



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

SOFTWAREVÝ HLUKOMĚŘ

SOFTWARE SOUND LEVEL METER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Milan Kotalík

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Petr Sysel, Ph.D.

BRNO 2020



Bakalářská práce

bakalářský studijní program **Audio inženýrství**
specializace Zvuková produkce a nahrávání
Ústav telekomunikací

Student: Milan Kotalík

ID: 195767

Ročník: 3

Akademický rok: 2019/20

NÁZEV TÉMATU:

Softwarový hlukoměr

POKYNY PRO VYPRACOVÁNÍ:

V prostředí Matlab vytvořte funkce pro výpočet hladiny akustického tlaku při použití kmitočtových váhovacích filtrů A, C a Z a integračního článku s časovou konstantou Fast a Slow podle doporučení IEC 61672. Vstupem funkce bude zvukový soubor wav a převodní konstanta udávající poměr hladiny akustického tlaku v jednotkách dB(SPL) a úrovně digitálního zvukového souboru v jednotkách dBFS. Dále reallizujte funkce pro výpočet hlukové expozice a intenzity zvuku pomocí dvoukanálové p-p sondy. Po úspěšném otestování funkce implementujte v jazyce C, C++ nebo C# a realizujte aplikaci, která bude v reálném čase provádět měření hladiny akustického hluku, hlukové expozice a intenzity zvuku.

DOPORUČENÁ LITERATURA:

- [1] Electroacoustics - Sound level meters: Part 1: Specifications. International Electrotechnical Commission, 2013.
- [2] KUO, S., M.; LEE, B., H.; TIAN, W. Real-time digital signal processing: implementations and applications. 2nd ed. Hoboken, NJ: John Wiley, 2006. ISBN 0-470-01495-4.

Termín zadání: 3.2.2020

Termín odevzdání: 9.6.2020

Vedoucí práce: doc. Ing. Petr Sysel, Ph.D.

doc. Ing. Jiří Schimmel, Ph.D.
předseda rady studijního programu

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

V této závěrečné práci se zabírám problematikou měření hluků. Popisuji lidské vnímání zvuku a vysvětluji základní parametry a veličiny zvuku podle normy ČSN EN 61672-1. Dále se zabírám problematikou měření zvukových veličin v reálném čase. Vytvářím ucelený soubor funkcí pro měření hladin zvuku v prostředí MATLAB, které se pokouším implementovat v jazyce C++ pomocí jeho JUCE framework.

Klíčová slova

Hlukoměr, kalibrace, hladina akustického tlaku, MATLAB, C++, JUCE framework

Abstract

In this thesis I focus on noise measuring. I describe how humans perceive sound and explain basic acoustic parameters and quantity according to norm ČSN EN 61672-1. Moreover, I focus on the issue of acoustic quantity measuring in real time. I create comprehensive group of functions in MATLAB environment, which I try to implement with C++ and JUCE framework.

Keywords

Sound level meter, calibration, sound pressure level, MATLAB, C++, JUCE framework

Bibliografická citace:

KOTALÍK, Milan. *Softwarový hlučkoměr* [online]. Brno, 2020 [cit. 2020-06-09]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/122877>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Petr Sysel.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma Softwarový hlukoměr 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 jsem si 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.

V Brně dne:

.....

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce doc. Ing. Petru Syslovi, Ph.D. Za vedení mé práce a mnoho užitečných rad.

V Brně dne:

.....

podpis autora

Obsah

1.	Úvod.....	13
2.	Zvuk	14
2.1	Šíření a rychlost zvuku.....	15
2.2	Zdroj Zvuku	15
2.3	Vnímání zvuku	15
2.4	Sluch.....	15
2.4.1	Vnější ucho	16
2.4.2	Střední Ucho	16
2.4.3	Vnitřní Ucho	17
2.5	Hluk.....	18
2.5.1	Poškození Sluchu.....	18
2.6	Akustický tlak	19
2.7	Akustický výkon	19
2.8	Referenční akustický tlak.....	19
3.	Měření hluku	20
3.1	Stupnice pro měření hluku	20
3.1.1	Číslo třídy hluku	20
3.2	Hladina Akustického tlaku.....	20
3.2.1	Vážení v kmitočtu.....	20
3.2.2	Vážení v čase	21
3.2.3	Časově vážená hladina akustického tlaku.....	21
3.2.4	Maximální časově vážená hladina akustického tlaku.....	21
3.2.5	Špičkový akustický tlak.....	22
3.2.6	Hladina špičkového akustického tlaku	22
3.2.7	Časově průměrovaná hladina akustického tlaku.....	22
3.3	Zvuková expozice	22
3.3.1	Hladina zvukové expozice	23
3.4	Vážení v kmitočtu	23
3.4.1	Váhové filtry	23
3.4.2	Číslicové filtry	25
3.4.2.1	FIR.....	25

3.4.2.2	IIR.....	26
3.4.2.3	Transformace analogových filtrů	26
3.5	Intenzita zvuku	27
3.6	Měřicí zařízení	28
3.6.1	Mikrofony	28
3.6.1.1	Citlivost	29
3.6.1.2	Mezní akustický tlak.....	29
3.6.1.3	Směrová charakteristika	29
3.6.1.4	Frekvenční charakteristika.....	30
3.6.2	Hlukoměry	30
3.6.2.1	Kalibrace hlukoměrů	31
3.6.2.2	Požadavky na prostředí.....	31
4.	zpracování audio signálu v reálném čase	32
4.1	Vyrovňovací paměť	32
4.1.1	Ukazatele	32
4.1.2	Abstraktní datový typ list.....	32
4.2	JUCE framework.....	33
5.	Realizace v prostředí matlab	34
5.1	Skript.....	34
5.1.1	Volba měření.....	34
5.1.2	Volba časové konstanty	35
5.1.3	Volba intervalu měření	35
5.1.4	Zadání hustoty vzduchu a distance mikrofonů	35
5.1.5	Načtení souborů	35
5.1.6	Volba vážení	35
5.1.7	Aplikace vážení.....	36
5.1.8	Tisk výsledku.....	36
5.2	Po.....	36
5.3	Funkce nacteni	36
5.4	Funkce FiltrA a FiltrC	38

5.4.1	Kmitočtové odezvy jednotlivých filtrů	41
5.5	Funkce HAT	43
5.6	Funkce HZE	44
5.7	Funkce IZ	45
5.8	Aplikace měření	45
6.	Implementace v c++	47
6.1	Třída MenuComponent	47
6.2	Main.cpp.....	48
6.3	Třída MainComponent	48
6.3.1	Důležité nastavení v konstruktoru	49
6.3.2	Funkce prepareToPlay	49
6.3.3	Funkce getNextAudioBlock	50
6.3.4	Funkce timerCallBack	51
6.3.4.1	Realizace kalibrace	51
6.3.4.2	Realizace výpočtu hladiny akustického tlaku.....	51
6.3.5	Funkce DrawAudioText	52
6.3.5.1	Funkce Paint	52
7.	průvod měřením aplikací hlukoměru	53
7.1	Volba měření a jeho parametrů	53
7.2	Kalibrace	54
7.3	Měření	54
8.	Závěr	55

Seznam symbolů a zkratk

Symbols:

p_{00}	...	barometrický tlak	[Ω]
W	...	akustický výkon	[W]
p	...	akustický tlak	[Pa]
v	...	akustická rychlost	[m/s]
S	...	plocha	[m ²]
L_p	...	hladina akustického tlaku	[dB]
L_{AF}	...	časově vážená hladina akustického tlaku	[dB]
τ	...	exponenciální časová konstanta	[s]
T	...	časový interval měření	[s]
ξ	...	pomocná proměnná časové integrace	[s]
p_A	...	okamžitý akustický tlak	[Pa]
p_0	...	referenční akustický tlak	[Pa]
I_r	...	intenzita zvuku	[dB]
C_{1000}	...	normující konstanta	[-]
A_{1000}	...	normující konstanta	[-]
Δr	...	vzdálenost mikrofónů od sebe	[dB]
t	...	čas	[s]
ρ	...	referenční akustický tlak	[kg/m ³]

Seznam obrázků

Obrázek 1 Kulová Vlna	14
Obrázek 2 střední ucho[14].....	16
Obrázek 3 kostěný labyrint spánkové kosti[14]	17
Obrázek 4 Cortiho orgán [14].....	18
Obrázek 5 graf signálových toků [9]	26
Obrázek 6 Přímá metoda pro měření intenzity zvuku[1].....	28
Obrázek 7 Polární diagram[14].....	29
Obrázek 8 Frekvenční charakteristika [14].....	30
Obrázek 9 Projucer	33
Obrázek 10 Ukázka volby měření v Command Window	35
Obrázek 11 Kmitočtová odezva filtru FA1	41
Obrázek 12 Kmitočtová odezva filtru FA2.....	41
Obrázek 13 Kmitočtová odezva filtru FC1	42
Obrázek 14 Kmitočtová odezva filtru FC2.....	42
Obrázek 15 Cyklus pro výpočet integrálu a dolní propusti	43
Obrázek 16 Funkce updateToggleState()	48
Obrázek 17 Konstruktor MainComponentu	49
Obrázek 18 Funkce pushNectSampleIntoFifo	50
Obrázek 19 Ukázka uživatelského rozhraní	53

Seznam tabulek

Tabulka 1 Praktické realizace kmitočtových funkcí C a A [1].....	39
Tabulka 2 Deset sekund pro L_{AFmax}	59
Tabulka 3 Deset sekund pro L_{CSmax}	59
Tabulka 4 Deset sekund pro L_{ZSmax}	60
Tabulka 5 Deset sekund pro L_{CIeq}	60

1. ÚVOD

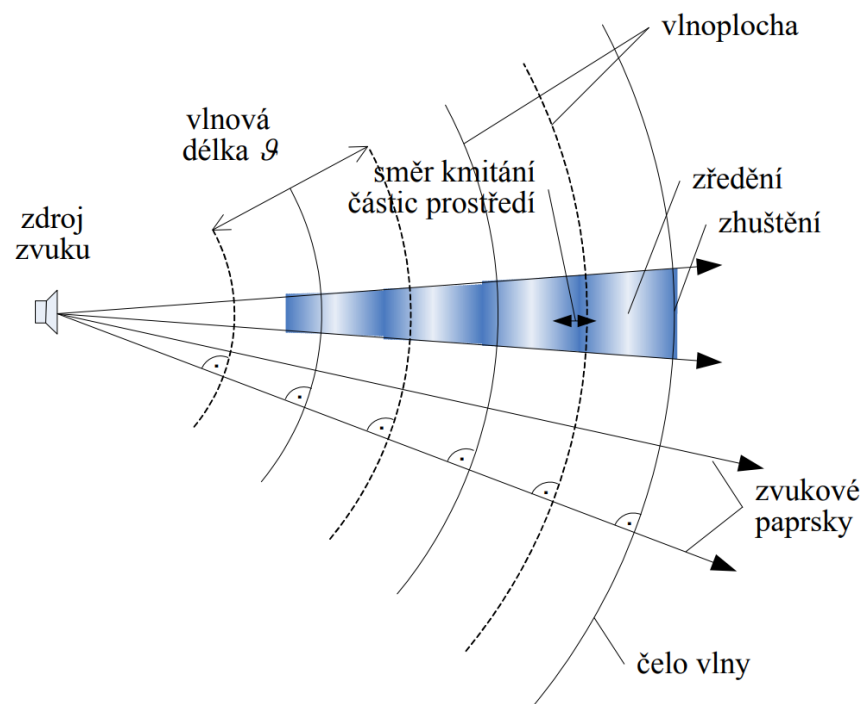
Cílem závěrečné práce bylo vytvořit návrh softwarového hlukoměru v prostředí MATLAB a následně realizovat navržené aplikace v jazyce C++, které budou pracovat reálném čase. Měření hluku je proces, který je zapotřebí ve velké škále profesí. V našem okolí je obrovské množství hluků, jejichž přemíra může mít nežádoucí účinky na sluch a zapříčinit jeho trvalé poškození, nebo dokonce i trvalou ztrátu sluchu. Právě proto je zapotřebí hluk regulovat, nebo o něm přesně informovat a z toho důvodu musí existovat normy, které stanovují, jak se má hluk měřit a jeho maximální přípustné hladiny. V případě dodržení těchto norem hodnota naměřená hlukoměrem je srovnatelná a informativní. V teoretické části se zabírám popisem zvuku a jeho vnímání. Dále se práce zabývá veličinami zvuk popisující, podle normy ČSN EN 61672-1. Jsou zde rozebrána zařízení potřebný pro měření hluku a problematika softwarového měření veličin zvuku v reálném čase. V praktické části se zabírám obsahem mého návrhu softwarového hlukoměru a popisem principu funkčnosti jednotlivým funkcí a aplikací měření. Dále je rozebrána implementace některých funkcí návrhu v C++ při použití rámce JUCE framework a vytvoření aplikace. V poslední části je stručný popis funkčnosti aplikace a jejího použití.

-

2. ZVUK

Zvuk je mechanické vlnění hmotného prostředí (zvukového pole). V prostředí se při šíření zvuku vytváří místa se zředěnými, nebo zhuštěnými částicemi vzduchu, a to v důsledku jeho pružnosti. Tyto částice postupně předávají kinetickou energii dál od zdroje k posluchači, ten pomocí sluchového orgánu vnímá zvuk.

Zvuková vlna je vytvořená kmitáním částic. Jedná se o změnu hustoty v prostředí. Vlnoplocha je plocha, kam se zvuková vlna dostane za stejný čas, od zdroje zvuku. Jsou to místa se stejnou fází. Čelo vlny je poslední vlnoplocha ve směru od zdroje zvuku. Vlnová délka je vzdálenost uražená zvukovým vlněním za jednu periodu kmitání.[1]



Obrázek 1 Kulová Vlna

Zvuky můžeme rozdělit na tóny a hluky, kde tóny jsou periodické signály (využívané převážně v hudbě).

Tóny dále rozdělujeme na složené a jednoduché, kde složené tóny obsahují tzv. vyšší harmonické složky. Poměr vzdáleností, počet a síla vyšších harmonických složek poté určuje barvu tónu (nástroje, hlasu). Vyšší harmonický jsou dány rezonancí

jednotlivých frekvencí v rezonátoru zdroje zvuku (hudby). Ty se pak dají popsat pomocí Fourierovy transformace.

2.1 Šíření a rychlost zvuku

Zvuk se šíří ve vlnoplochách (viz obrázek).

Rychlost zvuku je dána prostředím, ve kterém se zvuk šíří, převážně pak hustotou daného prostředí. Hustota vzduchu úzce souvisí s relativní vlhkostí a teplotou. Pro suchý vzduch je rychlost zvuku $331,4 \text{ m}\cdot\text{s}^{-1}$. V hustším prostředí se potom hůře přenášejí vysoké frekvence, jako příklad se nabízí efekt zvuku pod vodou, naopak rychlost zvuku se v hustších prostředích šíří rychleji.[7]

2.2 Zdroj Zvuku

Zdrojem zvuku se rozumí těleso, co mechanicky rozkmitá prostředí, kterým se vlnění může šířit. Každé chvějící se těleso může být zdrojem zvuku, včetně rezonátorů. Rezonátor je těleso kmitající vynuceně. Toto kmitání mu pak předává původní zdroj zvuku. Rezonátory se potom používají pro zesílení zvuku, popřípadě změně barvy zvuku (poměru a počtu vyšších harmonických složek).[2]

2.3 Vnímání zvuku

Lidské vnímání zvuku není pouze záležitostí fyzikální, ale i psychologické. Popisování zvuku z pohledu našeho lidského vnímání je složitý proces, už jen proto, že náš sluchový orgán nevnímá lineárně ani jednu fyzikální veličinu zvuku[3]. Od toho, jak lidé vnímají zvuk jsou odvozeny jednotky pro měření zvuku. Ovšem zvuk ve své podstatě není jen to, co člověk dokáže slyšet (přesto si to mnozí myslí).

2.4 Sluch

Sluch je schopnost, živočichů vnímat zvuky. Zvuky jsou zpracované uchem, které je orgánem sluchu. Tento párový orgán se skládá ze tří částí

Vnější ucho, střední ucho a vnitřní ucho, přičemž první dvě části jsou vyplněny vzduchem.

2.4.1 Vnější ucho

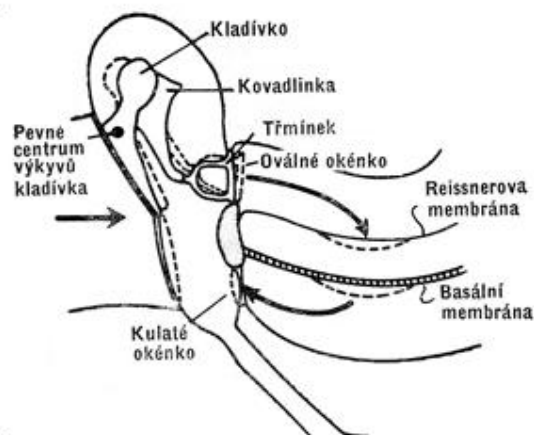
Vnější ucho obsahuje boltec, který je zde pro směřování zvuku dál do vnitřního ucha (tvořen chrupavkou).

Dále obsahuje zvukovod, kterým se vlnění šíří až k bubínku, který je poslední částí vnějšího ucha. Bubínek je vazovitá blána, která příchozí změny tlaku zachycuje pomocí mechanického vlnění (obdoba membrány mikrofону)[14]

2.4.2 Střední Ucho

Střední ucho obsahuje bubínkovou dutinu, ve které je kladívko kovádlínka a třmínek. Kladívko se dotýká bubínku a předává signál kovádlince, která jej předává třmínku (viz obrázek). Třmínek předává chvění do oválného okénka. Zde proběhne zesílení, které je dáno poměrem velikostí dotýkajících se ploch třmínku a oválného okénka. Další důležitou částí vnitřního ucha je napínač bubínku a třmínkový sval. Tyto malé kosterní svaly slouží k tlumení nadměrného hluku, kdy svým napínáním zabraňují nadměrnému chvění kladívka, kovádlinky, třmínku a bubínku (napínání je pak dáno intenzitou zvuku).

Částečně se ve středním uchu nachází Eustachova trubice. Hlavním úkolem Eustachovy trubice je vyrovnávání tlaku ve středním uchu s okolním tlakem.[14]

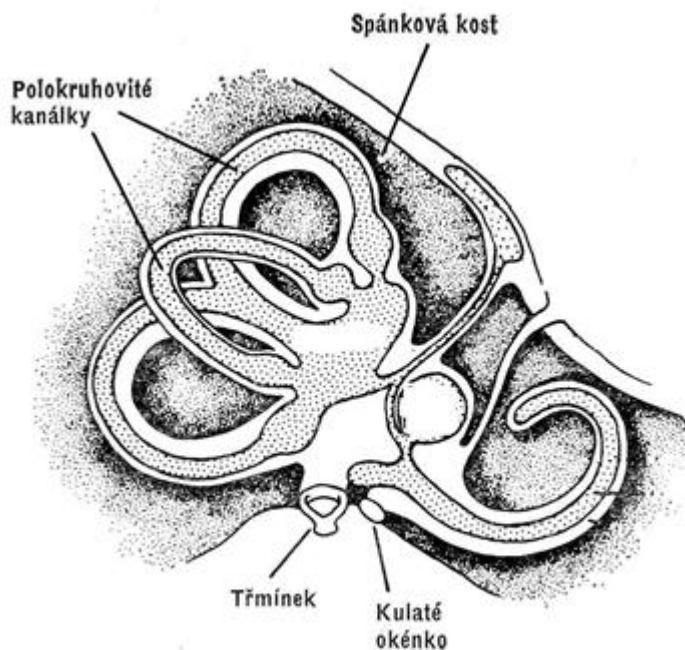


Obrázek 2 střední ucho[14]

2.4.3 Vnitřní Ucho

Tato poslední část sluchového orgánu se nachází v kostěném labyrintu spánkové kosti. Labyrint je potom vyplněný tekutinou, endolymfou (z toho důvodu je potřeba signál zesílit viz šíření zvuku). Jsou zde čtyři hlavní části.

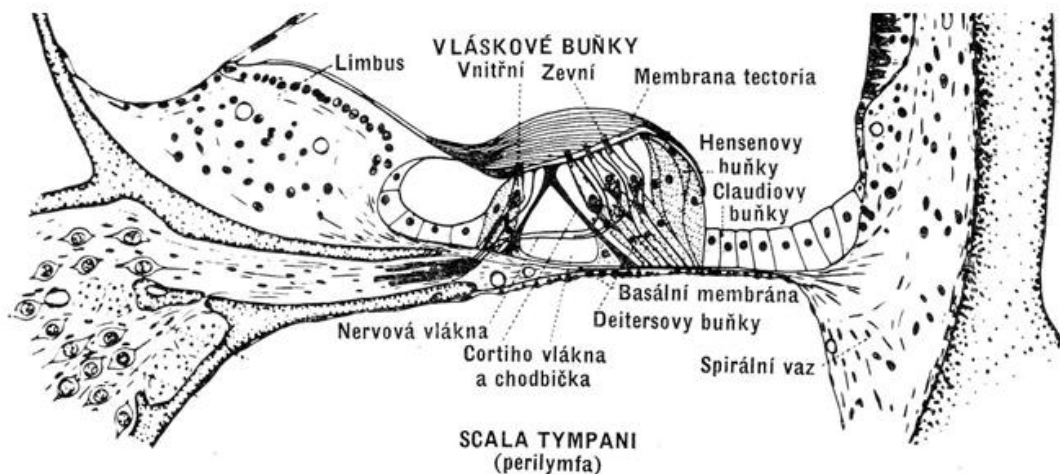
Polokruhové kanálky (přední, boční a zadní kanálek), Vejčitý váček, kulovitý váček a hlemýžď.[14]



Obrázek 3 kostěný labyrint spánkové kosti[14]

Polokruhové kanálky, vejčitý a kulovitý váček slouží především k vnímání zrychlení a polohy, proto nebudou dále rozebírány.

Hlemýžď je postupně se zmenšující zatočená trubička. Ten obsahuje Cortiho orgán, ve kterém jsou vláskovité buňky zodpovědné za zachycení zvuku šířící se tekutinou. Vláskovité buňky jsou sluchové receptory, které transformují mechanické vlnění na elektrický signál. V Cortiho orgánu je potom bazilární membrána, která se postupně rozšiřuje. Při shodném kmitočtu příchozí vlny s rezonančním kmitočtem určité části bazilární membrány se vytvoří impuls (pomocí vláskových buněk). Z vláskových buněk putuje již elektrický signál pomocí kochleárního nervu do mozku. Pouze to, co jsou vláskovité buňky schopny zachytit interpretuje mozek jako zvuk.[14]



Obrázek 4 Cortiho orgán [14]

2.5 Hluk

Jeden z důvodů, proč je potřeba měřit zvukové veličiny je právě hluk. Za hluk považujeme zvuky, které jsou pro náš sluchový orgán nepříjemné (nadměrná hlasitost), mezi ně patří i zvuky jejichž závislost intenzity zvuku na čase není periodická (např. šum).

Hluk potom rozdělujeme do dvou kategorií: hluk v pracovním prostředí (hluk kterému zaměstnanec v práci vystaven) a hluk v mimo pracovním prostředí.

Hluky mimopracovního prostředí se dají potom rozdělit na hluky dopravy a hluky stacionárních zdrojů.[7]

2.5.1 Poškození Sluchu

V případě, že je sluchový orgán vystaven větší intenzitě zvuku, může dojít k jeho poškození. Za nebezpečné se považují zvuky o hlasitosti vyšší než 80 dB_{SPL}.

Hladina akustického tlaku od 80 dB_{SPL} může poškodit sluch již po 8 hodinách. Čím vyšší je hladina akustického tlaku, tím se krátí čas, za který je možno sluch poškodit.[8]

2.6 Akustický tlak

Akustický tlak je skalární veličina, z definice tlaku víme, že tlak je síla působící na jednotku plochy. Akustický tlak je rozdíl mezi okamžitou velikostí tlaku v daném bodě za přítomnosti zvuku a statickým barometrickým tlakem $p_{00} = 10^5$. Jednotou tlaku je Pascal ($1 \text{ Pa} = \text{N/m}^2$). [1][2]

2.7 Akustický výkon

Akustický výkon je mechanický výkon (práce vykonaná za jednotku času) vyjádřený pomocí akustických veličin. Vyjadřujeme jej vztahem [1]

$$W = p \cdot v \cdot S \quad (2.1)$$

Kde

W je akustický výkon (W)

p je akustický tlak (Pa)

v je akustická rychlost a (m/s)

S je plocha, na kterou je rovnoměrně působící tlak (m^2)

2.8 Referenční akustický tlak

Jedná se o referenční veličinu. Pro zvuk, který se šíří vzduchem je tato veličina rovna $20 \mu\text{Pa}$. Je to minimální hodnota akustického tlaku, kterou může lidské ucho zachytit.

[1]

3. MĚŘENÍ HLUKU

3.1 Stupnice pro měření hluku

Vzhledem k tomu, že lidské ucho nevnímá hluk lineárně, je nejlepší měřítko pro znázornění hluku logaritmické. U měření hluku používáme decibelové veličiny, které se značí dB. Decibelové veličiny převádějí základní akustické veličiny na decibelové hladiny na základě jejich referenčních hodnot. [7]

3.1.1 Číslo třídy hluku

Číslo třídy hluku se používá při vyhodnocování hluků, u kterých je riziko poškození sluchového orgánu (nad 80 dB_{SPL}). Při hlucích s nižší hladinou akustického tlaku se častěji používá hladina akustického tlaku. [2]

3.2 Hladina Akustického tlaku

Jedná se o dvacetinásobek logaritmované efektivní hodnoty, která je vztažena k výše uvedenému referenčnímu tlaku. Vyjadřuje se v decibelech (dB) a její značka je L_p [1].

3.2.1 Vážení v kmitočtu

V případě zvukoměru je kmitočtové vážení rozdíl mezi indikovanou hodnotou a skutečným vstupem. Tento rozdíl je dán tzv. Kmitočtovými váhovými funkcemi A, C, Z [1].

3.2.2 Vážení v čase

Jedná se o exponenciální funkci času, kterou se váží kvadrát okamžitého akustického tlaku a má stanovenou časovou konstantu Slow, nebo Fast. [1]

3.2.3 Časově vážená hladina akustického tlaku

Jedná se o dvacetinásobek logaritmované efektivní hodnoty, která je vztažena k referenčnímu tlaku p_0 . Efektivní hodnota akustického tlaku je počítána po časovém a kmitočtovém vážení. Vyjadřuje se v decibelech. Značí se L a ve spodním indexu se nachází typ vážení (A, C) a typ časové konstanty (Slow, Fast) L_{AF} , L_{AS} , L_{CF} , L_{CS} . V symbolickém vyjádření je časově vážená hladina akustického tlaku A $L_{At}(t)$ reprezentována v jakémkoliv časovém okamžiku t . [1]

$$L_{At} = 20 \log_{10} \left\{ (1/\tau) \left[\int_{-\infty}^t p_A^2(\xi) e^{-(t-\xi)/\tau} d\xi \right]^{1/2} / p_0 \right\} \quad (3.1)$$

Kde

- τ je exponenciální časová konstanta S (slow 1 s), nebo F (fast 0,125 s) (s)
- ξ je pomocná proměnná časové integrace. Od počátku do času t (s)
- $p_A(\xi)$ je okamžitý akustický tlak A ; a (Pa)
- p_0 referenční akustický tlak (Pa)

3.2.4 Maximální časově vážená hladina akustického tlaku

Nejvyšší hodnota časově vážené hladiny akustického tlaku v stanoveném časovém intervalu. Vyjadřuje se v decibelech a značí se L_{AFmax} (pro vážení A časovou konstantu fast). [1]

3.2.5 Špičkový akustický tlak

Nejvyšší okamžitá absolutní hodnota akustického tlaku během stanoveného časového intervalu . [1]

3.2.6 Hladina špičkového akustického tlaku

Dvacetinásobek dekadického logaritmu špičkového akustického tlaku vztaženého k referenčnímu akustickému tlaku. [1]

3.2.7 Časově průměrovaná hladina akustického tlaku

Dvacetinásobek dekadického logaritmu efektivní hodnoty akustického tlaku za stanovený časový interval vztažené k referenčnímu akustickému tlaku. Akustický tlak se filtruje kmitočtovým váhovým filtrem. Vyjadřuje se v decibelech. [1]

$$L_{AT} = L_{AeqT} = 20 \log_{10} \left\{ (1/T) \left[\int_{t-T}^t p_A^2(\xi) d\xi \right]^{1/2} / p_0 \right\} \quad (3.2)$$

Kde

- T časový interval průměrování (s)
 ξ je pomocná proměnná časové integrace. Od počátku do času t (s)
 $p_A(\xi)$ je okamžitý akustický tlak A ; a (Pa)
 p_0 referenční akustický tlak (Pa)

3.3 Zvuková expozice

Časový integrál kvadrátu akustického tlaku ve stanoveném časovém intervalu.

Doba integrace je volitelná hodnota – doba zvukové expozice. [1]

3.3.1 Hladina zvukové expozice

Desetinásobek dekadického logaritmu zvukové expozice vztaženého k referenční zvukové expozici.

Referenční zvuková expozice je dána součinem kvadrátu akustického tlaku a referenčního časového intervalu 1 s. [1]

3.4 Vážení v kmitočtu

Jak je zde již několikrát zmíněno lidský sluchový orgán není lineární. Citlivost lidského zvuku není na všech kmitočtech stejná. Z tohoto důvodu se zavádějí váhové filtry. Ty používáme k přepočítání naměřených hodnot hladiny zvuku na hladinu zvuku tak jak jí vnímá lidský orgán [5]. Pro názornou ukázkou různé citlivosti hluku v závislosti na kmitočtu a hlasitosti slouží křivky stejné hlasitosti. Křivky znázorňují, jaký akustický tlak (v dB_{SPL}) působí pro lidský orgán hlasitostně stejně, jako referenční tón s kmitočtem 1 kHz. Od těchto hladin jsou odvozeny váhové funkce. Jednotkou hladiny hlasitosti je jeden fon [Ph].

-
-

3.4.1 Váhové filtry

Kmitočtové charakteristiky jsou ke korekčním křivkám inverzní, a to z toho důvodu, že se s naměřeným spektrem signálu sčítají [1]. Kmitočtové filtry se používají v mnoha odvětvích akustiky nebo elektroakustiky. Nejvíce používané filtry jsou A, C a Z, ovšem filtr Z značí, že spektrum signálu nebylo nikterak upravováno, tudíž má na všech hodnotách kmitočtu hodnotu 0 dB. Kmitočtová charakteristika váhového filtru A je odvozená od křivky hlasitosti na 40 Ph. Pro váhový filtr C je kmitočtová charakteristika odvozena od křivky hlasitosti na 100 Ph. Na kmitočtu 1 kHz je pro všechny kmitočtové váhové funkce jmenovitá hodnota 0 dB. Kmitočtová charakteristika váhové funkce C se realizuje pomocí dvou pólů na nízkém kmitočtu f_1 a dvou pólů na vysokém kmitočtu f_4 . Kmitočtová charakteristika váhové funkce A

se realizuje tak, že se kmitočtová charakteristika váhové funkce C doplní o dvě vázané hornofrekvenční propusti prvního řádu. Mezní kmitočet je u obou hornofrekvenčních propustí dán $f_A=10^{2,45}$ Hz. [1]

Na všech kmitočtech f se váhová funkce $C(f)$ vypočítá[1]

$$C(f) = 20 \log_{10} \left[\frac{f_4^2 f^2}{(f^2 + f_1^2)(f^2 + f_4^2)} \right] - C_{1000} \quad (3.3)$$

Váhová funkce $A(f)$ se vypočítá.

$$A(f) = 20 \log_{10} \left[\frac{f_4^2 f^4}{(f^2 + f_1^2)(f^2 + f_2^2)^{1/2}(f^2 + f_3^2)^{1/2}(f^2 + f_4^2)} \right] - A_{1000} \quad (3.4)$$

Kde C_{1000} a A_{1000} jsou normující konstanty, kterými reprezentujeme elektrický zisk potřebný k tomu, aby kmitočtové funkce váhové funkce měly zisk 0 dB na kmitočtu 1 kHz **Chyba! Nenalezen zdroj odkazů..**[1]

Váhová funkce Z vypadá takto **Chyba! Nenalezen zdroj odkazů.**[1]

$$Z(f) = 0 \quad (3.5)$$

Pro výpočet pólů f_1 a f_4 se používá kvadratická rovnice[1] **Chyba! Nenalezen zdroj odkazů.**

$$f_1 = \left[\frac{-b - \sqrt{b^2 - 4c}}{2} \right]^{1/2} \quad (3.6)$$

$$f_4 = \left[\frac{-b + \sqrt{b^2 - 4c}}{2} \right]^{1/2} \quad (3.7)$$

Pro výpočet konstant b a c odvodíme [1] **Chyba! Nenalezen zdroj odkazů.**

$$f_3 = \left(\frac{1}{1-D} \right) \left[f_r^2 \left(1 + \frac{f_L^2 f_H^2}{f_r^2} - D(f_L^2 + f_H^2) \right) \right] \quad (3.8)$$

$$c = f_L^2 f_H^2 \quad (3.9)$$

Kde D^2 je $1/2$, $f_r = 1000$, $f_L = 10^{1,5}$, $f_H = 10^{3,9}$ (Hz)

Při mezním kmitočtu f_A (viz. výše) se póly charakteristiky pro kmitočty f_2 a f_3 používají vztahy.[1]

$$f_2 = \left(\frac{3-\sqrt{5}}{2}\right) f_A \quad (3.10)$$

$$f_3 = \left(\frac{3+\sqrt{5}}{2}\right) f_A \quad (3.11)$$

3.4.2 Číslicové filtry

Problematika váhových filtrů v digitální sféře

Výše popsané rovnice jsou pro analogové přístroje, ovšem většina dnes používaných přístrojů pro výpočet zvukových veličin jsou digitální. Pro návrh digitálních váhových filtrů je tudíž potřeba transformovat rovnice do digitální domény. Pro tento přepočít se používá transformace (viz. možné transformace analogových filtrů).

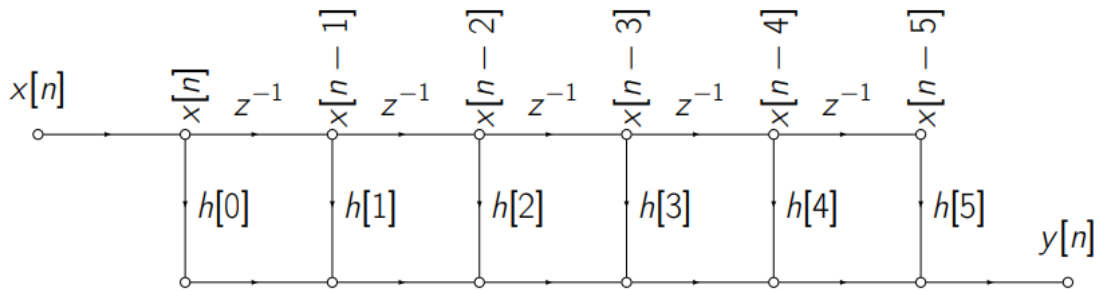
Digitální filtry rozdělujeme na dva druhy: FIR (finite impulse response) a IIR (infinite impulse response).[9][10]

3.4.2.1 FIR

Filtry s konečnou impulsní odezvou fungují tak, že vynásobí aktuální a předchozí vzorek signálu tzv. koeficienty filtru. Počítají se tedy pro nějaký počet vzorků (v případě softwarového hlukoměru vytvořeném v prostředí MATLAB se aplikuje po sekundách, v případě C++ bychom aplikovali filtr na buffer). Při návrhu FIR filtrů musí být splněna podmínka kauzality. Při splnění kauzality je FIR filtr stabilní. FIR filtr s přenosovou funkcí (3.12) je vždy kauzální (tudíž i stabilní).

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{N-1} b_i z^i}{z^{N-1}}, a_s =, s = r = N - 1 \quad (3.12)$$

Pro návrh FIR filtrů se používají metody: Metoda váhové posloupnosti, metoda vzorkování kmitočtové charakteristiky, metoda rovnoměrné zvlněné aproximace (též optimální metoda)



Obrázek 5 graf signálových toků [9]

Výhody: zaručená stabilita, možnost návrhu s lineární fází, jednodušší implementace v kódu

Nevýhody: Vysoký nárok na výpočet, pro realizaci strmějších filtrů je potřeba velký počet koeficientů, viz realizace filtrů v prostředí MATLAB. [9]

3.4.2.2 IIR

Filtry s nekonečnou impulsní odezvou realizují výpočet s aktuální vstupní hodnotou vzorku. Při realizaci se používá tzv. rekurzí (zpětných vazeb) a zpětnovazebních struktur. Pro dodržení kauzality je zapotřebí, aby polynom v čitateli byl menší nebo roven polynomu ve jmenovateli. Po návrhu je nutno provést kontrolu pólů přenosové funkce a ujistit se, že všechny lež uvnitř jednotkové kružnice (z důvodů stability).

$$H(z) = \frac{\sum_{i=0}^M b_i z^i}{\sum_{i=0}^N a_i z^i} \quad (3.13)$$

Výhody: Mnohem menší řád přenosové funkce (oproti FIR se uvádí desetinásobek),

Nevýhody: Není vždy zaručená stabilita filtru

[9]

3.4.2.3 Transformace analogových filtrů

Z výše uvedeného textu vyplývá, že pro realizaci váhových filtrů v digitální doméně je potřeba využití IIR filtrů, pro dostatečnou přesnost na velmi nízkých a velmi vysokých frekvencích (především pak u váhového filtru typu A). Pro tyto převody se

nejčastěji používá bilaterální transformace, nebo metoda invariance impulsní charakteristiky. [9]

3.5 Intenzita zvuku

Intenzita zvuku vyjadřuje činnou složku měrného akustického výkonu. Měrný akustický výkon je akustický tlak, nad kterým uvažujeme, jako nad silou působící na jednotku plochy. Tento měrný akustický výkon je znázorněn pomocí okamžitých hodnot akustického tlaku p a akustické rychlosti v . Pro výpočet Intenzity zvuku používáme vztah[1].

$$I_r = \frac{1}{T} \int_0^T p(t)v_r(t)dt \quad (3.14)$$

Kde

$p(t)$ je okamžitý akustický tlak (Pa)

$v_r(t)$ je složka okamžité akustické rychlosti ve směru r (m/s)

T je interval měření (s)

Pokud je vlnová délka větší, než je vzdálenost mikrofونů od sebe můžeme počítat[1]

$$I_r = \frac{p_A + p_B}{2\rho\Delta r} \int_0^T (p_B - p_A)dt \quad (3.15)$$

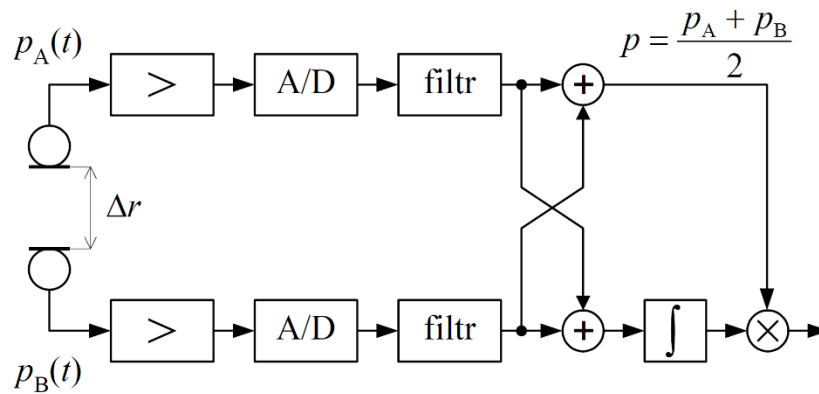
Kde

p_A je akustický tlak prvního mikrofونu (Pa)

p_B je akustický tlak druhého mikrofونu (Pa)

ρ je hustota vzduchu (kg/m^3)

Δr je vzdálenost mikrofونů od sebe



Obrázek 6 Přímá metoda pro měření intenzity zvuku[1]

3.6 Měřicí zařízení

3.6.1 Mikrofony

Mikrofon je elektroakustický měnič, který přeměňuje akustický signál na signál elektrický. Je základním článkem elektroakustického řetězce. Mikrofony můžeme rozdělit na:

- 1) Elektrodynamické mikrofony
- 2) Kondenzátorové mikrofony (kapacitní, typu RF)
- 3) Piezoelektrické
- 4) Uhlíkové
- 5) Páskové
- 6) Speciální

Mikrofony dále popisujeme podle jejich vlastností, které jsou do jisté míry dány způsobem transformace signálu. [14]

3.6.1.1 Citlivost

Citlivost je hlavním parametrem mikrofonu. Citlivost mikrofonu určíme měřením úrovně výstupního signálu pro 1 Pa akustický tlak při 1kHz. Citlivost mikrofonu je dána poměrem efektivní hodnoty výstupního napětí k akustickému tlaku.[14]

$$N = U/p \text{ [mV/Pa]}$$

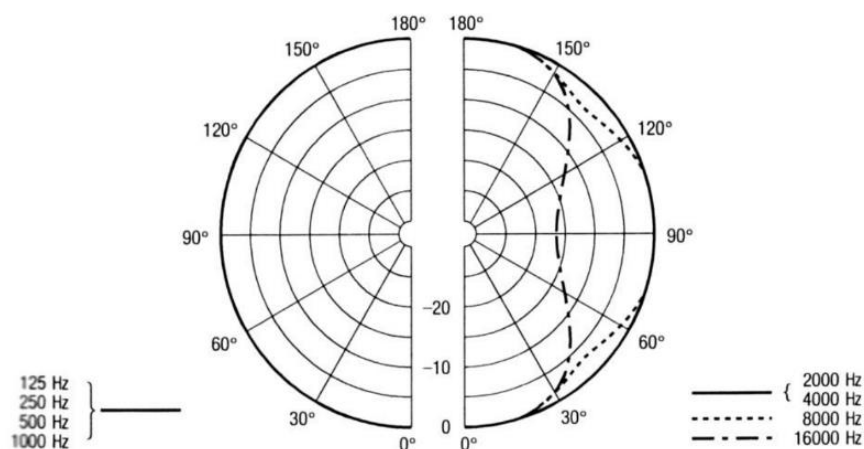
dynamické	$\eta = 1 - 5 \text{ mV/Pa}$
kondenzátorové	$\eta = 20 - 50 \text{ mV/Pa}$
elektretové	$\eta = 1 - 30 \text{ mV/Pa}$

3.6.1.2 Mezní akustický tlak

Mezní akustický tlak je horní hranice hodnoty akustického tlaku, kterou je mikrofon schopen zaznamenat bez zkreslení. To může být způsobeno nedokonalou mechanickou tuhostí, nebo poddajností použitých materiálů. Při zkreslení signálů, se zkreslí (znehodnotí) výsledky měření hlukoměru.[14]

3.6.1.3 Směrová charakteristika

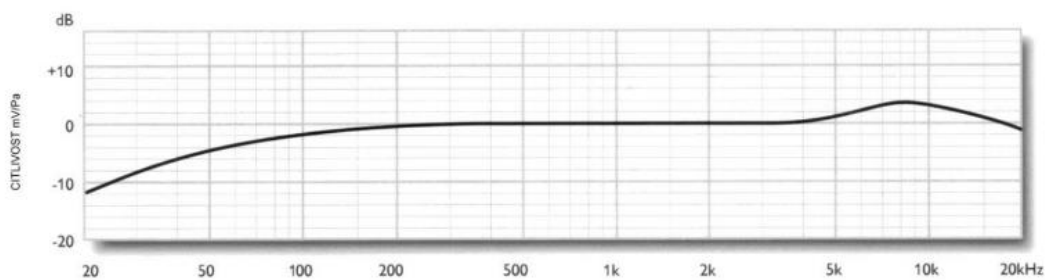
Směrová charakteristika mikrofonu nám popisuje schopnost mikrofonu zachycovat zvuky z různých směrů. Měří se v bezodrazové místnosti, kde se mikrofon otáčí vodorovně kolem své osy na otáčivém talíři. Referenční frekvence, které se na mikrofon vysílají dopadají na jeho membránu pod neustále se měnícím úhlem. Výstupní napětí se zapíše pomocí zapisovače. Směrová charakteristika se znázorňuje pomocí polárního diagramu.[14]



Obrázek 7 Polární diagram[14]

3.6.1.4 Frekvenční charakteristika

Frekvenční charakteristika určuje, jaké frekvence mikrofon zachytává v závislosti na síle mikrofonem transformovaného signálu. Je to závislost citlivosti mikrofonu na frekvenci pro konstantní akustický tlak (z pravidla referenční akustický tlak). Popisuje se křivkou závislosti citlivosti na frekvenci. Měří se ve směru akustické osy mikrofonu.



Frekvenční charakteristika mikrofonu Neumann KM 184.

Obrázek 8 Frekvenční charakteristika [14]

Pro potřeby hlukoměru se využívá co nejrovnější směrová charakteristika.

-

3.6.2 Hlukoměry

Hlukoměry (též zvukoměry) jsou měřicím přístrojem pro určování zvukových veličin. Hlukoměr může být jak analogový, tak digitální. Protože hluky a zvuky je potřeba měřit přesně, existuje několik norem, které definují výpočty a způsoby jejich aplikace pro hlukoměr. [2]

-

3.6.2.1 Kalibrace hlukoměrů

Kalibrace (též „justování“) je u hlukoměru potřebná ke správnému měření. Zajišťuje převod efektivní hodnoty elektrického napětí z mikrofonu na hladinu akustického tlaku. Díky tomu je hlukoměr schopen určit, pro jakou hodnotu napětí je ekvivalentní hladina akustického tlaku.

Kalibrace rozdělujeme na akustické a elektrické.

Akustická kalibrace se provádí pomocí pistonfonu, anebo tónového kalibrátoru.

Základní vlastností obou kalibračních zařízení je vytvoření přesné hodnoty akustického tlaku, díky čemuž se pak hodnoty přepočítávají (s touto metodou kalibrace měřicí program LoudnessMM počítá).

K elektrické kalibraci je zapotřebí přesný zdroj kalibračního napětí, který potom za pomoci citlivosti mikrofonu (viz mikrofony) přepočítaná na akustický tlak.

Elektrické kalibrátory jsou nejčastěji součástí hlukoměru.

[2]

3.6.2.2 Požadavky na prostředí

Nežádoucí vliv na měření hlukoměru má rosný bod, proto se v souladu s normou ČSN EN 61672-1 nesmí v teplotách na d 39 c a pod -15 c s kombinací relativní vlhkosti vzduchu, jejímž výsledkem je rosný bod[1]

4. ZPRACOVÁNÍ AUDIO SIGNÁLU V REÁLNÉM ČASE

4.1 Vyrovnávací paměť

Při zpracování signálu v reálném čase největší problémy nastávají s vyrovnávací pamětí. Ta je určena k dočasnému uchování dat. Při realizaci programu, který tuto paměť využívá je zapotřebí hlídat její velikost a nepřetěžovat ji (neuchovávat moc dat najednou). Hlavním úkolem vyrovnávací paměti je vyrovnávání rychlostí přijímání dat a jejich zpracování. V případě, že bude výpočet trvat déle nežli příjem dat, aplikace nepočítá v reálném čase, ale se zpožděním. V mnohých případech se pak aplikace stává nestabilní.

4.1.1 Ukazatele

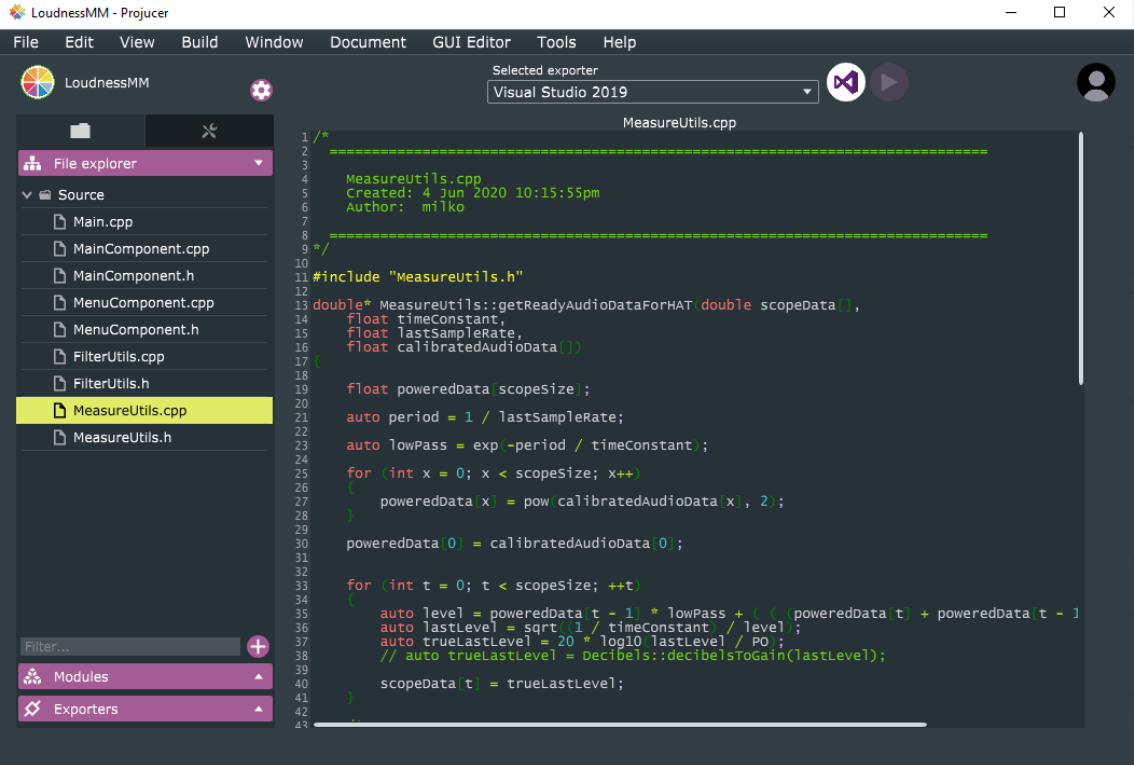
Ukazatel jsem datový typ (jako např. intiger, float, nebo double), který obsahuje cestu (adresu) k datům, které pak zpracováváme. Ukazatelé umožňují ulehčit vyrovnávací paměti od dat. Též se využívají u práci s daty jejichž velikost předem neznáme, nebo se v čase mění.

4.1.2 Abstraktní datový typ list

V případě, že předem neznáme velikost prvků, které budeme zpracovávat, popřípadě je jejich množství volitelný parametr, je krom ukazatelů možné použít i abstraktní datový typ list. Datový typ list představuje počítatelný počet hodnot na kterých se může stejná hodnota se vyskytují více než jednou. Dá se jednoduše popsat jako pole, které můžeme v průběhů výpočtů zvětšovat o předem neznámí počet prvků.

4.2 JUCE framework

Juce framework je aplikační rámec C++. Jeho zdrojový kód se sestavuje a spouští identicky na všech operačních systémech (Windows, Linux, Mac OS). Dále Juce podporuje více vývojových prostředí a kompilátorů. Pro Windows je největší podpora u Visual Studia a CLionu. Juce obsahuje JUCE třídy, které poskytují velký rámec funkcí. Důležité třídy (potažmo funkce) jsou podrobně rozebrány v části implementace funkcí v c++. Z hlediska zvuku Juce podporuje MIDI, polyfonické syntetizátory, většinu běžných audio zařízení a má vystavěné čtečky zvukových souborů (wav, flac atd.) Nejvíce používaný (alespoň podle diskuzí na internetu) je pro vytváření efektů, nástrojů vst a plug-inů. Pro vytváření kódu JUCE používá Projucer.[13]



```
1 /*
2
3
4 MeasureUtils.cpp
5 Created: 4 Jun 2020 10:15:55pm
6 Author: milko
7
8
9 */
10
11 #include "MeasureUtils.h"
12
13 double* MeasureUtils::getReadyAudioDataForHAT (double scopeData[] ,
14 float timeConstant,
15 float lastSampleRate,
16 float calibratedAudioData [])
17 {
18
19 float poweredData scopeSize ;
20
21 auto period = 1 / lastSampleRate;
22
23 auto lowPass = exp -period / timeConstant ;
24
25 for (int x = 0; x < scopeSize; x++)
26 {
27     poweredData x = pow calibratedAudioData x , 2 ;
28
29
30 poweredData 0 = calibratedAudioData 0 ;
31
32
33 for (int t = 0; t < scopeSize; ++t
34
35     auto level = poweredData t - 1 * lowPass + poweredData t + poweredData t - 1
36     auto lastLevel = sqrt 1 / timeConstant / level ;
37     auto trueLastLevel = 20 * log10 lastLevel / 10 ;
38     // auto trueLastLevel = Decibels::decibelstogain(lastLevel);
39
40     scopeData t = trueLastLevel;
41
42
43 }
```

Obrázek 9 Projucer

5. REALIZACE V PROSTŘEDÍ MATLAB

V této kapitole se budu podrobně zabývat jednotlivými m-soubory softwarového hlukoměru. Nachází se zde soubor pro komunikaci s uživatelem a soubory, které obsahují funkce pouze pro výpočet, nebo načtení. Mezi těmito soubory je Skript, který umožňuje ovládání hlukoměru a volá jednotlivé funkce, *nacteni*, což je funkce pro načítání měřených nebo kalibračních souborů, *filtrA*, funkce pro filtraci signálů pomocí váhové funkce A, *filtrC*, funkce pro filtraci signálů pomocí váhové funkce C, *HAT* je funkce pro výpočet hladiny akustického tlaku, *HZE* je funkce pro výpočet hladiny zvukové expozice, *IZ* je funkce pro výpočet intenzity zvuku a *Po*, což je globální proměnná referenčního tlaku p_0 .

5.1 Skript

Skript nás pomocí Command Window provádí měřením a volá jednotlivé funkce podle námi zadaných parametrů

5.1.1 Volba měření

V této části skriptu se volí měření mezi Hladinou akustického tlaku, Hladinou zvukové expozice a intenzity zvuku. V Command Window se vytisknou tyto možnosti a k nim přiřazená čísla které program načítá do proměnné *UrceniMereni*. Pomocí proměnné *UrceniMereni* program určuje, jaké měření si uživatel vybral

```
>> Skript
Volte měření
Hladina akustického tlaku - 1
Hladina hlukové expozice - 2
Intenzita zvuku - 3
```

Obrázek 10 Ukázka volby měření v Command Window

5.1.2 Volba časové konstanty

Volba konstanty je část skriptu, která se provede pouze, když si uživatel vybral možnost měření hladiny akustického tlaku. V Command Window se vytiskne text s výzvou k volbě časové konstanty s dvěma možnostmi slow a fast, obdobně jako u první části Skriptu.

5.1.3 Volba intervalu měření

Volba intervalu měření je část skriptu, která se provede pouze, když si uživatel vybral možnost měření hladiny zvukové expozice. V Command Window se vytiskne obdobný text jako u předchozích částí viz. výše.

5.1.4 Zadání hustoty vzduchu a distance mikrofonů

Tato část skriptu se provede pouze v případě, že si uživatel zvolí měření intenzity zvuku. Tyto hodnoty jsou pro výpočet důležité viz. (3.13)

5.1.5 Načtení souborů

Tato část skriptu volá funkci načtení a probíhá vždy. Blíže je funkce načtení popsána dále v textu.

5.1.6 Volba vážení

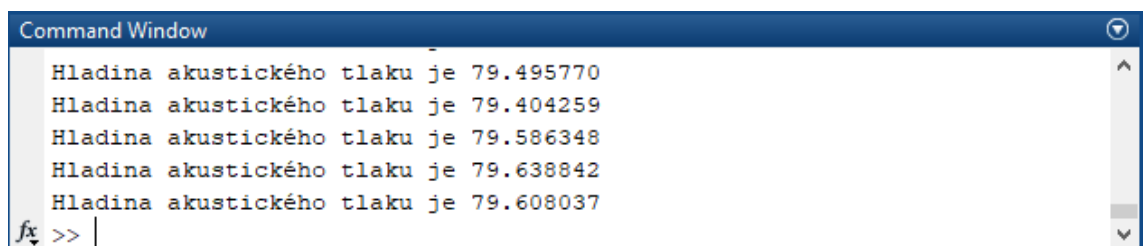
Tato část skriptu probíhá vždy. V Command Window se vytiskne výzva k volbě váhového filtru viz. výše. Uživatel může zvolit mezi filtry A, C a Z.

5.1.7 Aplikace vážení

Tato část skriptu proběhne vždy. Podle proměnné „*UrceniMereni*“ volá funkci pro výpočet hladiny akustického tlaku (hodnota proměnné 1), hladiny zvukové expozice (hodnota proměnné 2), nebo Intenzity zvuku (hodnota proměnné 3). Výstup funkce poté vkládá do proměnné „*Vysledek*“

5.1.8 Tisk výsledku

Závěrečná část skriptu, která probíhá vždy. Tato část upravuje text, který se tiskne (pomocí funkce *fprintf*) v Command Window, podle proměnné „*UrceniMereni*“, text příslušící danému měření. Za tímto textem vytiskne hodnotu proměnné „*Vysledek*“.



```
Command Window
Hladina akustického tlaku je 79.495770
Hladina akustického tlaku je 79.404259
Hladina akustického tlaku je 79.586348
Hladina akustického tlaku je 79.638842
Hladina akustického tlaku je 79.608037
fx >>
```

Obrázek 9 Vytisknutí výsledků v Command Window

5.2 Po

Funkce *Po* definuje hodnotu referenčního akustického tlaku p_0 . Vytváří globální proměnnou „*Po*“ s hodnotou $2 \cdot 10^{-5}$. Tuto funkci jsem vytvořil, protože referenční akustický tlak je hodnota, která je potřeba v několika funkcích a součástí návrh je tato globální proměnná.

5.3 Funkce nacteni

Funkce *nacteni* slouží pro načítání wav souborů. Výstupy této funkce jsou jednotlivé vzorky, převedené na hodnotu akustického tlaku a vzorkovací frekvence. Vstupem této funkce je proměnná *UrceniMereni* ze Skript.m.

Po zavolání si funkce zkontroluje, co chce uživatel načítat (podle proměnné *UrceniMereni*). Pokud má proměnná hodnotu „3“ (Měření intenzity zvuku), funkce *nacteni* nejdříve počítá s kalibračním souborem a vyzve uživatele k jeho vybrání. Zkontroluje si, zda je to kalibrační soubor, pomocí funkce *strfind()*.

Argumenty funkce *strfind()* je proměnná typu string, ve které chceme hledat slovo, které potom hledá. Funkce vrátí pozici prvního písmene slova v proměnné typu string. Pokud se zde slovo nenachází, vrátí prázdnou proměnnou[6]. Pokud se v názvu souboru, který uživatel vybral objeví slovo *calibration* poté program hledá číslo v názvu tohoto souboru, které reprezentuje úroveň hlasitosti v dB_{SPL}. Jelikož je tento kalibrační soubor dvoukanálový (stereofoní), tak si program musí rozdělit matici vzorků na dva jednobanálové (monofoní) signály. Dále si funkce přepočítá vzorky na úroveň v dBFS podle vztahu (4.1) najde si průměrnou hodnotu a od proměnné „z“ (která má hodnotu čísla reprezentující úroveň v dB_{SPL} ji odečte. Tímto krokem získá hodnotu v dB_{SPL} pro 0 dBFS, kterou potřebujeme pro převedení vzorků.

Poté vyskočí druhé dialogové okno, které vyzve uživatele pro načtení souboru, který chce měřit. Tento soubor je opět stereofoní, a proto se musí zase rozdělit do dvou vektorů. Po rozdělení je funkce převede úrovně dB_{SPL} pro 0 dBFS.

$$10 \cdot \ln\left(\frac{x_1}{1}\right) \quad (4.1)$$

Kde

x_1 je proměnná obsahující sumu kvadrátu jednotlivých vzorků podělenou jejich počtem

$$P_A = data(P_0 10^{Z/20}) \quad (4.2)$$

Kde

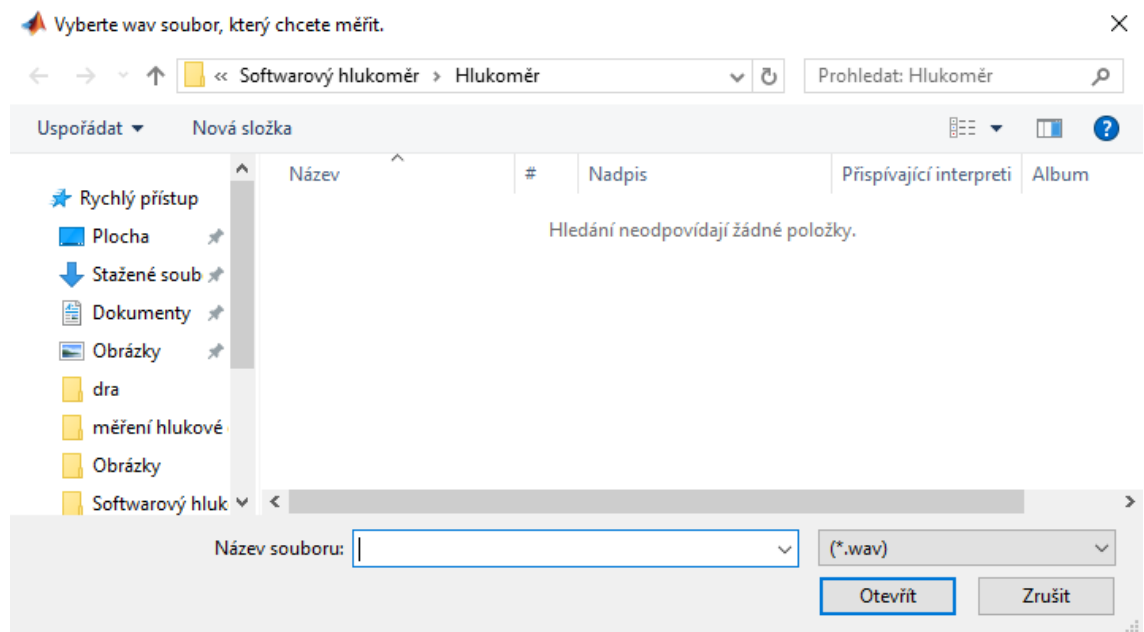
p_A je hodnota akustického tlaku (dB_{SPL})

data je hodnota (dBFS)

Z je převodní konstanta určená referenčním hlukoměrem (dB_{SPL})

Pokud se nejedná o měření intenzity zvuku, tak program akorát otevře dialogové okno a vyzve pro načtení souboru, ze jména souboru si vyjme úroveň dB_{SPL} pro 0 dBFS a převede vzorky podle vztahu (4.3).

otevře dialogové okno pro výběr wav souboru, který chce uživatel měřit. Toto okno se otevírá pomocí funkce „*uigetfile*“ [6], která načte název vybraného souboru do proměnné „soubor“ a cestu k tomuto souboru do proměnné „cesta“ (ovšem bez názvu souboru). Poté pomocí funkce „*audioread*“ se do proměnných „data“ a „fs“ načtou data a vzorkovací frekvence. Argumentem funkce „*audioread*“ je proměnná „*CelaCesta*“, která obsahuje cestu a název souboru spojených pomocí funkce „*strcat*“.[6]



Obrázek 10 Dialogové okno

5.4 Funkce FiltrA a FiltrC

Pro výpočet kmitočtových funkcí A a C používám funkci *fir2()*. Funkce *fir2()* slouží k návrhu filtrů typu FIR (Finite Impulse Response), což jsou filtry s konečnou impulzní odezvou. Pro návrh těchto filtrů jsem použil tabulku praktické realizace kmitočtových funkcí C, A a Z odvozených z analytických výrazů podle rovnic (3.3) a (3.4)[6] [9][1]

Tabulka 1 Praktické realizace kmitočtových funkcí C a A[1]

Jmenovitý Kmitočet Hz	Kmitočtová váhová funkce dB			Tolerance (dB)	
	A	C	Z	Třída	
				1	2
10	-70,4	-14,3	0,0	+3,5; -∞	+5,5; -∞
12,5	-63,4	-11,2	0,0	+3,0; -∞	+5,5; -∞
16	-56,7	-8,5	0,0	+2,5; -4,5	+5,5; -∞
20	-50,5	-6,2	0,0	±2,5	±3,5
25	-44,7	-4,4	0,0	+2,5; -2,0	±3,5
31,5	-39,4	-3,0	0,0	±2,0	±3,5
40	-34,6	-2,0	0,0	±1,5	±2,5
50	-30,2	-1,3	0,0	±1,5	±2,5
63	-26,2	-0,8	0,0	±1,5	±2,5
80	-22,5	-0,5	0,0	±1,5	±2,5
100	-19,1	-0,3	0,0	±1,5	±2,0
125	-16,1	-0,2	0,0	±1,5	±2,0
160	-13,4	-0,1	0,0	±1,5	±2,0
200	-10,9	0,0	0,0	±1,5	±2,0
250	-8,6	0,0	0,0	±1,4	±1,9
315	-6,6	0,0	0,0	±1,4	±1,9
400	-4,8	0,0	0,0	±1,4	±1,9
500	-3,2	0,0	0,0	±1,4	±1,9
630	-1,9	0,0	0,0	±1,4	±1,9
800	-0,8	0,0	0	±1,4	±1,9
1000	0	0	0,0	±1,1	±1,4
1250	+0,6	0,0	0,0	±1,4	±1,9
1600	+1,0	-0,1	0,0	±1,6	±2,6
2000	+1,2	-0,2	0,0	±1,6	±2,6
2500	+1,3	-0,3	0,0	±1,6	±3,1
3150	+1,2	-0,5	0,0	±1,6	±3,1
4000	+1,0	-0,8	0,0	±1,6	±3,6
5000	+0,5	-1,3	0,0	±2,1	±4,1
6300	-0,1	-2,0	0,0	+2,1; -2,6	±5,1
8000	-1,1	-3,0	0,0	+2,1; -3,1	±5,6
10000	-2,5	-4,4	0,0	+2,6; -3,6	+5,6; -∞
12500	-4,3	-6,2	0,0	+3,0; -6,0	+6,0; -∞
16000	-6,6	-8,5	0,0	+3,5; -17,0	+6,0; -∞
20000	-9,3	-11,2	0,0	+4,0; -∞	+6,0; -∞

a) Jmenovité kmitočty jsou z řady R10 uvedené v ISO 266:1997, tabulce 1 [5]
b) Kmitočtové funkce C a A byly vypočítány pomocí rovnic (6) a (7) s kmitočtem f vypočítaným z rovnice $f = (f_r)[10^{0.1(n-30)}]$, kde f_r a n je přirozené číslo mezi 10 a 43. Výsledky byly zaokrouhleny na desítky decibelu

Funkce *filtrA/C* filtruje vstupní signál podle váhových funkcí A a C. Vstupem této funkce je vektor akustického tlaku p_A a vzorkovací frekvence signálu. Výstupem funkce je vážený signál.

Na začátku funkce jsou proměnné F1 a F2, které obsahují vektory frekvencí uvedených v tabulce. Proměnná F1 obsahuje frekvence od 10 Hz do 1 000 KHz a proměnná F2 od 1 000 KHz do 20 000 KHz. Tyto hodnoty jsou poté přepočítány podle Shanon-Kotelnikova-Nyqvistova teoremu. Vztah:

$$f1 = \frac{F1}{\frac{fs}{2}} \quad (4.3)$$

Kde

F1 je jmenovitá frekvence (Hz)

fs je vzorkovací kmitočet (Hz)

Funkce *fir2* vyžaduje vektor frekvencí upravených podle Nyqvistova teoremu. Na začátku tohoto vektoru musí být nula a na konci jedna. Proto funkce vkládá po proměnné $f1$ na začátek nulu a na konec jednotku.

Dále funkce deklaruje proměnné A1 a A2 (pro funkci *filtrC* je to C1 a C2) do kterých vkládá úrovně decibelů z tabulky (1). Tyto hodnoty v decibelech musí funkce přepočítat na absolutní, a to delogaritmizací viz vztah:

$$mA1 = 10^{A1/20} \quad (4.4)$$

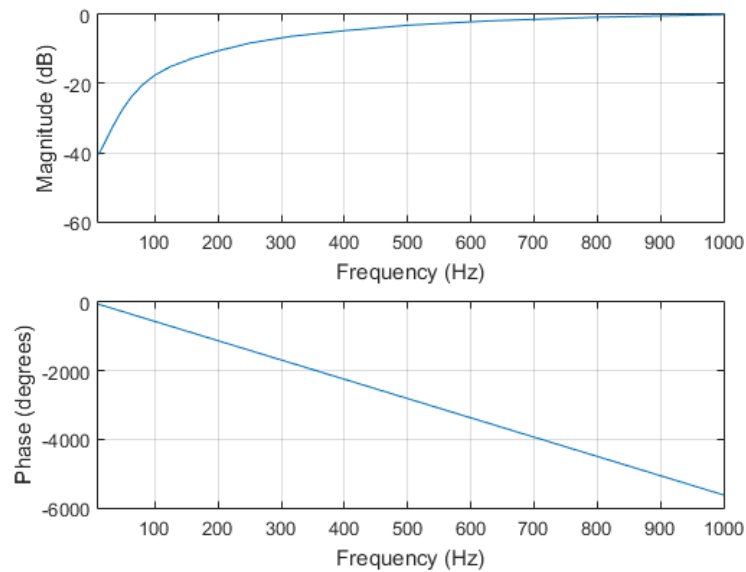
Následně funkce vytváří návrh dvou filtrů pojmenovaných FA1 a FA2 (pro funkci *filtrC* je to FC1 a FC2). Řád filtru je pro *filtrA* volen 1500 a pro *filtrC* 600. Hodnoty řádu filtrů jsem volil vysoké, kvůli větší přesnosti v nízkých kmitočtech.

Ke konci funkce je filtrován signál pomocí funkce *filter()*. Funkce aplikuje filtr FA1 na akustický tlak a uloží do proměnné a1 [6]. tento vyfiltrovaný signál je filtrován ještě

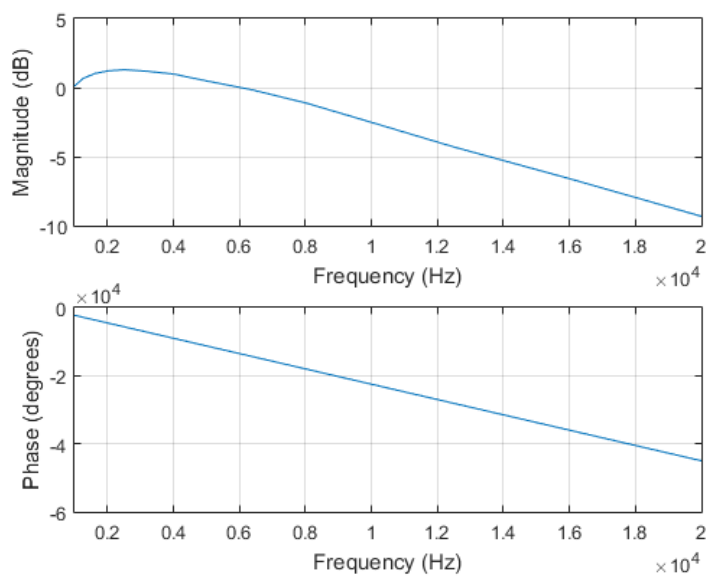
jednou filtrem *FA2* a uložen do proměnné *Vahovani*, která je výstupní proměnnou funkce *filtrA/C*.

5.4.1 Kmitočtové odezvy jednotlivých filtrů

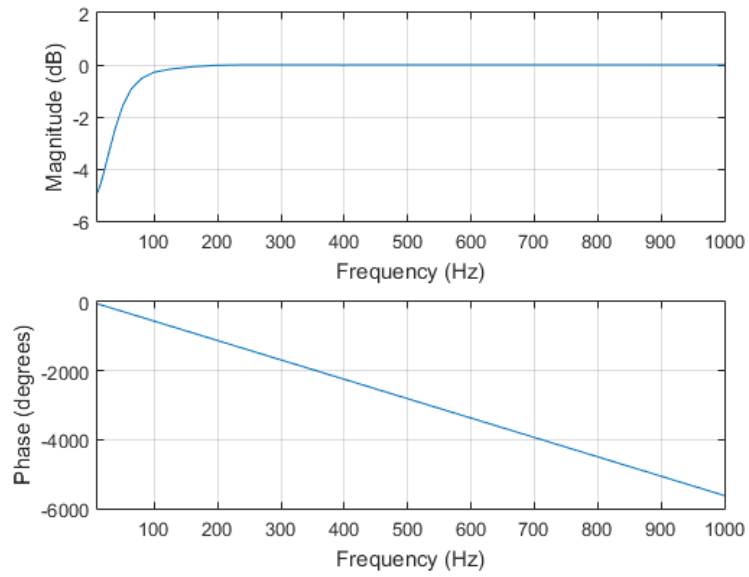
Z grafů je vidět nepřesnost u filtrů *FA1* a *FC1*. Tato nepřesnost se pohybuje až do cca 20 Hz. Ovšem pro měření hluků, které jsem prováděl v laboratořích je tato nepřesnost zanedbatelná, protože se frekvenčně nepřekrývá.



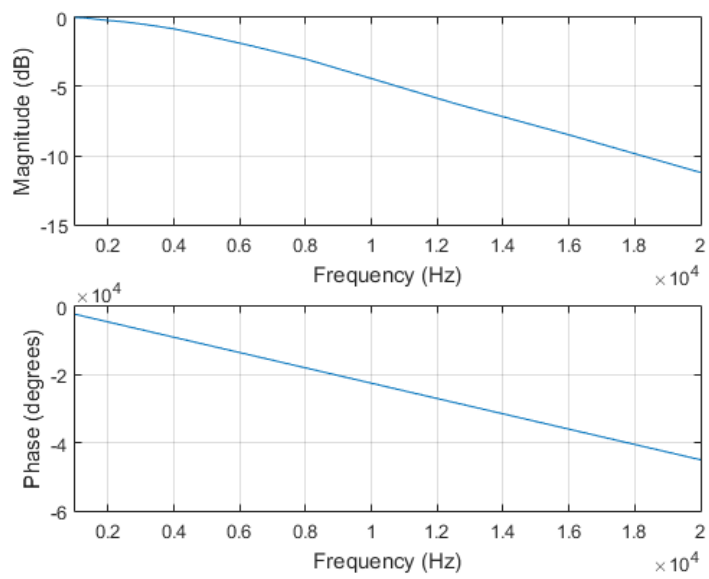
Obrázek 11 Kmitočtová odezva filtru *FA1*



Obrázek 12 Kmitočtová odezva filtru *FA2*



Obrázek 13 Kmitočtová odezva filtru FC1



Obrázek 14 Kmitočtová odezva filtru FC2

5.5 Funkce HAT

Funkce HAT vypočítává hladinu akustického tlaku. Vstupem do této funkce je časově vážený signál, jeho vzorkovací frekvence a uživatelem zvolená časová konstanta (slow, nebo fast). Výstupem této funkce se hladina akustického tlaku (dB_{SPL}).

Na začátku funkce projde vstupní proměnná P_a (akustický tlak) kvadrátem. Dále se ze vzorkovací frekvence vypočítá perioda podle vztahu:

$$T = \frac{1}{f_s} \quad (4.4)$$

Další částí funkce je integrace a dolnoproustní filtrace signálu. Zde se do proměnné v načte vztah pro dolní propust $e^{-T/k}$. Dále funkce zjistí počet vzorků daného signálu pomocí funkce `length()`[6]. Tuto délku použije v cyklu `for`, který má daný počet opakování od $t = 2$ do DPA (počet vzorků signálu). Ještě před proběhnutím cyklu funkce vyplní proměnnou „ v “ (pomocí funkce `zeros()`[6]) nulami od 1 do počtu vzorků signálů (DPA). Tento krok urychlí běh programu, program má vektor o přesné délce výstupu cyklu definovanou předem a nemusí ji definovat za chodu cyklu (velikost vektoru v na výstupu cyklu je dána počtem opakování cyklu).

V samotném cyklu funkce *HAT* řeší integraci a dolnoproustní filtraci.

```
for t=2:DPA
    v(t) = (v(t-1)*e + ((PA(t)+PA(t-1))/2) * T);
end
```

Obrázek 15 Cyklus pro výpočet integrálu a dolní propusti

Následně se výstup cyklu (vektor v) vynásobí $1/k$, kde k je proměnná časové konstanty fast, nebo slow (0,125, nebo 1) a odmocní pomocí funkce `sqrt()`[6]. Po odmocnění se vektor zlogaritmuje za použití vztahu pro výpočet hladiny akustického tlaku

$$L_{AT} = 20 \log_{10}(p_A/p_0) \quad (4.5)$$

Kde

p_A je akustický tlak (Pa)

p_0 je referenční akustický tlak (Pa)

V poslední části funkce *HAT* seřadí vektor v do matice $fs \cdot DS$, kde fs je vzorkovací frekvence a DS je délka signálu v sekundách (pomocí funkce *reshape()* [6]). Jinak řečeno funkce rozdělí vektor na matici, která v jednom sloupci obsahuje jednu sekundu signálu (v tomto případě o velikosti 48000). Nakonec pomocí funkce *max()* najde maximum z každého sloupce (sekundy) a vloží jej do proměnné La , která je jediným výstupem funkce *HAT* [6].

5.6 Funkce HZE

Funkce HZE vypočítává hladinu zvukové expozice. Vstupem této funkce je časově vážený signál, jeho vzorkovací frekvence a časový interval měření. Výstupem je hladina zvukové expozice.

Na začátku funkce HZE je kvadrátor, který umocní vstupní vážený signál.

Dále se do proměnné t uloží délka signálu v sekundách zaokrouhlená na celá čísla. Délka signálu se podělí vzorkovací frekvencí fs a výsledek se zaokrouhlí pomocí funkce *round(t)* [6].

Pro zrychlení hodu funkce deklaruje proměnnou EA jako vektor o délce 1 až t (celočíslná délka v sekundách)

V cyklu *for* funkce sčítá dva po sobě jdoucí vzorky, přičemž druhý dělí dvěma. Výsledný vektor EA funkce vynásobí převrácenou hodnotou intervalu měření a odmocní. Poté z vektoru EA vypočítá za použití upraveného vztahu na výpočet L_{AT} Hladinu zvukové expozice L_{AE} a uloží do proměnné LAE .

Nakonec funkce počítá efektivní hodnotu, pomocí cyklu *for*, a tu uloží do proměnné LAE . Obdobně, jako u funkce HAT používám funkci *reshape()*, pro rozdělení zvukových expozic do sloupců po sekundách.

5.7 Funkce IZ

Funkce IZ vypočítává intenzitu zvuku. Vstupem této funkce jsou dva vážené signály, hustota vzduchu a vzdálenost D_r mezi mikrofony. Výstupem funkce je Intenzita zvuku za předpokladu, že vlnová délka je delší nežli vzdálenost mezi mikrofony D_r . Na začátku si funkce deklaruje referenční intenzitu $I_0 = 10^{-12}$. Dále si funkce zjišťuje délku souboru v sekundách a zaokrouhluje jí dolů (kvůli funkci *reshape*, kterou na konci používá). Dále přichází cyklus for který se provádí po jednom vzorku a reprezentuje první část vztahu (3.12). Následuje další cyklus for, který reprezentuje druhou část vztahu (3.13). Po dokončení obou cyklů se jejich výstupy vynásobí a převedou do decibelů pomocí vztahu (4.6). Nakonec se výsledný vektor rozhodí do sloupců po sekundách a v každém z nich funkce IZ počítá efektivní hodnotu pomocí cyklu for. Efektivní hodnoty jednotlivých sekund jsou výstupem funkce IZ.

Vztah pro přepočítání intenzity do decibelů

$$10 \cdot \ln\left(\frac{I_r}{I_0}\right) \quad (4.6)$$

Kde

I_r je složka intenzity zvuku ve směru r (W/m^2)

I_0 je referenční intenzita zvuku (W/m^2)

5.8 Aplikace měření

Pro spuštění a správný chod programu je potřeba mít program MATLAB verze R2015 a výš, a všechny soubory skriptu a funkcí výše popisovaných v jednom souboru. Po spuštění MATLAB aplikace si jako „Current Folder“ vybereme soubor, kde výše zmíněné m-soubory máme a do Command Window zadáme příkaz „run Skript.m“, nebo si soubor Skript.m otevřeme v prostředí MATLAB a použijeme funkci Run v liště EDITOR. Pokud si zvolíme použití příkazu „run“ musíme dávat pozor na velká a malá písmena, která MATLAB odlišuje [6].

Po spuštění Skript.m se v Command Window vytiskne text:

Volte měření

Hladina akustického tlaku - 1

Hladina hlukové expozice – 2

Intenzita zvuku – 3

Zde se musí zvolit jedna ze tří možností a zadat jako číslo (1,2,3). Pokud bude zadáno jinak, vyskočí error.

Při volbě „hladina akustického tlaku“ se objeví výzva k zadání časové konstanty SLOW, nebo FAST obdobně, jako u volby měření. Zadání musí být jedna z možností. Při volbě „hladina hlukové expozice“ se objeví výzva k zadání intervalu měření v sekundách. Interval musí být zadaný číselně.

Při volbě „intenzita zvuku“ se objeví výzva k zadání hustoty vzduchu a vzdálenosti mikrofonů od sebe. V případě, že nevíme hustotu vzduchu, zadáváme „1.2“. Vzdálenost mikrofonů od sebe se udává v metrech (v případě přiložených měřených souborů je vzdálenost 0,01). Oboje hodnoty musíme zadávat s desetinou tečkou!

Při volbě měření hladiny akustického tlaku, nebo hladiny hlukové expozice se objeví dialogové okno s výzvou pro zadání souboru. Pokud je volba měření intenzity, toto okno se objeví dvakrát. wav soubor, kterým chceme měřit, buď hladinu akustického tlaku, nebo hladinu zvukové expozice, musí mít následující formát: „název_SML_Audio_FS(přepočteníkonstanta)dB(PK)_00“. Pokud si uživatel vybral měřit intenzitu zvuku, je nutné po vyskočení prvního dialogového okna vložit kalibrační soubor ve formátu: „název (úroveň v dB_{SPL})dB“ (je nutná mezera, mezi názvem a číslem). U souboru na kterým chceme měřit intenzitu zvuku, není předem stanovený formát názvu. Pokud bude soubor jiný, program nebude správně fungovat, nebo vyhodí error.

POZOR! Kalibrační soubor a soubor u kterého chceme měřit Intenzitu zvuku musí být stereofoní (dvoukanálový).

Po vložení měřeného souboru se objeví výzva pro volbu kmitočtové váhové funkce z možností A, C, nebo Z. volba je stejná jako u volby měření, či časové konstanty.

Po volbě váhové funkce má program všechna potřebná data a začne výpočet. Doba výpočtu se liší od zadaných požadavků. Po skončení výpočtu se výsledek vytiskne v Command Window a uloží do proměnné *Vysledek*

6. IMPLEMENTACE V C++

Zásadním rozdílem mezi návrhem funkcí v prostředí MATLAB a implementací funkcí do jazyka C++ je provádění v reálném čase. Pro implementaci využívám rámec jazyka C++ JUCE framework. JUCE má několik tříd zabývajících se zpracováním zvuku v reálném čase. Třída, kterou využívám se jmenuje *AudioAppComponent*. V této třídě jsou předefinované metody k získání a zpracování zvuku. (a mnoho jiných tříd). Ty jsou pak klíčové pro implementaci kódu v objektově orientovaném C++. [13]

6.1 Třída MenuComponent

Třída *MenuComponent* obsahuje uživatelské rozhraní.

Třída *MenuComponent* má vlastnosti třídy *Component*. Dále obsahuje časovou konstantu. Funkce *Resized()* je funkce kterou obsahuje každý kód vytvořený pomocí juce framework. V této funkci se nastavují hranice všech viditelných prvků. Všechny prvky jsou zviditelněny v konstruktoru ve kterém jsou zároveň volány funkce *subOptionHAT()*, *subOptionHZE()* a *subOptionIZ()*. Tyto funkce obsahují podmínosti ke každému měření obdobně jako ve skriptu „skript“ při realizaci v prostředí MATLAB. Podmínosti u měření hladiny zvukové expozice a intenzity zvuku jsou řešeny pomocí posuvníku, což umožňuje dynamicky měnit parametry těchto měření. V podmínosti pro měření hladiny akustického tlaku je definice časových konstant. Všechny tlačítka pro nastavení měření (např. Hladina akustického tlaku, fast, A) jsou rozděleny do tří skupin deklarovaných ve výčtovém datovém *enum* na: Měřicí tlačítka, tlačítka pro volbu časové konstanty a tlačítka pro výběr vážení. Poslední funkcí třídy Menu komponent je *updateToggleState()*. Tato funkce určuje změny při stisku tlačítka (např. změna barvy)

```

void MenuComponent::updateToggleState(Button* button, String name)
{
    auto state = button->getToggleState();

    String stateString = state ? "ON" : "OFF";
    String selectedString = state ? " (Vybrano)" : "";
    Logger::outputDebugString(name + " Tlacitko zmeneno na " + stateString);

    button->setButtonText(name + selectedString);
    button->setColour(TextButton::buttonOnColourId, Colours::lightgrey);
    button->setColour(TextButton::textColourOnId, Colours::black);
}

```

Obrázek 16 Funkce updateToggleState()

6.2 Main.cpp

Main.cpp je soubor vytvořen JUCE v projuceru. Obsahuje dvě třídy. Třidu *MainWindow*, která vlastní kontent vytvoření v *MainComponent* třídě a vytváří okno aplikace s vlastnostmi třídy *MainComponent*. Zajišťuje responzivnost. Responzivnost je nastavení velikosti okna aplikace při spuštění aplikace na různých platformách a zařízeních. Druhou třídou v souboru Main.cpp je v tomto případě *LoudnessMMAApplication*. Ta má vlastnosti třídy *JUCEApplication*, která zajišťuje spuštění, chod a destrukci aplikace.

6.3 Třída MainComponent

MainComponent vlastní všechny funkce (potažmo třídy) třídy *AudioAppComponent*. Nejdůležitějšími funkcemi jsou *prepareToPlay* a *getNextAudioBlock*, díky kterým aplikace získává data, které jsou dále zpracovávány. Další důležitou funkcí je funkce *paint()* díky které funkce vykresluje naměřená data. Ve třídě *MainComponent* jsou realizovaný výpočty kalibrace a hladiny akustického tlaku.

6.3.1 Důležité nastavení v konstruktoru

V konstruktoru třídy *MainComponent* se nastavuje viditelnost *MenuComponent*, nastavení přepisovatelnosti, a frekvence Timeru (blíže popsaná v kapitole *timerCallBack*). Krom nastavení velikosti (po zapnutí aplikace) je posledním nastavím v konstruktoru počet vstupních a výstupních kanálů. Toto nastavení je opatřené pro případ, že zařízení, které je připojeno vyžaduje oprávnění. Nastavení tolika vstupních (potažmo i výstupních) kanálů, které potřebujeme sníží možnost problémů při získávání dat ze vstupně výstupního zařízení.

Destruktor obsahuje pouze funkci *shutDownAudio()*

```
10  MainComponent::MainComponent(): window(ADSize, dsp::WindowingFunction<float>::hann)
11  {
12      // Menu Component displays UI elements
13      addAndMakeVisible(menuComponent);
14      setOpaque(true);
15      // How often is called function timerCallBack
16      startTimerHz(30);
17
18      // Size of canvas
19      setSize(1000, 900);
20
21
22      if (RuntimePermissions::isRequired(RuntimePermissions::recordAudio)
23          && !RuntimePermissions::isGranted(RuntimePermissions::recordAudio))
24      {
25          RuntimePermissions::request(RuntimePermissions::recordAudio,
26                                     [&](bool granted) { if (granted) setAudioChannels(2, 0); });
27      }
28      else
29      {
30          setAudioChannels(2, 0);
31      }
32  }
33
```

Obrázek 17 Konstruktor MainComponentu

6.3.2 Funkce prepareToPlay

Tato funkce má na svém vstupu očekávaný počet prvků v jednom audio bloku a vzorkovací frekvenci. V této funkci, pomocí globální proměnný *lastSampleRate* je

sdílen s ostatními funkcemi vzorkovací kmitočet příchozích audio dat. V případě implementace váhových filtrů by se většina parametrů definovala právě tady.[13]

6.3.3 Funkce getNextAudioBlock

Vstupními argumenty této funkce jsou zdrojové informace z kanálů (popřípadě z kanálu), vstupně/výstupního zařízení. V této funkci získáváme všechna audio data. Funkce proběhne pouze, když je tlačítko „Merit“, Status kalibrace je „true“ (kalibrace proběhla) a počet vstupních kanálů je větší než 0. Pokud jsou podmínky splněné, tak se vytvoří proměnná *channelData* do které se pomocí funkce *getWritePointer* deklarují ukazatele příchozího bufferu. Funkce *getWritePoiter* kvůli rychlosti nekontroluje případy, ve kterých jsou data mimo předpokládaný rozsah, v případě, že jsou data očividně poškozená (z pohledu rozsahu) aplikace nezjistí a mohou tak vznikat chyby měření. Dále se po jednom vzorku volá funkce *pushNextSampleIntoFifo*, která po každém celém bufferu vynuluje pole *audioData* a pomocí globální proměnný povolí další načítání dat do bufferu. [13]

```
227 void MainComponent::pushNextSampleIntoFifo(float sample) noexcept
228 {
229     // If Fifo containt amount of data for make next block ready.
230     if (fifoIndex == ADSize)
231     {
232         // Set next buffer ready
233         if (!nextAudioBlockReady)
234         {
235             zeromem(audioData, sizeof(audioData));
236             memcpy(audioData, fifo, sizeof(fifo));
237             nextAudioBlockReady = true;
238         }
239         fifoIndex = 0;
240     }
241     fifo[fifoIndex++] = sample;
242 }
243
244
245
```

Obrázek 18 Funkce pushNectSampleIntoFifo

6.3.4 Funkce `timerCallback`

Ve funkci `timerCallback` se realizuje většina výpočtů pro určení hladin. Dále se je zde implementovaná funkce kalibrace. V této funkci je předpřipravené části pro realizaci měření hladiny zvukové expozice.

6.3.4.1 Realizace kalibrace

Kalibrace se spustí v tu chvíli, kdy je stlačené tlačítko kalibrace. Deklaruje se proměnná `poweredDataC`, která bude mít velikost bufferu. Následuje cyklus `for`, který po vzorcích `audioData` ze vstupního zařízení (nejčastěji zvuková karta) umocní a umocněná data sečte. Potom se mimo tělo cyklu sečtená data podělí velikostí bufferu v proměnné `calibrationData`. Dále funkce počítá desetinásobek dekadického logaritmu z proměnné `calibrationData` a tím je přepočítá na jednotky `dBFS`. Kalibrace počítá s tím, že akustický kalibrátor bude produkovat hladinu akustického tlaku o hodnotě 100 `dB SPL`, proto se proměnná `Z` definuje jako rozdíl mezi `stem` a přepočítanou hodnotou proměnné `DbFS`. V definici proměnné `calibrationConstant` je podíl proměnné `Z` ku 20, čímž se realizuje část vztahu pro přepočet z návrhu. Proměnná `transferConstant` poté vztah dokončí a definuje se tedy jako `PO` (globální proměnná o hodnotě referenčního akustického tlaku) krát deset umocněné proměnnou `calibrationConstant`. Na začátku každé realizace kódu potom musí být vstupní audio data vynásobeny touto konstantou, která je uložena jako globální proměnná. Kalibrace se ukončí po odkliknutí tlačítka kalibrace.

6.3.4.2 Realizace výpočtu hladiny akustického tlaku

Hladina akustického tlaku je realizovaná v těle podmínky `if` (tak by měli být realizované všechny výpočty). Ta kontroluje, jestli je zvolena možnost „měření hladiny akustického tlaku“ a možnost „merit“ obdobně jako kalibrace ověřuje stlačení tlačítka kalibrace. Toto ověření je realizováno pomocí funkce `getToggleState()`. Díky rozdělení podmínek ve funkci `timerCallback()` je možno při měření přepínat mezi jednotlivými výpočty. Na začátku těla podmínky je cyklus `for`, který přepočítává audio data na kalibrované data pomocí globální proměnné `transferConstant`. Dále následuje definice několika proměnných obdobně, jako v návrhu v prostředí `MATLAB`. Je zde definovaná perioda, jakožto převrácená hodnota vzorkovací frekvence získané pomocí globální proměnné `lastSampleRate`, která je definovaná v těle funkce `preapereToPlay`. Další proměnná `lowPass` je definovaná jako Eulerovo konstanta mocněná záporným podílem periody ku časové konstantě získané ze třídy

MenuComponent. Díky tomu je možno dynamicky měnit časové konstanty v průběhu měření. Následuje kvadrátor realizovaný pomocí cyklu `for`. Poté se definuje první vzorek obdobně jako v návrhu. Následuje cyklus `for`, který se opakuje po vzorcích do velikosti bufferu. V těle cyklu jsou deklarované a definované tři proměnné a to *level*, *lastLevel* a *trueLastLevel*. Kde v proměnné *level* je aplikovaný výpočet z návrhu funkce s dolnopropustní filtrací. Proměnná *lastLevel* je potom odmocnina z převrácené hodnoty časové konstanty vynásobené proměnnou *level*. Proměnná *trueLastLevel* vypočítá hladinu akustického tlaku vztaženého k referenční hodnotě akustického tlaku. Výsledek se uloží do globální proměnné *scopeData1*.

6.3.5 Funkce DrawAudioText

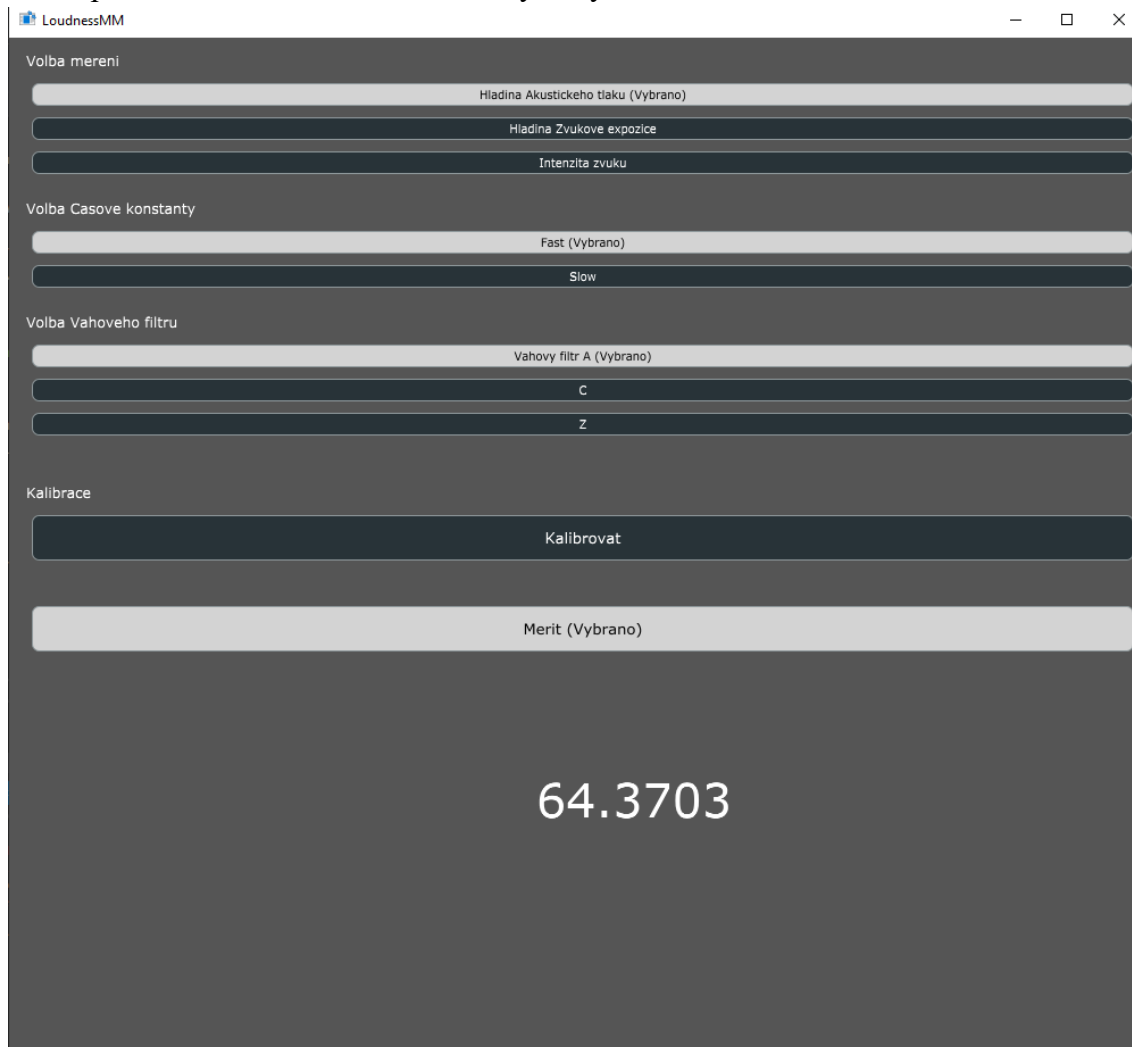
Funkce *drawAudioText()* má za úkol tisknout data do uživatelského rozhraní. Využívá pro to třídu *JUCE::Graphic*, která má předefinované funkce pro vykreslování. Na svém začátku funkce pomocí podmínky `if` kontroluje zda je stlačené tlačítko „Merit“ a zároveň jestli není stlačené tlačítko „kalibrovat“, je totiž zapotřebí aby ve chvíli, kdy se tisknou data, neprobíhala kalibrace, jelikož by se hodnoty proměnné *calibrationTransfer* neustále přepisovali a došlo by ke znehodnocení měření. Pokud je podmínka dodržena, zjišťuje se, zda se jedná, či nejedná o měření hladiny zvukové expozice, u které je zapotřebí tisknout hodnoty po sekundách. Pokud tomu tak není, tak se pomocí cyklu `for` hledá největší hodnota z jednoho bufferu měření (obdobně, jako u návrhu). Potom, co se hodnota najde, volají se dvě funkce grafické třídy *JUCE*: *g.setFont()* (která nastaví velikost textu na 50 pixelů) a *g.drawText*, která tiskne hodnoty do uživatelského rozhraní. Argumenty funkce *drawText* je tisknutá hodnota (která musí být převedena na datový typ `string` pomocí funkce *JUCE::String()*) a pozice ve které se má daný text vytisknout do uživatelského rozhraní. Na konci se nastavuje proměnná *result* (která obsahuje výsledek výpočtu) na nulu, aby nedocházelo k chybám.

6.3.5.1 Funkce Paint

Funkce *paint*, je předefinovaná funkce, která vykresluje veškeré grafické obrazce. Také se zde nastavuje barva pozadí pomocí funkce *g.fillAll()*. V této funkci se volá funkce *drawAudioData()*, která vypisuje naměřené hodnoty. Funkce *paint* se nesmí volat, proto se v těle funkce *timerCallback()* volá funkce *repaint()*, která znovu zavolá funkci *paint()*.

7. PRŮVOD MĚŘENÍM APLIKACÍ HLUKOMĚRU

Je potřeba dodržet několik zásad, aby nebylo měření znehodnocené.



Obrázek 19 Ukázka uživatelského rozhraní

7.1 Volba měření a jeho parametrů

Po zapnutí aplikace se otevře okno s uživatelským rozhraním, ve kterém je několik tlačítek a posuvníků, kterými se nastavují jednotlivé parametry měření. Jako první je zapotřebí si vybrat měření, které chceme provádět. Po volbě měření se nám ukážou podmožnosti měření, které se s volbou mění. Když zadáme parametry měření můžeme kalibrovat. Uživatelské rozhraní obsahuje ještě tlačítka pro volbu kmitočtových váhových filtrů, ty jsou ovšem nefunkční, tudíž jakákoliv volba filtrace neovlivní výsledek. Výsledek bude vždy vážen filtrem typu Z (žádným filtrem).

7.2 Kalibrace

Kalibrace probíhá pomocí akustického kalibrátoru. Je zapotřebí, aby kalibrátor generoval hladinu akustického tlaku 100 dB_{SPL}, s tím totiž proces kalibrace počítá. Kalibrace začne se stisknutím tlačítka „Kalibrovat“ je tedy zapotřebí kalibrátor přiložit k mikrofonu před stlačení tohoto tlačítka. Kalibrace končí odkliknutím tlačítka kalibrace (měl by zmizet nápis „(vybrano)“), je tedy nutné ukončit kalibraci před tím, než kalibrátor přestane generovat signál.

7.3 Měření

Měření je možné spustit jen když je možnost kalibrace odkliknutá (není možné měřit a kalibrovat zároveň). V případě že není vybráno měření a jeho parametry bude měření neprůkazné.

8. ZÁVĚR

Ve své závěrečné práci jsem se zabíral problematikou hluků a sluchového orgánu. Dále jsem rozebral problematiku měření hluků podle normy ČSN EN 61672-1. V prostředí MATLAB jsem vytvořil soubor funkcí, který z vložených souborů a zvolených časových konstant, intervalů měření a kmitočtových váhových filtrů vypočítá hladinu akustického tlaku, hladinu zvukové expozice, nebo intenzitu zvuku. U kmitočtových filtrů jsem porovnal kmitočtovou odezvu s hodnotami z normy ČSN EN 61672-1 (viz tabulka 1) a filtry FA1 a FC1 se na nízkých kmitočtech do 25 Hz odchyľují až o cca 35 decibelů u filtru FA1 a o cca 9 decibelů. Tuto odchylku si zdůvodňuji nevhodným použitím funkce pro návrh filtru. Na místo filtru typu FIR (filtr s konečnou kmitočtovou odezvou), by byl vhodnější filtr typu IIR (filtr s nekonečnou kmitočtovou odezvou). Hladina akustického tlaku a hladina zvukové expozice je srovnána s hodnotami naměřenými digitálním hlukoměrem NTi AL1. Průměrná relativní odchylka od hodnot u hladiny akustického tlaku je 0,24 % (pro L_{ZF}) a u hladiny zvukové expozice 2,4 % (pro L_{Ceq}). U měření intenzity zvuku nemám k dispozici referenční hodnoty, tudíž nemohu hodnoty srovnat. Ovšem hodnota intenzity drážkovačky přibližně 132 dB_{SPL} není nějak extrémně vzdálená od předpokládané hodnoty. Dále funkce IZ nepočítá s možností, že vlnová délka λ je delší než vzdálenost mikrofonů od sebe Δ_r vzdálenost mikrofonů od sebe.

Při implementaci v jazyce C++ jsem se seznámil s programovacím jazykem C++ a jeho rámec JUCE framework. A studoval jsem problematiku zpracování zvuku v reálném čase. Vytvořil jsem uživatelské prostředí pro softwarový hlukoměr a realizoval výpočty kalibrace pro akustické kalibrátory při akustickém tlaku 100 dB_{SPL}. Pokusil jsem se o implementaci výpočtů zvukové expozice a intenzity zvuku. Implementaci měření jsem ověřil pouze relativně, kdy jsem simuloval kalibraci velkým hlukem a poté se pokoušel měřit hladinu akustického tlaku. Hodnoty této hladiny se pohybovali od 0 dB_{SPL} do cca 140 dB_{SPL}. Tyto hodnoty jsou u akustického tlaku poměrně očekávané, vezmeme-li v potaz, že simulace kalibrace pravděpodobně nedosáhla 100 dB_{SPL}, tudíž nižší hodnoty hladiny akustického tlaku hlukoměr interpretuje jako vysoké.

Literatura

- [1] ČSN EN 61672-1: Elektroakustika – Zvukoměry – Část 1: Technické požadavky. Česká technická norma, Český normalizační institut, listopad 2003.
- [2] SCHIMMEL, Jiří. Elektroakustika. Brno: Vysoké učení technické v Brně Fakulta elektrotechniky a komunikačních technologií Ústav telekomunikací Purkyňova 118, 612 00 Brno, 2016. ISBN 978-80-214-4716-5.
- [3] SMETANA, C. a kol. *Hluk a vibrace, měření a hodnocení*. Sdělovací technika, Praha 1998. ISBN: 80-901936-2-5
- [4] MIŠUREC, Jiří a Zdeněk SMÉKAL. Číslicové zpracování signálů. Brno: Vysoké učení technické v Brně Fakulta elektrotechniky a komunikačních technologií Ústav telekomunikací Purkyňova 118, 612 00 Brno, 2011.
- [5] VAŠINA, M. a kol. Environmentální fyzika - studijní texty. Zlín: UTB, 2010 [online]. Dostupné z URL: <http://ufmi.ft.utb.cz/texty/env_fyzika>[cit. 13. 12. 2018].
- [6] *MathWorks* [online]. Natick, Massachusetts, USA: Mathworks, 1994- [cit. 2018-12-13]. Dostupné z: <https://www.mathworks.com/>
- [7] HLOUŠEK, Pavel Měření indexu hlasitosti: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2012. 60 s. Vedoucí práce byl Ing. David Kurc
- [8] KŘÍŽ, Petr Software pro zpracování dat hlukoměru: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. 61 s. Vedoucí práce byl Ing. David Kurc
- [9] SYSEL, Petr a Zdeněk SMÉKAL. *Číslicové filtry* [online]. Brno, 2012 [cit. 2020-05-11]. Skript. Vysoké učení technické v Brně Fakulta elektrotechniky a komunikačních technologií Ústav telekomunikací Purkyňova.
- [10] *Design of digital filters for frequency weightings (A and C) required for risk assessments of workers exposed to noise* [online]. 10.2486/indhealth.2013-0003, 2014 [cit. 2020-04-08]. Dostupné z: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4331191/>

- [11] List (abstract data type). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2020-06-01]. Dostupné z: [https://en.wikipedia.org/wiki/List_\(abstract_data_type\)](https://en.wikipedia.org/wiki/List_(abstract_data_type))
- [12] <https://stackoverflow.com/> [online]. [cit. 2020-05-24]. Dostupné z: <https://stackoverflow.com/>
- [13] *JUCE* [online]. [cit. 2020-05-08]. Dostupné z: <https://juce.com/>
- [14] JAROŠ, Karel. Skripta pro zvukovou tvorbu. Písek: FAMO.

Seznam příloh

Příloha 1 - Tabulka naměřených hodnot.....	59
--------------------------------------------	----

Příloha 1 - Tabulka naměřených hodnot

Tabulka 2 Deset sekund pro L_{AFmax}

Program [dB]	XL2 NTI [dB]	absolutní odchylka [dB]	relativní odchylka [%]
79,3	79,3	0,00040	0,040
79,5	79,5	0,00028	0,028
79,8	79,8	0,00018	0,018
79,9	79,9	0,00025	0,025
79,5	79,6	0,00080	0,080
79,6	79,6	0,00051	0,051
79,7	79,7	0,00038	0,038
80,0	80	0,00021	0,021
80,0	80	0,00016	0,016
79,7	79,7	0,00001	0,001
79,8	79,8	0,00046	0,046

Tabulka 3 Deset sekund pro L_{CSmax}

Program [dB]	XL2 NTI [dB]	absolutní odchylka [dB]	relativní odchylka [%]
77,8	77,8	0,00016	0,016
77,7	77,7	0,00003	0,003
77,9	77,9	0,00055	0,055
78,0	78	0,00010	0,010
78,0	78	0,00051	0,051
77,9	77,9	0,00022	0,022
77,9	77,9	0,00010	0,010
77,9	77,9	0,00026	0,026
78,0	78	0,00041	0,041
78,0	78	0,00016	0,016
77,9	77,9	0,00002	0,002

Tabulka 4 Deset sekund pro L_{ZSmax}

Program [dB]	XL2 NTI [dB]	absolutní odchylka [dB]	relativní odchylka [%]
79,2	79,2	0,00018	0,018
79,1	79,1	0,00049	0,049
79,3	79,3	0,00030	0,030
79,4	79,4	0,00007	0,007
79,4	79,4	0,00021	0,021
79,3	79,3	0,00040	0,040
79,3	79,3	0,00008	0,008
79,3	79,3	0,00034	0,034
79,4	79,4	0,00059	0,059
79,4	79,4	0,00032	0,032
79,3	79,3	0,00046	0,046

Tabulka 5 Deset sekund pro L_{CIeq}

Program [dB]	XL2 NTI [dB]	absolutní odchylka [dB]	relativní odchylka [%]
78,7	80,6	0,02414	2,414
79,0	81	0,02529	2,529
78,9	80,9	0,02481	2,481
78,7	80,6	0,02374	2,374
79,0	80,9	0,02389	2,389
78,8	80,8	0,02487	2,487
78,7	80,6	0,02365	2,365
78,7	80,7	0,02438	2,438
79,0	81	0,02466	2,466
78,7	80,6	0,02414	2,414
79,4	81,4	0,02480	2,480