádný vliv). Je možno vybírat z akordů typu: DUR, MOLL, TONE, DUR7 a DUR12.

Výběr tónů - vpravo dole - umožňuje vybrat si, který tón ovládají táhla Harm. V MIDI módu se tóny generují podle prvního tónu, jakýkoliv jiný výběr než „All the same“ nebo „First“ nebudou mít vliv na změnu hlasitosti jednotlivých harmonických složek.

4.5.2 Ovládání aplikace

Plug-in je po spuštění nastaven do základního módu generování akordů se základním tónem a^1 (440Hz) a typem akordu DUR. Výšku hraného tónu lze měnit táhlem Note, případně je možné ji přesně nastavit zahráním jedné libovolné noty na kla-



Obr. 4.2: Grafické uživatelské rozhraní zásuvného modulu v hostitelské aplikaci MiniHost

viatuře. Pro generování slyšitelného zvuku je potřeba posunout táhla jednotlivých harmonických složek z pozice 0 % více doprava. Zároveň se v tomto případě budou měnit amplitudy všech tónů současně, pokud se v kontextovém menu vpravo dole nepřepneme do jiné nabídky, např. „First“. Táhla se při přepínání mezi jednotlivými tóny překreslí vždy do aktuální pozice. Ve chvíli opětovného výběru první položky „All the same“ si všechny tóny nastaví amplitudy podle prvního tónu.

Úroveň hlasitosti jednotlivých tónů se při přepínání mezi typy akordů nemění. Každý tón, pokud je nastaveno „All the same,“ má stejnou hlasitost, pokud se např. přepne z akordu DUR na typ akordu DUR7 je slyšet přidání dalšího tónu a předchozí tóny hrají pořád stejně hlasitě. Pokud by ale jednotlivé táhla harmonických složek a hlasitosti byly nastaveny tak, že by došlo k přebuzení výstupu (absolutní hodnota větší než 1), automaticky se táhlo Volume stáhne o 0.02, toto stahování se opakuje do doby, dokud na výstupu nevzniká přebuzení. Kontrola přebuzení se provádí jednou za generující cyklus, což je při předpokladu nepříliš velké vyrovnávací paměti dostatečné a tento kontrolní mechanismus tak reaguje i na rychlé změny hlasitostí jednotlivých harmonických složek. Pro možnosti ukázat, jak nelineární zkreslení zní

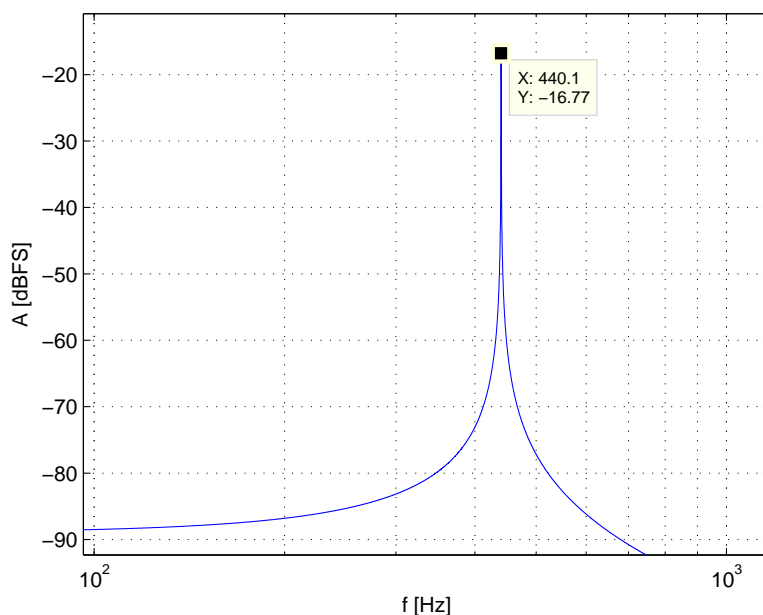
je v programu tlačítko Peaking, které vypíná kontrolu překročení maximální hodnoty výstupu.

Po přepnutí do MIDI módu bude situace podobná. Zde lze zároveň hrát pět tónů, ale jejich celková hlasitost je dále upravována kromě pozicí táhel také silou úderu (velocity). Všechny hrané tóny mají poměry amplitud nastaveny stejně, rozdíly v hlasitosti jednotlivých tónů jsou způsobeny rychlostí úderu na klávesu. Pokud bude zahráno více jak pět tónů současně, zahrají se ty, které byli stlačeny nejdříve, ostatní zahrány nebudou. Pro zahrání tohoto tónu je první potřeba některé tóny pustit a vybraný tón zmáčknout znovu.

5 ANALÝZA GENEROVANÉHO SIGNÁLU

Výsledek této bakalářské je funkční VST plug-in modul, specifikovaný jako VSTi – Virtual Studio Technology instrument, takže hostitelské aplikace jej primárně vnímají jako hudební nástroj. Modul má bezproblémové chování v různých hostitelských aplikacích, testováno v programech GT Player EDU, MiniHost, Cubase 5.¹ Je schopen plynule měnit amplitudy jednotlivých harmonických složek za běhu programu a generovat zvuk až pro 5 tónů v akordu současně.

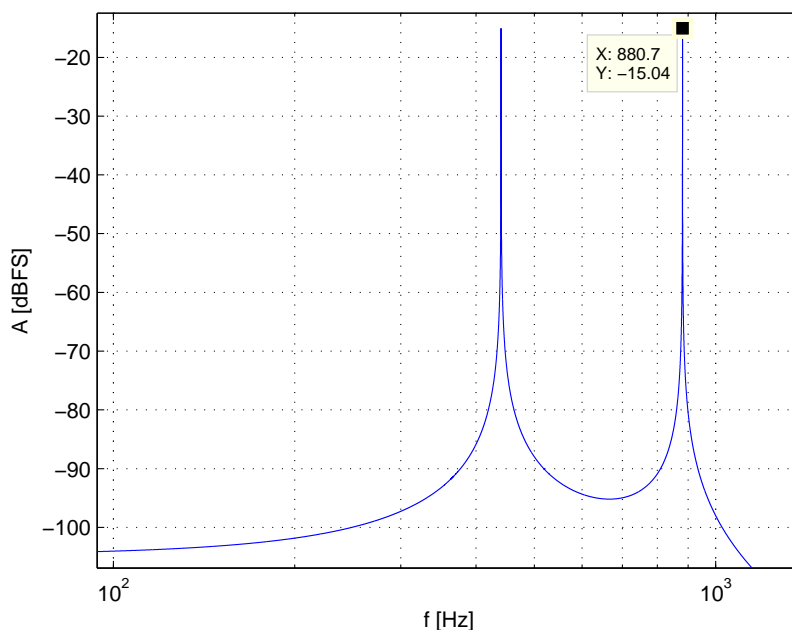
Na prvním obr. 5.1 z této série je vidět modulové kmitočtové spektrum signálu VST plug-inu první harmonické složky tónu *komorní a* s frekvencí 440 Hz, získaného rychlou furierovou transformací. Jak je vidět i z odečtu grafu, základní tón nemá přesnou hodnotu 440 Hz, přesná hodnota generovaného tónu je 440,14195 Hz, což je ale v přípustné odchylce $\pm 0,5$ Hz, jak uvádí norma ISO R16. (Viz kap. 1.5.) Tato chyba je zřejmě způsobena tím, že v rekurzivním generátoru dochází k násobení frekvence v každém kroku a tím, že používaný datový typ je `float`, který při tomto typu generování signálů způsobí nepřesnosti, které se zvětšují, pokud se s výsledkem dále pracuje. [8]



Obr. 5.1: Modulové kmitočtové spektrum signálu obsahující pouze první harmonickou složku o frekvenci 440 Hz.

¹GT Player, zkušební demo verze je dostupná z: http://www.stahuj.centrum.cz/multimedia/mp3_a_audio/tvorba/dsound-gt-player/, MiniHost zdarma dostupný z: http://www.tobybear.de/p_minihost.html, Cubase 5 je placený program společnosti Steinberg: <http://www.steinberg.net/en/products/cubase/start.html>

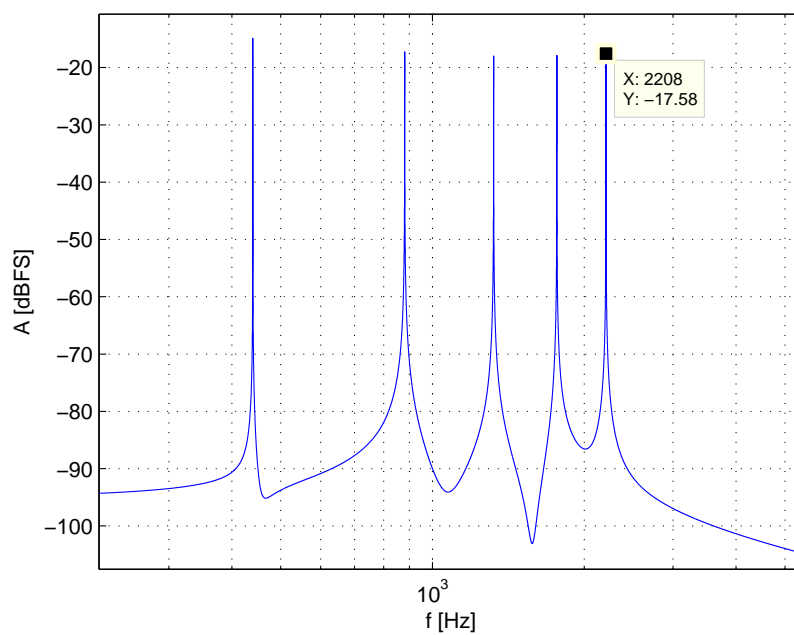
Obrázek 5.2 ukazuje situaci kdy se k první harmonické přidá druhá. Druhá harmonická složka by měla být dvojnásobkem první harmonické složky, což potvrzuje i tento graf.



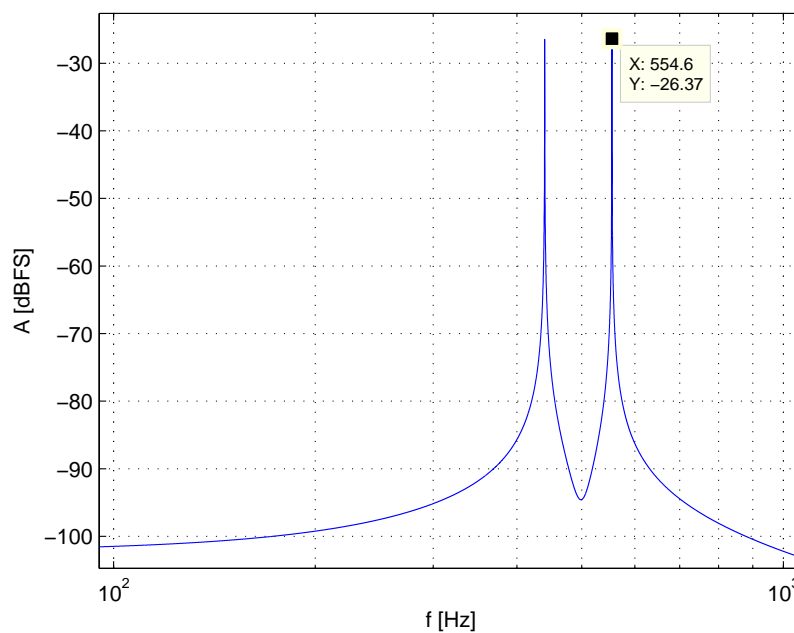
Obr. 5.2: Modulové kmitočtové spektrum signálu základního tónu s frekvencí 440 Hz a jeho druhé harmonické složky.

Další obrázek 5.3 zobrazuje obdobnou situaci kdy byly do spektra signálu přidány další harmonické složky. Celkový počet je tedy 5 harmonických složek. Zde je také vidět posunutí frekvencí vyšších harmonických složek, pro pátou je výsledná frekvence $f = 2208$ Hz, místo $f = 440 \times 5 = 2200$ Hz. Tyto chyby výpočtu jsou u páté harmonické složky slyšet a akord se zdá být lehce rozladěný, první čtyři harmonické složky jsou pro posluchače ještě v pořádku.

Ověřit správné generování tónů akordu z půltónové masky pomůže obrázek 5.4. Při zvolení akordu DUR a tóniny a^1 jsou tóny akordu a^1 , cis^2 , e^2 . Podle tabulky 1.2 má druhý tón akordu, cis, frekvenci 554,37 Hz, jak je ve zmíněném obrázku vidět, druhý tón je generován se správným kmitočtem.

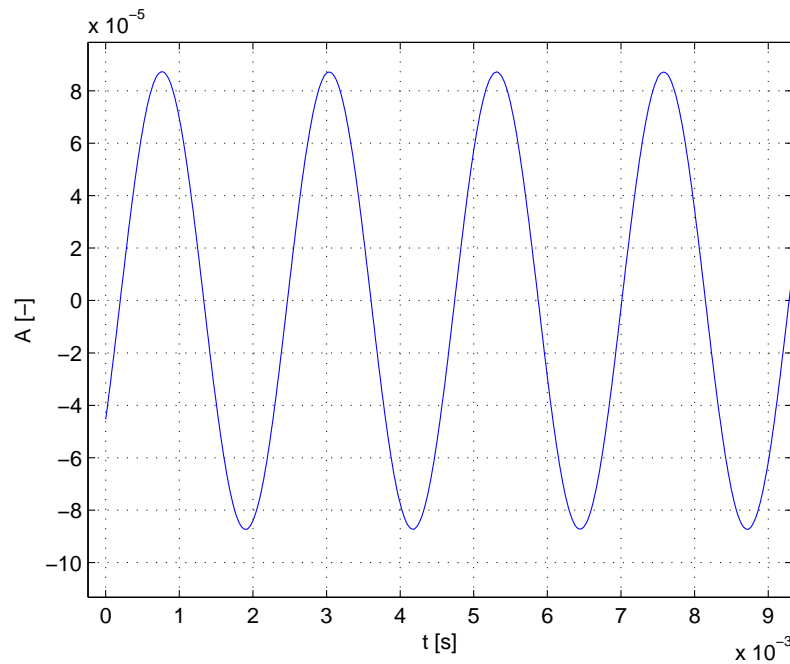


Obr. 5.3: Modulové kmitočtové spektrum signálu základního tónu s frekvencí 440 Hz a jeho pěti harmonických složek.



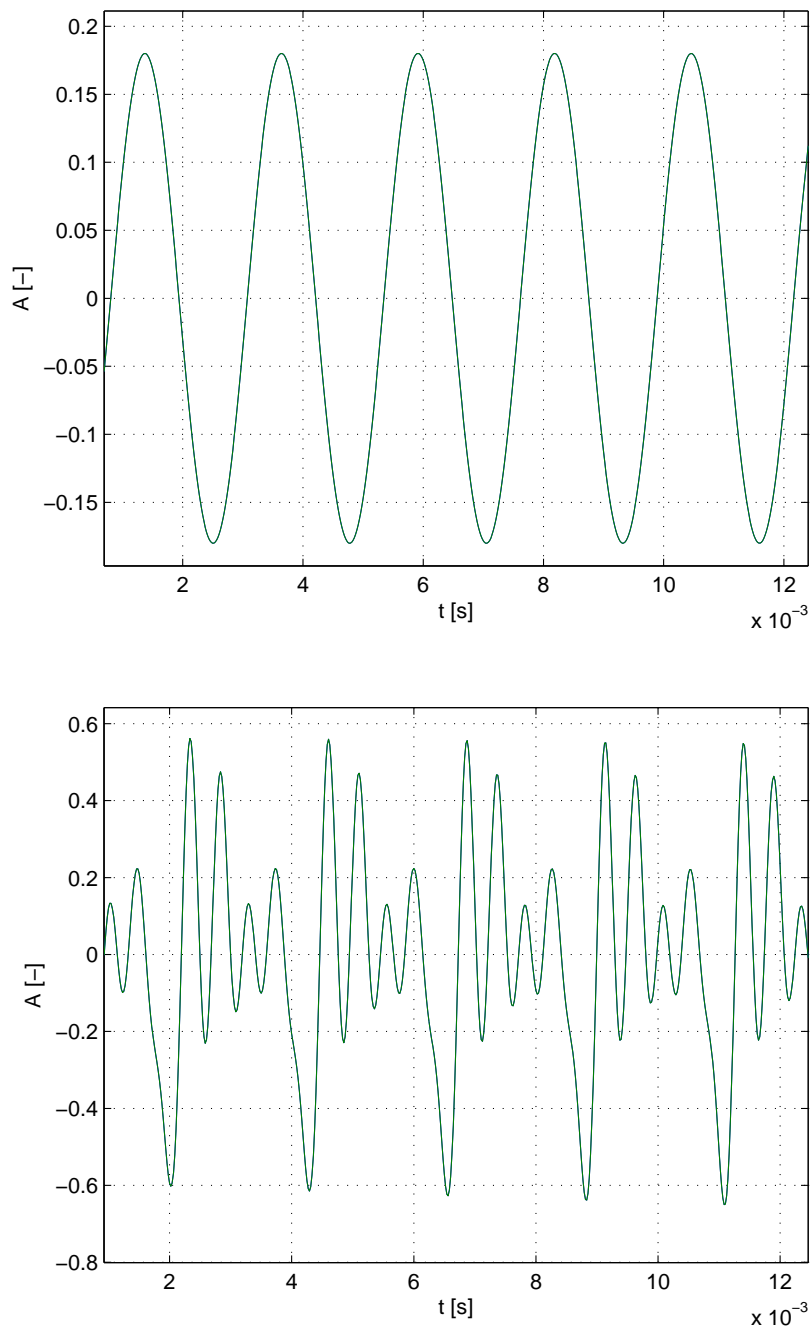
Obr. 5.4: Modulové kmitočtové spektrum akordu a-DUR, signál obsahuje pouze první harmonické složky prvních dvou tónů.

Přesnost použitého algoritmu pro rekurzivní generování jsem otestoval tak, že jsem si v hostitelské aplikaci nahrál krátký kousek generovaného zvuku o frekvenci 440 Hz, pouze jeho první harmonickou složku, se vzorkovací frekvencí 48000 Hz a bitovou hloubkou 24 bit. Dále jsem si v prostředí programu Matlab vygeneroval signál o stejné amplitudě, frekvenci a vzorkovací frekvenci. Signály jsem fázově posunul tak, aby oba začínali co nejbližší nule a pokračovaly do kladných hodnot a následně jsem tyto signály od sebe odečetl. Výsledek odečtení je ke shlédnutí na obrázku 5.5. Chyba tohoto algoritmu při generování jednoho tónu je v řádu 10^{-5} .



Obr. 5.5: Časový průběh rozdílu generovaného a referenčního signálu o frekvenci $f = 440,14195$ Hz a amplitudě $A = 0,18$.

Poslední obrázek č. 5.6 zobrazuje odlišnost časových průběhů, pokud je v signálu obsažena pouze první harmonická nebo všech pět harmonických složek pro signál se základním kmitočtem 440 Hz.



Obr. 5.6: Časové průběhy pouze první harmonické a pěti harmonických složek signálu o frekvenci první harmonické složky $f = 440$ Hz.

6 ZÁVĚR

V této bakalářské práci jsem vytvořil pomocí technologie VST funkční generátor zvukového signálu, typu VST instrument, který je hostitelskou aplikací rozeznán jako softwarový hudební nástroj, má vlastní grafické uživatelské rozhraní a je možné jej ovládat pomocí tohoto rozhraní a připojené MIDI klaviatury. Program dokáže vytvořit zvuk až pěti tónů a to dvojnásobným způsobem

- a) generuje souzvuk více tónů, akord, v závislosti na vybraném typu akordu a frekvenci (výšce), základního tónu (tóniny),
- b) generuje až pět různých tónů podle stisknutých kláves MIDI klaviatury, pokud je přepnut do MIDI režimu.

Hlasitost jednotlivých harmonických složek lze libovolně nastavovat a v průběhu generování plynule měnit, stejně tak jde plynule měnit i frekvence tónů. Přesné určení základního tónu, tak aby hudebně ladil, lze provést pomocí stisku klávek MIDI klaviatury (nebo vysláním MIDI příkazu z hostitelské aplikace).

Tento program by měl sloužit pro prohloubení znalostí v oboru hudební akustiky studentů hudebních oborů. Snaha bylo o co nejlepší spojení teorie změny barvy zvuku pomocí vyšších harmonických složek se sluchovým vjemem.

Programová část kódu vycházela z upravené šablony pro vývoj aplikací typu VST od vedoucího této práce, vlastní práce je vytvořena úpravou souboru `editor.cpp` a `vst_temp.cpp` a jejich hlavičkových souborů. Úpravy v prvním zmíněném souboru měly vliv na grafické rozhraní aplikace, ve druhém souboru jsou vytvořeny třídy, které provádí veškeré hlavní úpravy a obsahují rekurzivní generátor. Kromě vytvoření tříd jsem v souboru `vst_temp.cpp` upravoval funkce pracující s MIDI daty a funkce pro obsluhu změny v grafickém rozhraní.

Podařilo se mi snížit počet ovládacích prvků na nejnutnější minimum, takže aplikace je příjemnější na ovládání. Změna generujícího algoritmu ušetřila výkon procesoru, neměl by být problém tento zásuvný modul použít i na starších stolních počítačích a v libovolné hostitelské aplikaci.

Pro další vývoj aplikace by bylo dobré vyřešit problém s násobením čísel v datovém typu `float`, který způsobuje problémy s posunem frekvence vyšších harmonických složek. Poslední drobnost by mohl být zásah grafika do podoby a rozložení prvků grafického uživatelského prostředí, pokud by to uživatelé vyžadovali.

LITERATURA

- [1] BACHMANN, C. BISCHOFF, H. BRÖER, M. PFEIFER, S. *Cubase 5: Operation Manual*. Steinberg Media Technologies, 2009. 641 s.
- [2] GEIST, B. *Akustika: Jevy a souvislosti v hudební teorii a praxi*. Muzikus Praha, 2005. 281 s. ISBN 80-86253-31-7
- [3] GUÉRIN, R. *Velká kniha MIDI: standardy, hardware, software*. Vyd. 1. Brno: Computer Press, 2004, 340 s. ISBN 80-722-6985-2.
- [4] HELMHOLTZ, H. *On the sensation of tone*. 2. anglické vydání. Dover publications New York, 1954. ISBN 0-486-60753-4
- [5] KÁŇA, L. *Elektroakustika*. 1. vydání. VUT Brno, 2002. 91 s.
- [6] KÁŇA, L. SCHIMMEL, J. *Studiová a hudební elektronika* VUT Brno, 2002. 150 s.
- [7] KOFRONĚ, J. *Učebnice harmonie*. 6. vydání. Supraphon Praha, 1981. 195 s.
- [8] *Reprezentace dat: přednáška* INP 2012, FIT VUT v Brně.
- [9] SCHIMMEL, J. *Elektronické hudební nástroje: přednáška*. 2012.
- [10] SCHIMMEL, J. *Implementace algoritmů číslicového zpracování signálů pomocí technologie VST*. 2009.
- [11] SCHIMMEL, J. *Komunikační rozhraní MIDI: přednáška*. 2012.
- [12] SCHIMMEL, J. *Studiová a hudební elektronika*. 1. vydání. VUT Brno, 2012. 158 s. ISBN 978-80-214-4452-2.
- [13] SCHIMMEL, J. *Šablona VST plugin modulu*. verze 1.2, VUT Brno, 2008.
- [14] *Sine Generation Tutorial* [online]. hugi.scene.org, Vydání 3.1, [cit. 2013-05-08]. Dostupné z <http://www.hugi.scene.org/online/coding/hugi%2016%20-%20cosine.htm>
- [15] *Steinberg Cubase VST* [online]. Sound on Sound, 1996 [cit. 2012-12-06]. Dostupné z: http://www.soundonsound.com/sos/1996_articles/jul96/steinbergcubase3.html
- [16] *VST Plug-Ins SDK Documentation* [online]. Steinberg Media Technologies, 2003. [cit. 2013-05-08]. Dostupné z <http://www.gersic.com/vstsdk/>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ASIO Audio streaming input/output

ČSN česká státní norma

DAW Digital Audio Workstation – digitální hudební pracovní stanice

dB Decibel – podílová veličina

dBFS Decibel Full Scale – podílová veličina vztažená k hodnotě 1

dsp Digital signal processing – číslicové zpracování signálu

DSP Digital signal processor – digitální procesor na zpracování signálu

GUI Graphic user interface – grafické uživatelské rozhraní

Hz Hertz – veličina kmitočtu

ISO International Organization for Standardization – mezinárodní organizace zabývající se tvorbou norem

MIDI Musical Instrument Digital Interface

SDK Software development kit – integrované vývojové prostředí

UMI Universal Musical Instrument – univerzální hudební nástroj

USB Universal Serial Bus

VST Virtual Studio Technology – technologie zpracování signálů firmy Steinberg

WDM Windows Driver Model

A OBSAH PŘILOŽENÉHO CD

ToneGenerator.dll - výsledný VSTi zásuvný modul.

Generátor - adresář obsahující všechny zdrojové kódy.

Šablona - adresář šablony VST plug-in modulu, ze které tato práce vycházela.

text_práce.pdf - text bakalářské práce