



**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

---

**Fakulta elektrotechnická  
Katedra radioelektroniky**

**Měření parametrů analogového záznamu**

**Measurement of analog recording parameters**

Bakalářská práce

Studijní program: Komunikace, multimédia a elektronika  
Studijní obor: Multimediální technika

Vedoucí práce: Ing. František Rund, Ph.D.

**Kritskii Vladislav**

---

**Praha 2016**



Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze 27.05.201



České vysoké učení technické v Praze  
Fakulta elektrotechnická  
katedra radioelektroniky

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Vladislav Kritskii**  
Studijní program: Komunikace, multimédia a elektronika  
Obor: Multimediální technika  
Název tématu: **Měření parametrů analogového záznamu**

### Pokyny pro vypracování:

Seznamte se se základními vlastnostmi analogového záznamu a definicemi sledovaných parametrů (kmitočtová charakteristika, THD, SNR, ...). Připravte metodiku pro získání parametrů analogového záznamu z digitální kopie měřicího média. Výstupem práce bude program implementovaný v prostředí MATLAB, který umožní měřit parametry analogového záznamu z digitalizovaného měřicího média (např. digitalizované měřicí pásy). Výsledky porovnejte s výsledky získanými jinou metodou.

### Seznam odborné literatury:

- [1] Bašta, I.: Zpracování a záznam signálu. Praha: ČVUT. 1999.
- [2] Bohn, D.: Audio Specifications [online]. Last revision January 2003 [cit. 2014-10-23]. <http://www.rane.com/note145.html>.

Vedoucí: Ing. František Rund, Ph.D.  
Platnost zadání: do konce letního semestru 2015/2016

L.S.

doc. Mgr. Petr Páta, Ph.D.

prof. Ing. Pavel Ripka, CSc.

vedoucí katedry

děkan

V Praze dne 10. 2. 2015



## Abstrakt.

V této práci se zaměřím na seznámení s analogovými záznamy a jejich vlastnostmi. V programu Matlab bude provedena analýza těchto záznamů, která bude následně vypovídat o úrovni šumu v záznamech. Dále bude analyzováno spektrum signálu, z čehož bude možné určit harmonické signálů a spočítat THD a THD+N. Takže bude spočítán SNR, a to dvěma metodami – ze spektra signálů a v časové rovině. V závěru práce bude také pojednáno o kmitočtové charakteristice, a to kreslením spektra v kmitočtové rovině.

## Abstract.

In this work, I will take a look at the analog recordings and their basic properties. In Matlab software, I will make analysis of these recordings, after which would be shown the level of noise of the signals. Spectrum of the signals will also be found, as it would be used to find harmonics and to calculate THD and THD + N. In addition, signal-to-noise ratio (SNR) will be calculated using two methods – with signal's spectrum and in the time dimension. At the end of the work there would be demonstrated frequency characteristic by plotting spectrum of signal on frequency axes.

## Klíčová slova:

Matlab, THD, Odstup signál-šum, Analogový záznam zvuku, Spektrum, A-filtr

## Key words:

Matlab, Total Harmonic distortion, Signal-to-noise ratio, Analog recording, Spectrum, A-weighted filter



# Obsah

Česné prohlášení	2
Zadání Bakalářské Práce	3
Úvod	6
Abstrakt	4
<b>1. Teoretický úvod</b>	<b>7</b>
<b>1.1 Analogový signál</b>	<b>7</b>
1.1.1 Klasifikace signálů	7
1.1.2 Základní vlastnosti analogového signálu	7
1.1.3 Klasifikace analogových záznamů	8
<b>1.2 Šum a jeho vlastnosti</b>	<b>8</b>
1.2.1 Definování šumu	8
1.2.2 Odstup signál/šum(SNR)	10
1.2.3 Kvantizační šum	11
<b>1.3 Zkreslení signálu</b>	<b>11</b>
1.3.1 Total Harmonic Distortion(THD)	11
1.3.2 Total Harmonic Distortion + Noise (THD+N)	12
<b>2. Praktická část</b>	<b>13</b>
<b>2.1 Výpočet parametrů záznamu. Funkce eval_params</b>	<b>13</b>
2.1.1 Čtení záznamu	13
2.1.2 Spektrum signálu	15
2.1.3 Hledání harmonických signálu	15
2.1.4 A-filtr (A - weighting filter)	16
2.1.5 Hledání hlavní harmonické	17
2.1.6 Výpočet THD	17
2.1.7 Výpočet THD+N	18
2.1.8 Výpočet SNR	20
2.1.9. Ukončení výpočtů. Ukládání výsledků	22
<b>2.2 Návrh uživatelského rozhraní(interface)</b>	<b>23</b>
2.2.1 Funkce main	23
2.2.2 Funkce button_callback	25
2.2.3 Funkce button_callback2	28
2.2.4 Funkce button_callback3	28
<b>2.3 Výsledky měření</b>	<b>32</b>
2.3.1 Popis určitých záznamů	32
2.3.2 Hodnoty získané v Matlab	33
2.3.3 Porovnání s výsledky z SigView	34
<b>3. Závěr</b>	<b>35</b>
<b>4. Seznám použité literatury</b>	<b>36</b>



## Úvod.

Ruční měření parametrů nahrávek z minulého století může být obtížným a zdlouhavým procesem. Příčinou mohou být těžko detekovatelné hodnoty (například kvůli schopnostem měřicího přístroje) nebo nutnost velkého počtu měření, což je časově náročné. V současnosti existují metody digitalizace analogových záznamů, které umožňují převést záznam do paměti počítače ve tvaru čísel. Zdlouhavé a obtížné měření se tedy dá provést pomocí počítače.

Při měření parametrů a hodnocení kvality nahrávky je nejdůležitější samotný signál a jeho vlastnosti. Pod pojmem signál si můžeme představit nekonečnou množinu definic. Hlavní věcí spojující všechny definice je to, že signál je v podstatě přenášení jakékoliv informace. Abstraktně, signál (z latiny *signum* - znak) je nějaký jev nebo fyzický proces, který v sobě nese zprávu o nějakém ději, stavu objektu nebo řídicí příkaz. Struktura signálu může být implementována pomocí různých prostředí: například u rádia a televize je signál elektromagnetickou vlnou, obrázek je prostorovou strukturou intenzity různých barev, video a obrázky jsou v digitální formě vyjádřeny bitovým tokem.

Na to, aby člověk nebo počítač byl schopný rozumět zprávě směřující k němu, je potřeba, aby samotný signál byl srozumitelný, tj. kvalitní. Ve své práci se podívám na kvalitu určitých analogových záznamů a jejich vlastností. V teoretickém úvodu vysvětlíme pojem šum, a parametry s ním související – THD, THD+N a SNR, na čem závisí a způsoby jejich výpočtu. V praktické části tyto parametry budou implementovány v programu Matlab, kde se bude provádět celé měření. Bude provedena analýza úrovní šumů záznamů a porovnání s úrovní užitečné části signálů. Cílem mé práce je odhadnout parametry určených analogových nahrávek, analyzovat výstupní data a porovnat je s výsledky získanými jinou metodou. Konečným cílem práce je návrh kódu v Matlab, který by umožnil měření následujících parametrů analogových záznamů: THD, THD+N, SNR a vliv váhového A-filtru na tyto parametry.



# 1. Teoretický úvod.

## 1.1 Analogový signál.

### 1.1.1 Klasifikace signálu.

V radiotechnice je signálem časová funkce  $s(t)$ , která popisuje změnu napětí a proudu. Když signál prochází obvodem, na výstupu se kromě užitečné části přičte “nežádoucí” (ve většině případů) část, kterou dále budeme zvat šum.

V radiotechnice se typy signálů dělí na 4 hlavní skupiny [18]:

- a) Impulzní – kolísání jen v časově omezeném intervalu.
- b) Analogové – kontinuální v čase funkce.
- c) Diskrétní – v určitém časovém intervalu je přesný počet diskretizačních vzorků, které navrhují signál
- d) Digitální – jeden z typů diskrétního signálu, kde je signál představen ve tvaru čísel.

### 1.1.2 Základní vlastnosti analogového signálu.

Podle definice je analogový signál kontinuální funkcí, která je spojitá v čase a amplituda takového signálu může teoreticky mít libovolné hodnoty v určitém časovém intervalu [18]. Mezi základní parametry analogového signálu patří amplituda, kmitočet a fáze. Chování prostého analogového signálu se dá popsat pomocí funkce [19] :

$$s(t) = A \cdot \sin(\omega t + \phi). \quad (1)$$

Kde  $A$  je amplituda signálu,  $\omega$  je úhlová frekvence ( $\omega = 2\pi/f$  [19]),  $\phi$  je fáze.

V praxi se ale většinou budeme setkávat se složitějšími analogovými signály, které v sobě mají více než jednu sinusovou funkci. Pokud je signál periodický s periodou  $T$ , dá se takový signál popsat pomocí koeficientů Fourierovy řady [20]:

$$s(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cdot \cos k\omega_1 t + b_k \cdot \sin k\omega_1 t) \quad (2)$$

Kde  $\omega$  je úhlová frekvence ( $\omega = 2\pi/T$  [19]),  $k$  je číslo harmonické,  $a_k$  a  $b_k$  jsou koeficienty Fourierovy řady.

Důležitým parametrem signálů je taky jeho kmitočtová charakteristika. Kmitočtová charakteristika je změnou koeficientu zesílení neboli výstupního signálu zesilovače v závislosti na kmitočtu. Alternativně se dá říct, že kmitočtová charakteristika je změna amplitudy signálu v závislosti na kmitočtu, většinou zobrazená v logaritmickém měřítku [28].



### 1.1.3 Klasifikace analogových záznamů.

S postupujícím technickým rozvojem se lidé pokoušeli natáčet zvuk a během 20 století se objevili 3 hlavní typy natáčení analogového zvuku:

1. Mechanické natáčení. Podle názvu je pochopitelné, že natáčení probíhá mechanickým způsobem – zvuk se zaznamenává potlačením jehly na povrch materiálu (například vinylu) do hloubky nebo šířky podle změny vzduchového tlaku (což je zvuk), nebo pomocí elektrického zesilovače. Signál takového záznamu byl často doprovázen šumem, jeho hodnota SNR je většinou v rozsahu 60-70 dB [21],[22].
2. Optické natáčení. Takovýto typ záznamu se objevil poprvé ve filmech. Natáčení se provádí pomocí žárovky s konstantní intenzitou. Světlo dopadá přes masku na štěrbinu tak, že osvětlená část štěrbinu odpovídá okamžité hodnotě akustického tlaku potřebného signálu zvuku. Silné zkreslení u takových záznamů se objevovalo při kopírování [23].
3. Magnetické natáčení. Natáčení je realizováno magnetováním pásky tak, že se do povrchu materiálu pomocí magnetické hlavy převádí elektrický signál se správnou ve tvaru magnetického pole, pak na pásce zůstává tzv. magnetický sled, který je záznamem [24]. Úroveň šumu v takových záznamech velmi závisí na použitém materiálu a součástkách uvnitř natáčejícího obvodu [25].

Kromě mechanických šumů, které se můžou objevit při reprodukce takových záznamů, existují taky vnitřní zdroje šumů, neboli šumy součástek uvnitř obvodů[27].

## 1.2 Šum a jeho vlastnosti.

### 1.2.1 Definování šumu.

Šum je nežádoucí chování elektrického signálu v obvodu. Alternativní definicí je jakýkoliv elektrický signál v obvodu, odlišný od užitečného. Je důležité rozdělovat pojmy zkreslení a šum, protože neznamenají to samé[26].

Zdroje šumů radíme do 3 hlavních skupin:

1. Přírodní šumy - blesky a výbuchy sluneční aktivity atd.
2. Umělé zdroje šumů – motory, přepínače, stroje atd.
3. Vnitřní zdroje šumů – vznikají uvnitř fyzických systémů v důsledku fluktuací. Příkladem mohou být výstřelový nebo tepelný šum[27].

Při práci se signálem se bude jednat o šumy uvnitř obvodu, tj. 3 skupina šumů. V elektrotechnice má šum tvar kolísání (fluktuace) napětí nebo proudu na svorkách elektrických zařízení. Příčinou vzniku takového šumu je náhodné chování elektrických nosičů uvnitř součástek obvodu na mikroskopické úrovni. Ve většině případů je šum v obvodech nežádoucím jevem, s výjimkou případů, kdy se šum používá pro měření. Jako příklad nežádoucího jevu může sloužit nízkosumový frekvenční zesilovač, ve kterém minimální detekovatelný výkon závisí především na úrovni šumů vstupní kaskády[11].





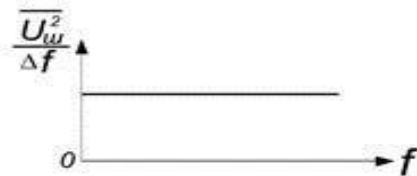
Nejrozšířenějšími typy šumu uvnitř obvodů jsou tepelný a výstřelový [11],[27].

### Tepelný šum.

Příčinou vzniku tepelného šumu (takže se nazývá Johnson-Nyquist šum) je chaotický pohyb nosičů (velmi podobný Brownovu pohybu) náboje (elektrony a díry) v odporovém materiálu. To vede k tomu, že v objemu vodiče vznikají náhodné změny hustoty elektrického náboje, přitom na koncích vodiče budou chaoticky měnit se rozdíly potenciálů. Tepelný šum byl objasněn v roce 1928 J. Johnsonem. Ve stejném roce Nyquist popsal tento jev vztahem [11],[27],[30]:

$$\overline{U_w^2} = 4kTRB \quad (3)$$

kde,  $k$  je Boltzmannova konstanta,  $R$  je odpor uvnitř obvodu v Ohmech,  $T$  je absolutní teplota odporu,  $B$  je efektivní šířka pásma, ve kterém se měří tepelný šum. Z tohoto vztahu se dá vyjádřit to, že spektrální hustota tepelného šumu (tj. výkon na jednotkový kmitočtový interval), nezávisí na frekvenci (obrázek č. 1).



Obrázek č. 1. Spektrální hustota tepelného šumu (převzato z [11]).

Spektrální hustota tepelného šumu popsána vztahem [11]

$$S_f = 4kTR \quad (4)$$

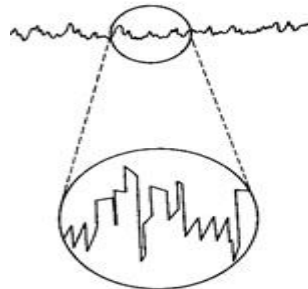
### Výstřelový šum.

Vznik výstřelového šumu souvisí s průchodem proudu přes bariéru. Jako příklad procesu vzniku může sloužit vakuová dioda ve které se elektrony imitují z katody náhodně, a pak se ovlivněné elektrickým polem pohybují k anodě. Proud vytvořený z těchto elektronů má chaotické fluktuace v okolí střední úrovně, fluktuace vzniká kvůli náhodnému vzniku disktrétní emise. Spektrální hustota výstřelového šumu je popsána vztahem [11],[31]:

$$S_f(\omega) = 2qI \quad (5)$$

Kde  $q$  je náboj elektronu,  $I$  je střední hodnota proudu.

Příčina vzniku těchto dvou typů šumů se od sebe liší, ale zároveň se dá mluvit o podobnosti jejich struktur.



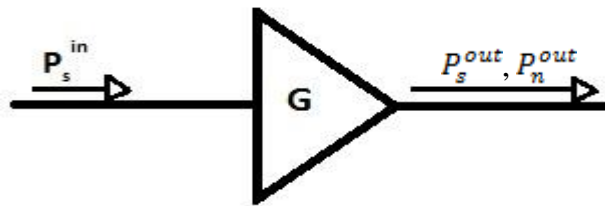
Obrázek č. 2. Příklad šumového signálu (převzato z [11]).



Struktura obou signálů má tvar posloupnosti náhodných impulzů, které mají podobnou formu a náhodné rozložení v čase (obrázek č. 2) [11].

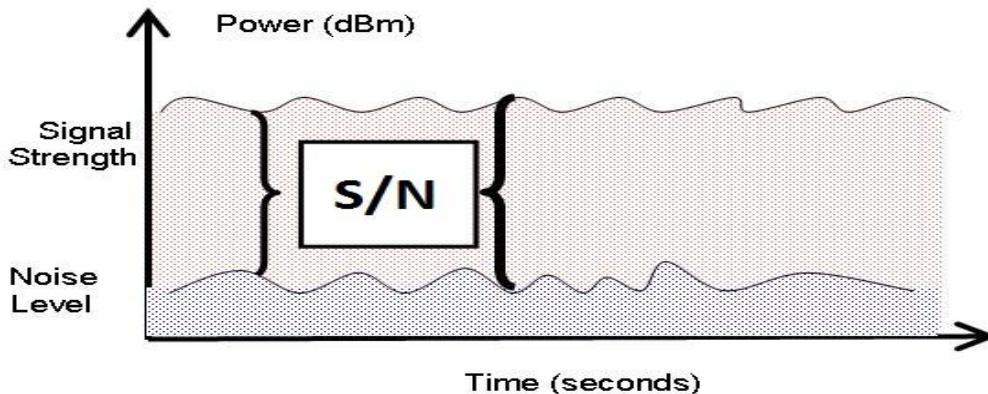
### 1.2.2 Odstup signál/šum (SNR).

Řekněme, že na vstupu máme zesilovač se zesílením  $G$  (obrázek č. 3). Zesilovač má vstupní signál o výkonu  $P_s^{in}$ . Na výstupu, kromě užitečného signálu, v důsledku nedokonalostí součástek obvodu a fyzickým jevem protékajícím uvnitř součástek, dostaneme šum, tj. výstup zesilovače bude  $P_s^{out}$ ,  $P_n^{out}$  [1],[6],[26]



Obrázek č.3. Schematické představení vstupního a výstupního signálu.(prekresleno z [6])

Odstup signál/šum – je poměr úrovně výkonu užitečného signálu k úrovni výkonu šumu, vyjádřené většinou v decibelech. Alternativně se tento parametr dá představit jako poměr druhé mocniny dvou amplitud – užitečného signálu a šumu[1],[26].



Obrázek č.4. Schematické představení pojmu SNR (převzato z [17])

$$SNR = \frac{P_{signal}}{P_{noise}} = \left(\frac{A_{signal}}{A_{noise}}\right)^2; \quad (6)$$

kde  $P$  – střední výkon,  $A$  – RMS hodnoty amplitudy[26].

Vyjádření v decibelech[26]:

$$SNR(dB) = 10^{10} \log\left(\frac{P_{signal}}{P_{noise}}\right) = 20^{10} \log\left(\frac{A_{signal}}{A_{noise}}\right); \quad (7)$$

Na výstup zesilovače se někdy zapojuje filtr s charakteristikou typu A, která určitým způsobem zohledňuje charakter lidského ucha. Hodnoty získané při měření s tímto filtrem pak vycházejí příznivější. Princip filtru založen na simulace ucha člověka, které má největší citlivost při zvuku o kmi-



točtu přibližně 1kHz. A-filtr je podobný jako filtr s pásmovou propustí – má největší propust na 1kHz a potlačuje šum o nízkých a vysokých frekvencích a chrání ucho od příliš velkých hlasitostí a intenzit. Je vhodný pro malé úrovně šumu[1],[3],[9].

### 1.2.3 Kvantizační šum.

Z důvodů toho, že počítač umí číst jenom číslicové signály, je potřeba převést analogové záznamy do digitální formy. Digitální záznam z analogového je možný dostat pomocí tzv. A/D převodníku. Spojitý analogový signál bude popsán pomocí posloupnosti vzorků, přičemž spojitost se změní na diskrétnost[32],[33]. Následně jednotlivé vzorky kvantováním se převádějí na čísla. Kvůli nedokonalité rozlišení A/D převodníků, během kvantování vzniká tzv. kvantizační šum, což je odlišnost kvantovaných hodnot od navzorkovaných hodnot. Šum takového typu má klasifikace aditivního šumu a jeho vliv na užitečný signál popsán následujícím vztahem[32]:

$$SNR_{A/D} = 20 * \log 2^N \approx 6,02 * N \text{ [dB]} \quad (8)$$

Kde  $2^N$  je maximální počet kvantizačních úrovní skutečného převodníku a N je počet bitů Podle vztahu (7) se da vyjádřit to, že úroveň SNR A/D převodníku závisí jenom na velikostí užitečného signálu, tj. na hodnotě N [32].

## 1.3 Zkreslení signálu.

### 1.3.1 Total Harmonic Distortion (THD).

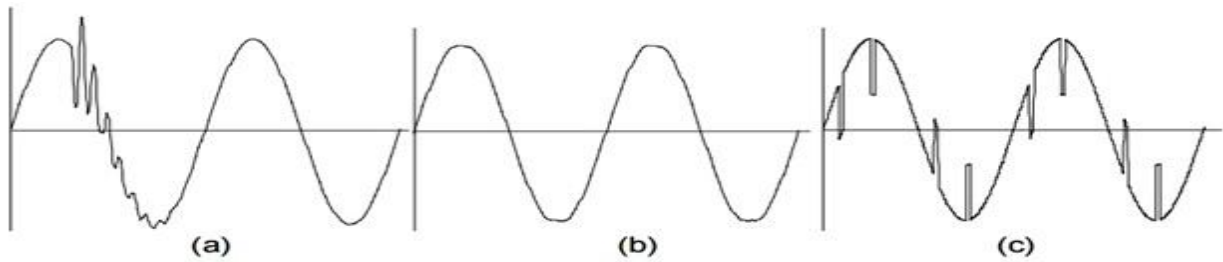
Harmonické šumy jsou změny tvaru napěťové vlny jdoucí od zdroje, jinými slovy zkreslení sinusové funkce napětí. Největší vliv na vznik takových šumu mají přístroje, které se zapojují do sítí, a mají velkou impedanci. Zkreslení sinusové funkce je v podstatě směs signálů, které se nazývají harmonické[2].

Každá harmonická má podle pořadí kmitočet, který je činitel hlavní harmonické, tj. když hlavní harmonická bude mít kmitočet 500 Hz, druhá harmonická bude mít kmitočet 1000 Hz, třetí 1500, čtvrtá 2000 atd. Sumace všech harmonických v jediný systém udává pojem Total Harmonic Distortion – THD[1],[2].

Ideální sinusovka nemá v sobě harmonické komponenty, kromě hlavní harmonické.

Existují 3 hlavní typy zkreslení (obrázek č.5)[15]:

- a) neharmonické zkreslení
- b) flap-top zkreslení
- c) notching (vyřezové) zkreslení



Obrázek č.5.a)neharmonické zkreslení.b)flap-top zkreslení.c) vyřezové zkreslení (převzato z [15])

Total Harmonic Distortion (THD) je RMS (Root Mean Square) všech vyšších harmonických s porovnáním ke hlavní harmonické, přičemž výsledek se udává procentuálně[1],[2]:

$$THD = \frac{\sqrt{V_2^2 + V_3^2 + V_4^2 \dots}}{V_1} * 100\% \quad (9)$$

Kde  $V_1$  je hlavní harmonická a  $V_2, V_3, V_4$  atd. jsou ostatní harmonické signálu  
Čím vyšší bude hodnota THD, tím víc zkreslení je v sinusovce[15].

### 1.3.1 Total Harmonic Distortion + Noise (THD+N).

THD+N je velmi podobný parametr jako THD. Rozdíl je v tom, že kromě vyšší harmonických, se bere v úvahu jakýkoliv šum. Prakticky to znamená, že za šum se bude považovat všechna energie, která nepatří do užitečného signálu[1],[29].

$$THD + N = \frac{\sqrt{V_2^2 + V_3^2 + V_4^2 \dots + N^2}}{V_1} * 100\% \quad (10)$$

Kde  $V_1$  je hlavní harmonická,  $V_2, V_3, V_4$  atd. jsou ostatní harmonické signálu a  $N$  je šum.

THD+N se používá pokud je potřeba zjistit nejen zkreslení harmonických ale taky zkreslení mezi nimi, což neumožňuje THD. Měří se tak, že se bere všechno kromě hlavní harmonické, která se odstraňuje pomocí filtru s pásmovou zádrží a zbytek se porovnává s hlavní harmonickou procentuálně podle vztahu (9)[1],[29].



## 2. Praktická část.

Výpočet parametrů záznamů byl proveden pomocí programu Matlab R2014b s použitím dvou Toolboxů – Signal Processing ToolBox, DSP System Toolbox. Postup měření bude prezentován ve tvaru úseků zdrojového kódu s popisem. Každá funkce popsaná dále je v podstatě tlačítkem v uživatelském rozhraní celého programu. Proto je popis programu rozdělen na popis každého z těchto tlačítek.

### 2.1 Výpočet parametrů záznamu. Funkce `eval_params`.

V této části bude představen postup a metody realizace měření parametrů analogových záznamů. Prvním krokem bude čtení záznamů z paměti počítače a ochrana proti chybovému výběru. Ve druhém kroku bude navržena rychlá Fourierova Transformace (FFT) pro hledání spektra vybraného signálu, pomocí kterého potom ve třetím kroku najdeme všechny harmonické, včetně hlavní. Následně bude navržen váhový A-filtr, který použijeme pro filtraci signálu a následný výpočet všech parametrů odfiltrovaného signálu. Posledním krokem bude výpočet a ukládání potřebných parametrů. Výpočet všech parametrů vybraného uživatelem záznamu se provádí ve funkci `eval_params`. Pomocí spektra bude spočítané THD, THD+N a SNR. V časové rovině se bude počítat SNR. Funkce `eval_params` začíná pracovat při zmáčknutí tlačítka “Výpočet...”. V této části budou uvedeny metody výpočtu a způsoby realizace předložených parametrů.

*Poznámka: celé uživatelské rozhraní a princip fungování všech tlačítek bude popsán v části 2.2*

#### 2.1.1 Čtení záznamu.

Pro čtení záznamu Matlabem, se většinou používá funkce `audioread`. Nahrávky formátu `.wav` nesou v sobě kromě samotného signálu informaci o vzorkovacím kmitočtu. Proto se na výstupu tohoto příkazu vytváří pole `[Y, fs]`, kde `Y` je vektor, který v sobě nese informace o amplitudách záznamu a `fs` je vzorkovací kmitočet.

Při práci daného programu mohou nastat dva případy při čtení záznamu:

- První čtení
- Druhé čtení neboli čtení přiblížené (zoom in) části

Pro realizaci odlišování dvou případů je použit cyklus `if`. Tento cyklus pracuje na základě toho, že při prvním čtení vybraného záznamu Matlabem, má pole `hGUI` (pole uživatelského rozhraní) v sobě 4 prvky, vytvářené ve funkci `main`. Pokud program analyzuje přiblížený kousek načteného signálu, pole `hGUI` bude mít 5 prvků, tedy program bude navigován přechíst jenom uříznutou část načtené nahrávky, pomocí informací o nových hodnotách na ose X.

Postup realizace:

```
hGUI = getappdata(findobj('Tag', 'MainWin'), 'hGUI');
```

- sběr informace o poli `hGUI`. Příkaz `getappdata` vytáhne uloženou informaci z paměti programu. Hledání uložené informace se provádí při použití přiděleného jména – `tag`.



Pokud bude potřeba analyzovat jinde, než ze začátku nahrávky, uživatel může nastavit začátek v proměnně `start_sample`

```
if length(fields(hGUI))>4
frame = hGUI.axes(1).XLim;
[~, fs] = audioread(datasource, [1 2]);
```

pokud je délka pole `hGUI` větší než 4: proměna `frame`, s informací o limitech na ose X čtení z uložené v paměti (`datasource`) adresy vzorkovacího kmitočtu.

Čtení audiofilu v Matlabě jde od prvního vzorku, což znamená, že na prvním místě nemůže být nulový vzorek. Takovýto případ však může nastat, pokud bude uživatel zoomovat začátek nahrávky. Ochrana proti takovému případu, a zároveň čtení, je realizováno v dalším cyklu *if*:

Pokud levá hranice se rovná 0, čte se file od 1 vzorku. Takže počet vzorků dostaneme pomocí pravé hranice, tj. součinu počtu sekund na vzorkovací kmitočet, přičemž počet vzorků bude zaokrouhlován z důvodu toho, že se můžou vyskytnout hodnoty jiné než cela čísla.

```
if frame(1)==0
Y = audioread(datasource, [1 round(frame(2)*fs)]);
```

Pokud se čtení začíná z nenulového čísla, příkaz `audioread` bude vyplněn pomocí obou hranic na osách:

```
else
Y = audioread(datasource, round(frame*fs) + (start_sample-1));
end
```

Pro určení parametrů analogového záznamu stačí vzít jenom jeden z kanálů, pokud signál je stereo. Proto následujícím krokem je určení, jestli záznam má jeden nebo dva kanály, a pokud má dva, odstranění jednoho z nich.

```
[~, Nch] = size(Y);
if Nch ~= 1
Y = Y(:,1);
End
```

Vytvořena matice `[~, Nch]` v sobě nese jenom počet sloupců načteného záznamu `Y`. Pokud je sloupců víc než jeden, odstraníme ostatní tím, že vrátíme do `Y` jenom první sloupec.

Zakládá se pole proměnných `params.`, do kterého se budou ukládat všechny vypočtené parametry daného záznamu.

```
params.fs = fs;
```

prvním uloženým parametrem do paměti programu je vzorkovací kmitočet `fs`.

Dosud bylo řečeno o opakovaném přečtení záznamu. Tedy pokud program vidí záznam poprvé, přečte záznam pomocí funkce `audioread`, kde na vstup dáme `datasource`, tj. adresa k filu. Vybraný úsek pro analýzu je 6 milionů vzorků, jdoucí od 1. Ten parametr šířky úseku určen experimentální metodou, je univerzální a podle potřeby může být změněn.

Výstupem této funkce je vektor `[Y, fs]`.

`Y` – nese v sobě informace o záznamu ve tvaru matice.

`fs` – vzorkovací kmitočet, skalární veličina.





Stejně, jako v prvním případě cyklu *if*, provádí se ověření, zda file je stereo nebo mono, odstranění pokud je stereo, a taky přečtená hodnota vzorkovacího kmitočtu *fs* se zapisuje do pole *params*.

```
else
    [Y, fs] = audioread(datasource, [3e6 7e6]);
    [~, Nch] = size(Y);
    if Nch ~= 1
        Y = Y(:,1);
    end
    params.fs = fs;
end
```

## 2.1.2 Spektrum signálu.

Dalším krokem je hledání spektra signálu, který je základem pro frekvenční analýzu. Nejvhodnější metodou pro výpočet spektra signálu je Diskrétní Fourierova Transformace (DFT). Spektrum signálu je v podstatě řada koeficientu Fourierovy Transformace. Konkrétně byla vybraná rychlá verze této metody – FFT (Fast Fourier Transformation) – rychlá Fourierova Transformace. Důvod pro použití FFT je zřejmý, když porovnáme složitost těchto dvou typu výpočtu[35],[36]:

- DFT má počet aritmetických operací  $O(N^2)$
- FFT má pouze  $O(N \log N)$  operací

FFT má výrazně menší počet operací a to znamená, že program bude fungovat rychleji při použití této metody.

Zavedeme novou proměnnou *L*, která nese informace o délce signálu.

```
L = length(Y);
```

V Matlabu FFT se provádí pomocí funkce *fft*.

Funkce *fft* má maximální rychlost při dodržování vztahu  $N = 2^n$ , kde *n* je počet diskretních hodnot. Najdeme ho jako nejbližší mocninu dvou, délky signálu *L* (*n* musí být větší nebo rovnat se délce *L*)

Hledáme *n* v Matlabu pomocí příkazu *nextpow2* (`nfft = 2^nextpow2(L)`)

Proměnná *freq* je tady ve tvaru frekvenčního vektoru. Udává nám veličiny, které budou na ose *X*, při kreslení spektra. Vektor tvoří funkce *linspace* tak, že zadáme počet bodů tohoto vektoru rovný *nfft* a položíme tyto body do záporné a kladné strany násobením -1 a 1.

```
freq = fs/2*linspace(-1,1,nfft);
```

Praktické použití tohoto řádku bude uvedeno v dalších částech kódu.

Rychlá Fourierova Transformace signálu *Y* přes jeho délku *L* s počtem diskretních hodnot *nfft*:

```
S = fft(Y, nfft)/nfft;
```

Z výsledku počítání spektra je spočítaná amplituda jako absolutní hodnota spektra a vyjádření hodnoty amplitudy v [dB]: `mag = abs(S); mag_dB = 20*log10(mag);`

## 2.1.3 Hledání harmonických signálu.

Předpoklad je v tom, že každá harmonická má odlišné rozměry od celého spektra, proto bude princip hledání proveden na základě parametrů výšky, šířky a vzdálenosti mezi sebou jednotlivých harmonických.



```
[peaks, locs] = findpeaks(fftshift(mag_dB), 'MinPeakDistance', 400, ...  
    'MinPeakHeigh', -100, ...  
    'MinPeakProminence', 1.3*max(mag_dB) - mean(mag_dB));
```

Funkce **findpeaks** hledá harmonické na základě zadaných parametrů:

- fftshift(mag\_dB)** – posunutí nulové frekvence uprostřed spektra.
- MinPeakDistance** – parametr vzdálenosti, na jaké minimální vzdálenosti od každé harmonické má program provádět hledání.
- MinPeakHeigh** – parametr výšky, tj. minimální výška, od které začne program hledání
- MinPeakProminence** – minimální výtečnost, hrubě řečeno minimální předpokládaná výška a šířka harmonických.

Na základě výše uvedených parametrů, Matlab najde a uloží do pole **[peaks, locs]** informace o jednotlivých špičkových hodnotách a pozice těchto špiček.

*Poznámka: všechny číselné odhady tady byly provedeny experimentálně, proto změna jakéhokoliv parametru může výrazně ovlivnit proces hledání špiček signálu.*

## 2.1.4 A – filtr ( A– weighting filtr).

Zavedeme A-filtr pomocí funkce **fdesign.audioweighting** . Tato funkce má v sobě uloženo několik typů filtru, proto nastavením písmena A vybereme přesně A-filtr.

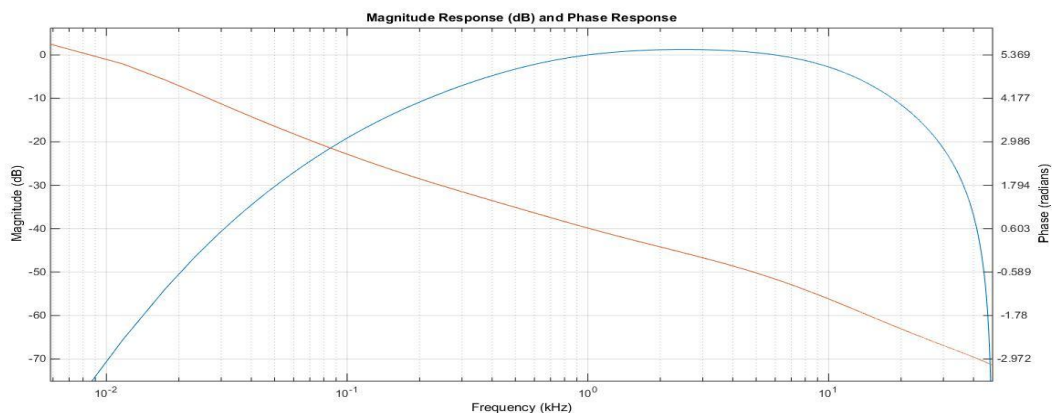
```
h = fdesign.audioweighting('WT,Class','A',1,fs);
```

Proměnnou **Ha** vytvoříme tzv. Filter System Object, který je nutný pro určení kmitočtové charakteristiky filtru. Filter System Object je speciální typ objektu v Matlab, který je určen pro práci s velkým dynamickým objemem dat.

```
Ha = design(h, 'ansis142', 'SystemObject', true);
```

Kmitočtová charakteristika filtru se vypočítává v proměnné **hfilter**. Funkce **freqz** má na svém vstupu už vypočtený Filter System Object a na výstupu vrátí kmitočtovou odezvu pro **Ha** s použitím **nfft/2** počtem vzorků.

```
hfilter{1} = freqz(Ha, nfft/2);
```



Obrázek č. 6. Kmitočtová charakteristika A – filtru.





Vytvořený A-filtr momentálně má tvar matice koeficientu, tzv. forma Second Oder Section  
Tato matice má následující tvar:

$$\text{sos} = \begin{bmatrix} b_{01} & b_{11} & 1 & a_{11} \\ b_{02} & b_{12} & 1 & a_{12} \\ \dots & \dots & \dots & \dots \\ b_{0L} & b_{1L} & 1 & a_{1L} \end{bmatrix}$$

Funkce **filter** potřebuje mít na vstupu koeficienty přenosové funkce  $H(z)$ , aby bylo možné filtrovat signál, proto je potřeba provést transformace *sos* do tvaru koeficientu přenosové funkce  $H(z)$ .

Transformace se provádí pomocí příkazu **sos2tf**, kde vstupem je uvedená výše SOS matice A-filtru a výstupem je pole s koeficienty přenosové funkce **[b, a]**

```
[b, a] = sos2tf(Ha.SOSMatrix);
```

**y\_A** je signál **Y**, který prošel A-filtrem. Na vstup se dávají koeficienty přenosové funkce  $H(z)$  a původní signál **Y**, výstupem, pomocí příkazu **filter**, je odfiltrovaný signál.

```
y_A = filter(b, a, Y);
```

Stejným způsobem jako pro neodfiltrovaný signál najdeme spektrum signálu po filtraci **S\_A** (pomocí FFT) a jeho amplitudu **mag\_A**:

```
S_A = fft(y_A, nfft)/nfft;  
mag_A = abs(S_A);
```

Ukládání vypočtených hodnot do pole **spec**:

```
spec.signal = mag_dB; - amplituda signálu do filtrace  
spec.A weigh = S_A; - spektrum signálu po filtrace
```

## 2.1.5 Hledání harmonických odfiltrovaného signálu.

Pro výpočet THD signálu, který prošel A-filtrem, je potřeba zopakovat proces hledání všech harmonických v tomto odfiltrovaném signálu. Realizace bude provedena stejným způsobem jako pro původní signál pomocí příkazu **findpeaks** a vstupních parametrů pro odhad špiček:

```
[peaks_A, ~] = findpeaks(fftshift(mag_A_dB), 'MinPeakDistance', 20, ...  
'MinPeakHeigh', -100, ...  
'MinPeakProminence', 1.6*max(mag_dB) - mean(mag_dB));
```

## 2.1.6 Výpočet THD.

Použitím nalezených špiček ze signálu **Y** se dá vypočítat THD podle vztahu (9).

Je potřeba najít hlavní harmonickou ve spektru signálu, tj. mezi harmonicky v proměně **peaks**, přičemž hodnoty každé harmonické až do té chvíle byly v dB, zatímco pro výpočet a vyjádření v procentech jsou potřebné jednotky výkonu. Proto je použit příkaz **db2pow**, který je určen přesně pro převedení z jednotek dB do jednotek výkonu.

```
V_fund = db2pow(max(peaks)); - hledání v proměně peaks maximální hodnoty, což je hlavní harmonická.
```



Spektrum má vlastnost symetrie na obou stranách, to znamená, že počet špiček je taky dvounásobný[7]. Proto se příštím krokem každá harmonická ukládá jen jednou (tj. polovina ze všech nalezených harmonických) do proměny `peaks1`:

```
peaks1 = db2pow(peaks(1:end/2));
```

Výpočet THD má následující tvar:

```
THD(1) = sqrt(sum(peaks1(peaks1~=max(peaks1))).^2)/V_fund;
```

Podobným způsobem se provádí odhad hodnoty THD pro odfiltrovanou verzi signálu:

```
V_fund = db2pow(max(peaks_A));  
peaks1_A = db2pow(peaks_A(1:end/2));  
THD(2) = sqrt(sum(peaks1_A(peaks1_A~=max(peaks1_A))).^2)/V_fund;
```

Následné vyjádření hodnot THD procentuálně násobením stovkou obou prvku pole THD:

```
THD = THD*100;
```

## 2.1.7 Výpočet THD+N.

Podle definice, THD+N je poměr hlavní harmonické signálu ke všem ostatním harmonickým plus všechno, co není užitečným signálem[1]. Myšlenka je v tom, aby se pomocí filtru s pásmovou zádrží daly vyříznout užitečné harmonické, tedy všechno co zbyde, bude „nežádoucím jevem“ ve jmenovateli definičního vztahu (10).

K dispozici v paměti programu v poli `[peaks, locs]` jsou uloženy harmonické signálu `Y` a její pozice. Položíme pozice každé harmonické na dřív vytvořený frekvenční vektor `freq` a uložíme tento nový vektor do proměny `f0`. Vektor `freq` má jak záporné tak i kladné hodnoty, proto nařídíme programu hledat jenom v kladné rovině

```
f0 = f0(f0>0);
```

Výsledná hodnota `f0` se ukládá do elementu struktury `spec`:

```
spec.f0 = f0;
```

Realizace filtru s pásmovou zádrží:

Při realizaci bylo vyzkoušeno několik způsobů vytvoření filtrů s pásmovou zádrží a bylo odhaleno, že s různými záznamy se může každý filtr chovat nestabilně a náhodně. Řešením byl návrh speciálního filtru v Matlab tak, že zadáváme parametry tohoto filtru - řád filtru, levou a pravou hranici, kde má být pokles o 3 dB. Pak Matlab pomocí vnitřních výpočtů vybere metodu návrhu potřebného filtru s pásmovou zádrží sám, podle nejmenší odchylky.

Zkoušením byl vybrán řád filtru 20: `n = 20;`

Původní a signál po filtraci(A-filtrem) se ukládají do proměných `y_n1` a `y_n2`, pro následující práce s nimi:

```
y_n1 = Y;  
y_n2 = y_A;
```



Postup návrhu filtru:

1. Prvním krokem je určení hranic filtru pro každou harmonickou. Šířka pásma **bw** byla stanovena zkosením, je optimální pro filtraci a neškodí signálu:

```
for ii = 1:length(f0)
    bw = 0.2*f0(ii);
    cut_freq = [f0(ii)-bw/2 f0(ii)+bw/2];
```

Výpočet se provádí po celé délce vektoru **f0**.

Proměna **cut\_freq** je pole, ve kterém prvním prvkem je spočítaná levá hranice a druhým prvkem je pravá hranice.

2. Provádí se kontrola dodržení Kotělnikovova teorému. Pokud kmitočtová charakteristika filtru přesáhne hranici  $fs/2$  budou hodnoty uměle změněny na pohraniční hodnotu  $fs/2$ :

```
if cut_freq(2)>fs/2
    cut_freq(2) = fs/2;
end
```

3. Tvorba filtru. Příkazem **fdesign.bandstop** bude vytvořena proměnná **d**, která má typ objektu **D**. Takový objekt v sobě nese parametry budoucího filtru (řád, hranice). Z několika představených šablon byl vybrán následující, vhodný pro práci daného programu:

```
d = fdesign.bandstop('N,F3dB1,F3dB2', n, cut_freq(1), cut_freq(2), fs);
```

Pak funkce **design** používá parametry vytvořeného objektu **D** pro konstrukce filtru s pásmovou zádrží:

```
hD = design(d, 'butter');
```

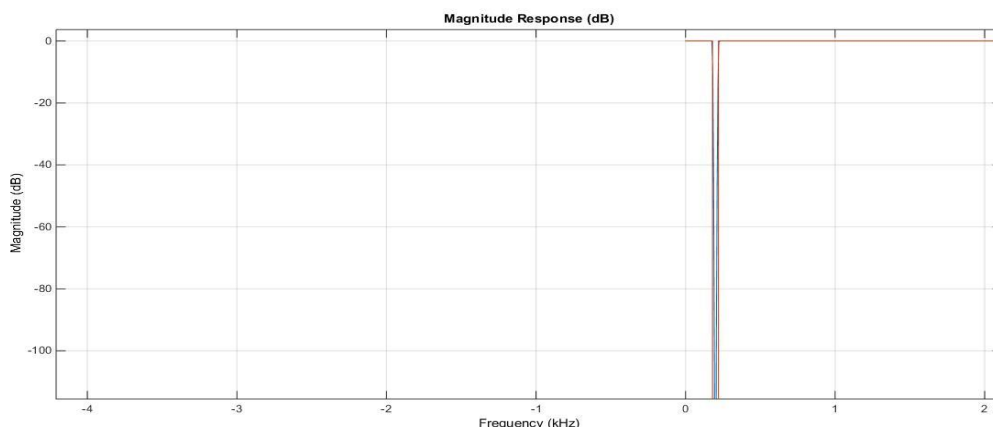
Výsledkem je **hD** – Filter System Object vybraného Butterworth IIR filtru.

4. Návrh kmitočtové charakteristiky filtru. Používá se příkaz **freqz**, a matice **SOS** z proměnné **hD**:

```
hfilter{2} = freqz(hD.sosMatrix, nfft/2);
```

5. Filtr je hotový, provádí se filtraci jednotlivých signálů.

```
y_n1 = filter(hD,y_n1);
y_n2 = filter(hD,y_n2);
end
```



Obrázek č. 7. Kmitočtová charakteristika filtru s pásmovou zádrží pro jednu harmonickou.



Výpočet THD+N.

Použitím nalezených výsledků filtrace se dá vypočítat další parametr vybraného záznamu – THD+N. V dvou proměnných –  $y_{n1}$  a  $y_{n2}$  jsou uloženy všechny nežádoucí elementy pro původní signál a signál, který prošel A-filtrem. Poměrem těchto elementů a užitečného signálu dostaneme potřebný parametr:

```
THD_N(1) = sqrt(sum(y_n1.^2))/sqrt(sum(Y.^2))*100;  
THD_N(2) = sqrt(sum(y_n2.^2))/sqrt(sum(y_A.^2))*100;
```

## 2.1.8 Výpočet SNR

V daném programu existují 2 metody pro výpočet SNR:

### 1. Spektrální metoda výpočtu.

Pokud je vybraná spektrální metoda výpočtu v tlačítku „radiobutton“ „Metoda výpočtu SNR“ Matlab to přečte a bude počítat SNR přes amplitudu spektra.

```
switch snr_method  
case 'spectrum'
```

Amplitudová definice SNR je kvadratický poměr amplitudy užitečného signálu (proměnná  $mag$ ) k amplitudě šumu ( $mean(mag)$ )(vztah 6)[1].

Tento poměr je implementován v Matlabu následujícím způsobem:

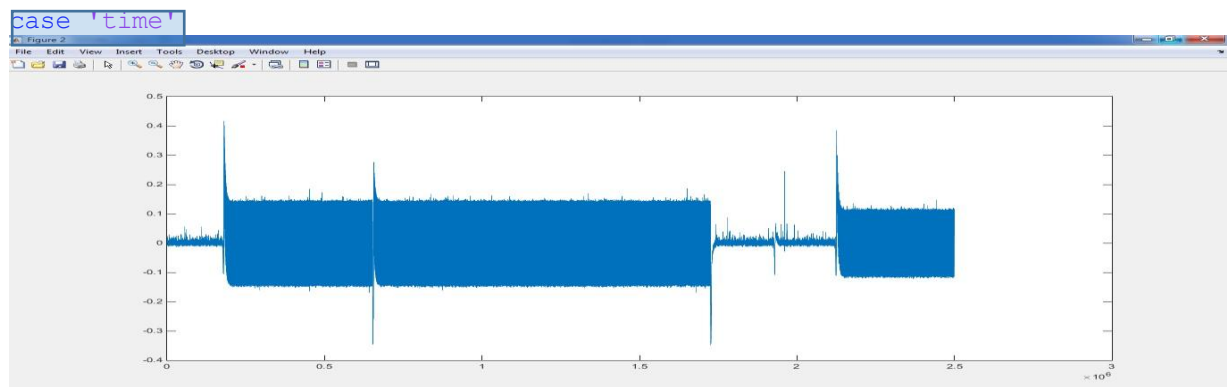
```
SNR = [max(mag)^2/mean(mag)^2 max(mag_A)^2/mean(mag_A)^2];  
SNR_dB = 10*log10(SNR);
```

$SNR$  je proměnná ve tvaru pole, které má v sobě výpočet zároveň s použitím A-filtrování a bez něho.

$SNR_{dB}$  vyjádřená hodnota SNR v decibelech (vztah 7).

### 2. Časová metoda výpočtu.

Pokud by byla uživatelem vybraná časová metoda výpočtu, Matlab bude analyzovat v čase, kde je užitečný signál a kde je šum. Pak na základě výsledků bude vypočítán SNR

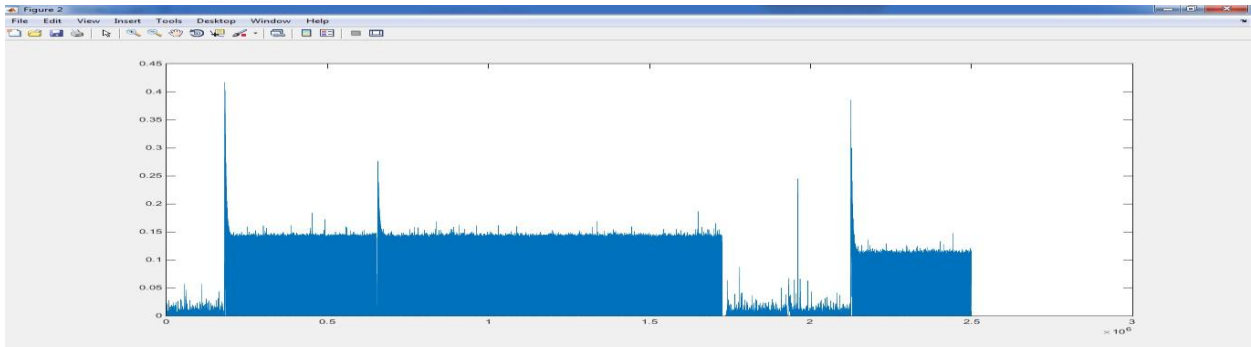


Obrázek č. 8. Základní tvar vybraného úseku signálu(příklad)



Jak je vidět z obrázku č. 8, hodnoty amplitudy signálu jsou v kladné i záporné rovině. Pro výpočet SNR v čase je potřebná jenom jedna část, proto bylo rozhodnuto vynulovat záporné hodnoty.

```
y = Y;  
y(y<0) = 0;
```

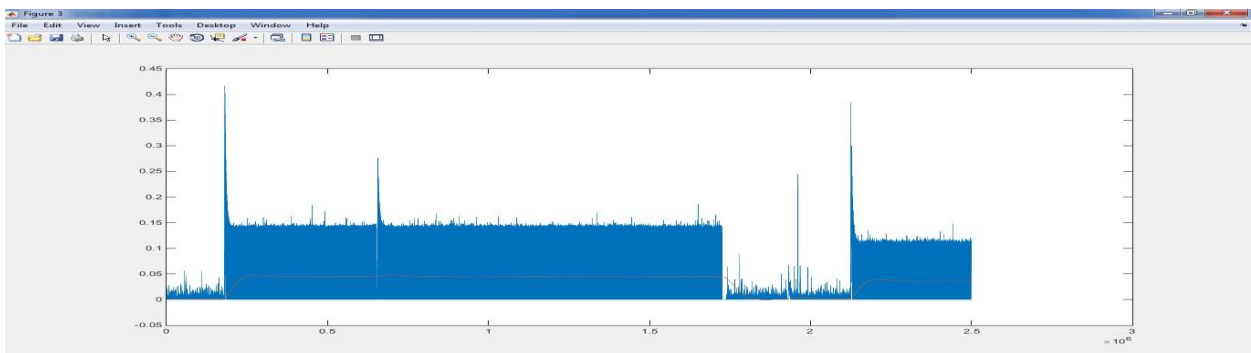


Obrázek č. 9. Kladná část signálu

Podle obrázku se dá pochopit, že amplituda signálu není “rovná”. Myšlenka je tedy v tom, vzít střední hodnotu amplitudy kde je úsek užitečného signálu, střední hodnotu amplitudy šumu a udělat z nich poměr. Výsledkem poměru bude SNR.

Realizace se bude provádět pomocí nízkofrekvenčního filtru.

```
n = 2; - n je řad filtru.  
fc = 0.5; - fc je hodnota řezu (cutoff) filtru.  
[b,a] = butter(n, fc*2/fs); - koeficienty nf filtru pomocí funkce butter (Butterworth IIR  
filtr)  
y1 = filter(b, a, y); - y1 je odfiltrovaný signál.
```



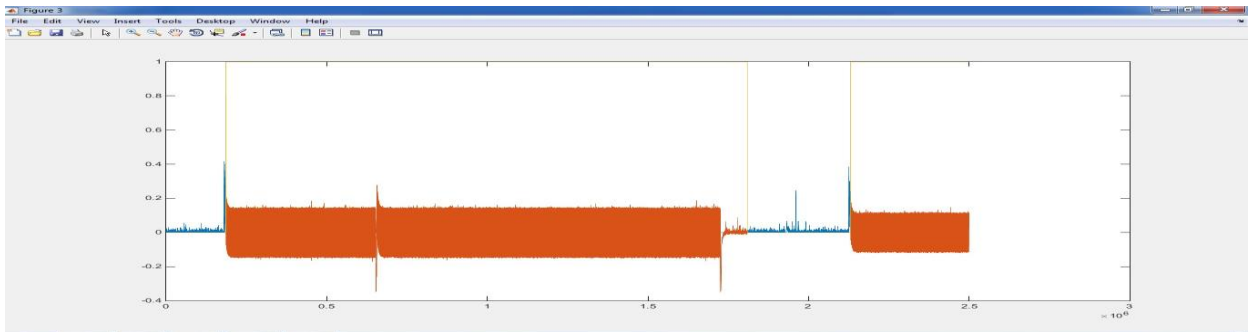
Obrázek č. 10. Demonstrace návrhu nf filtru.

Křivka filtru je pod maximální amplitudou, ale to neovlivní konečný výsledek, protože nás zajímá jen poměr dvou hodnot amplitudy, který bude stejný, i když leží několikrát níž. Pak je potřeba ukázat programu, kde je užitečný signál a kde je šum.

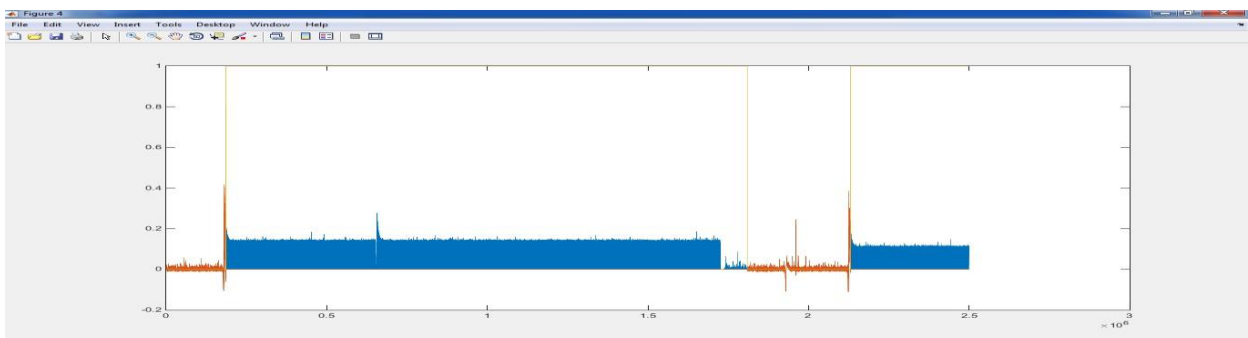


```
ind = y1 > 0.1 * max(y1);  
ys = Y; yn = Y;  
ys(ind == 0) = 0;  
yn(ind == 1) = 0;
```

- **ind** má hodnoty, které jsou více než 10% od max. amplitudy
- zakladáme 2 nové proměnné.
- proměnná, která má v sobě jen užitečný signál
- proměnná, která nese v sobě šum



Obrázek č. 11. Jak Matlab uvidí užitečný signál (proměna **ys**)



Obrázek č. 12. Jak Matlab uvidí šum (proměna **yn**)

Pak se provádí výpočet SNR jako RMS poměr dvou amplitud.

```
SNR = (rms(ys)/rms(yn))^2;  
otherwise  
error('Nespravny vyber metody vypoctu SNR')  
end
```

V případě, že při hodnocení tlačítka s výběrem metody výpočtu došlo k chybě, zobrazí se výstražní pop-up okno.

Kromě toho, proměna **y1** bude uložena do pole **params**:

```
params.signal.sig LF = y1;
```

## 2.1.9 Ukládání výsledků

Všechny výsledky počítání, které ještě nebyly uloženy se vkládají do struktury **params** :

```
params.THD_N = THD_N;  
params.SNR_dB = 10*log10(SNR);  
params.hfilt = hfilter;  
params.freq = freq;
```



```
params.signal.source = Y;  
params.signal.fund.pks = peaks;  
params.signal.fund.lcs = locs;
```

Pokud všechno proběhne úspěšně, na obrazovce se objeví zpráva „Výpočet ukončen“:  
`msgbox('Vypocet ukoncen')`

## 2.2 Návrh uživatelského rozhraní (interface).

V této části se bude jednat o vytvoření uživatelského rozhraní a grafické zpracování výsledků z části Cílem návrhu uživatelského rozhraní je zjednodušení ovládání vytvořeného programu. Bude vytvořen panel ze 4 tlačítek, které umožní vybrat a načíst záznam, vybrat metodu pro výpočet SNR a zobrazovat výsledky graficky.

### 2.2.1 Funkce MAIN

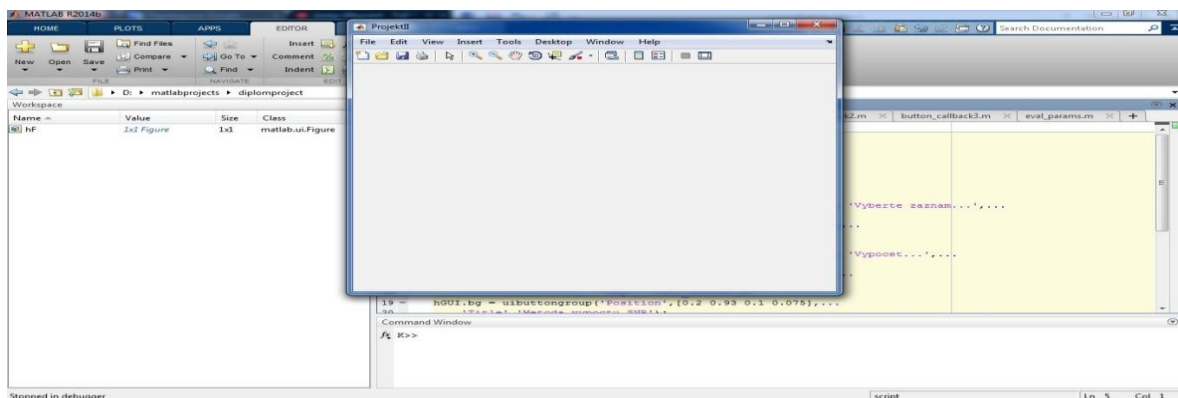
Základem uživatelského rozhraní je hlavní okno, ve kterém uživatel bude moct nastavovat parametry výpočtů.

```
hF = figure('Name', 'Spectralaudio v2.3', 'NumberTitle', 'off',...  
          'Tag', 'MainWin');
```

Kde příkazem `figure` se vytvoří proměna typu `matlab.io.figure`, což je okno, a do závorek udáváme parametry tohoto okna:

```
('Name', 'Spectralaudio v2.š'..  
'NumberTitle', 'off',...  
'Tag', 'MainWin');
```

- název okna;
- vypnutí numerace oken, protože budeme mít jenom jedno.
- přiřadíme pro okno klíčové slovo, tzv. tag, pomocí kterého budeme moct zavolat okno v jiných funkcích.



Obrázek č. 12. Vytvořené okno

`pause(0.5)` - experimentálně zjištěna hodnota. Zavedena z důvodu toho, že bez pauzy program přeskakuje následující řádek.

Pak se rozšíří okno na celou obrazovku pomocí tzv. handleru:

```
set(get(handle(hF), 'JavaFrame'), 'Maximized', true);
```





`get(handle(hF), 'JavaFrame'), -` volá se funkce `hF` a dovoluje programu upravovat okno `hF`  
`'Maximized', true);` - rozšiřuje se okno do maximálního rozměru, tj. do celé obrazovky.

`clc;` tímto příkazem tzv. command window bude očištěno.

Při další práci v okně nepoužíváme horní panel, proto ho následujícím příkazem odstraňujeme:

`set(hF, 'toolbar', 'none')`

V dalším kroce je vytvořeno první tlačítko v okně, pomocí kterého se bude otevírat menu výběru záznamů:

```
hGUI.butons(1) = uicontrol('Style', 'pushbutton', 'String', 'Vyberte  
zaznam',...  
    'Units', 'normalized',...  
    'Position', [0.01 0.95 0.075 0.03],...  
    'Callback', 'button_callback',...  
    'Tag', 'Load_button');
```

`hGUI.butons(1) = uicontrol` – do struktury `hGUI` se zakládá proměna `butons(1)`, element interfacu uživatele, která je v našem případě tlačítkem;

`'Style', 'pushbutton',` – naznačíme typ tlačítka “pushbutton”, což označuje, že bude aktivovaná zmáčknutím;

`'String', 'Vyberte zaznam',` – uvnitř tlačítka bude napsáno „Vyberte záznam“;

`'Units', 'normalized',` – znamená, že při nastavování polohy a rozměrů tlačítka, bude referenční hodnota rozměr celého okna (levý nižší uhel je poloha 0,0; horní pravý uhel je 1,1)

`'Position', [0.01 0.95 0.075 0.03],` – nastavování polohy a rozměrů tlačítka. První dvě čísla v závorkách je poloha na ose X a Y, třetí číslo je šířka a čtvrté číslo je výška;

`'Callback', 'button_callback',` – při zmačknutí tlačítka, program nabídne funkci, která se nazývá `button_callback` a program začne pracovat s touto funkcí.

`'Tag', 'Load_button'` – přiřadíme tag „Load\_button“.

Stejné operace se provádějí s tlačítky `hGUI.butons(2)` a `hGUI.butons(3)`. Ve vlastnostech se jenom mění nadpis uvnitř, jejich poloha a klíčové slovo (tag).

```
hGUI.butons(2) = uicontrol('Style', 'pushbutton', 'String', 'Vypocet',...  
    'Units', 'normalized',...  
    'Position', [0.1 0.95 0.075 0.03],...  
    'Callback', 'button_callback2',...  
    'Tag', 'Calc_button');
```

```
hGUI.butons(3) = uicontrol('Style', 'pushbutton', 'String', 'Graficke  
zobrazeni',...  
    'Units', 'normalized',...  
    'Position', [0.31 0.95 0.075 0.03],...  
    'Callback', 'button_callback3',...  
    'Tag', 'plots');
```





Při analyzování záznamů da se spočítat SNR buď metodou časovou nebo spektrální. Proto následujícím krokem je zakládání tlačítka, které umožní výběr mezi dvěma způsoby výpočtu.

Uvnitř struktury `hGUI` přidáváme novou proměnou `bg`. Příkazem `=uibuttongroup` se zakládá malé okno, uvnitř kterého bude, jak se předpokládá, skupina tlačítek.

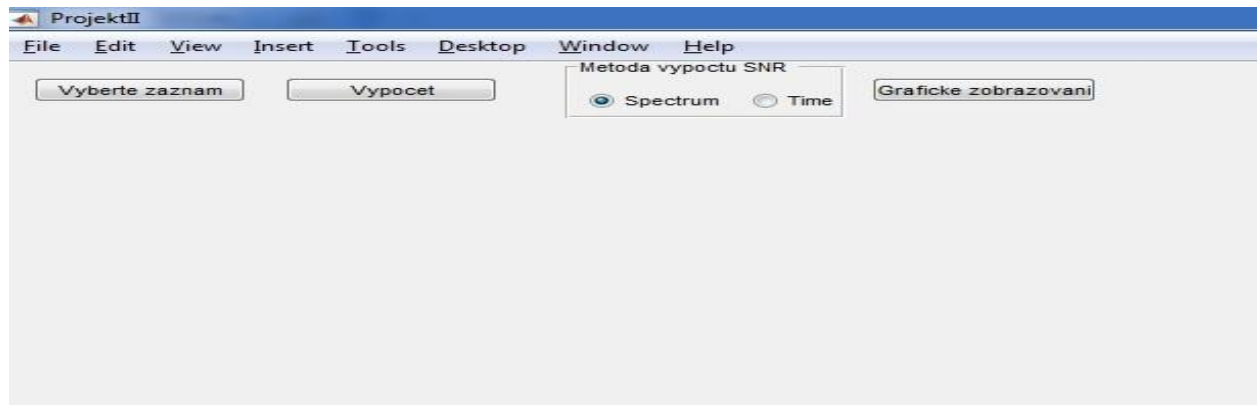
```
hGUI.bg = uibuttongroup('Position',[0.2 0.93 0.1 0.075],...  
    'Title','Metoda vypočtu SNR');
```

`r1` je nová proměnná v poli `hGUI`, která patří do okna `bg`. Typ nového tlačítka má být takovým, aby uživatel mohl přepínat, a dělat výběr mezi metodami výpočtu. Proto k `r1` je přiřazen typ `radiobutton`.

```
hGUI.r1 = uicontrol(hGUI.bg,'Style',...  
    'radiobutton',...  
    'String','Spectrum',...  
    'Position',[10 0 100 30], 'Tag','rb_s');
```

Stejným způsobem se přidává druhé tlačítko `r2`:

```
hGUI.r2 = uicontrol(hGUI.bg,'Style','radiobutton',...  
    'String','Time',...  
    'Position',[90 0 100 30], 'Tag','rb_t');
```



Obrázek č. 13. Přibližný pohled na vytvořený hotový interface programu

## 2.2.2 Funkce `button_callback`.

Při zmáčknutí tlačítka `Vyberte záznam`, začne pracovat funkce `button_callback`.

`global source_path` - zavedení globální proměnné. Dělá se pro zrychlení práce programu, aby Matlab nehledal adresu vybraného záznamu pokaždé, když ho potřebuje. Udává krátkost v kódu.

Dále se zakládá interface, pop-up okno, ve kterém uživatel programu bude moct vybírat záznam:

```
[FileName,PathName,FilterIndex] = uigetfile('*.*wav',...  
    'Vyberte zaznam');
```

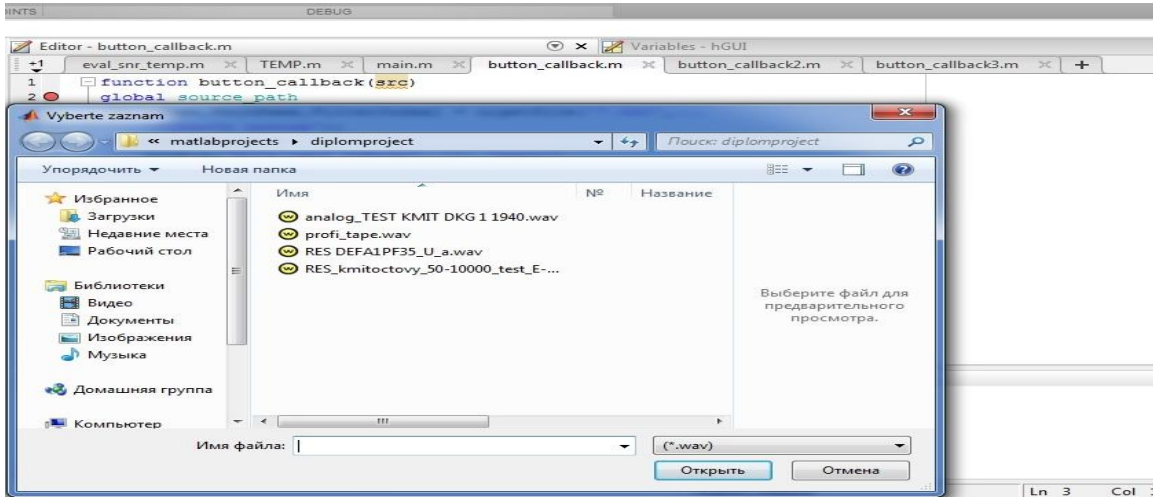
Otevře se nové okno “Vyberte záznam“, kde program nabídne vybrat záznam ve formátu `.wav`. Po vybraní se vrátí hodnota v následujícím tvaru:

`[Jméno filu, adresa k filu, 0 nebo 1]`; `1` znamená, že operace prošla úspěšně.

`0` znamená, že operace z nějakého důvodu byla přerušena.



Poznámka: je potřeba vzít v úvahu, že 0 a 1 jsou tady jako čísla, nikoliv indexy.

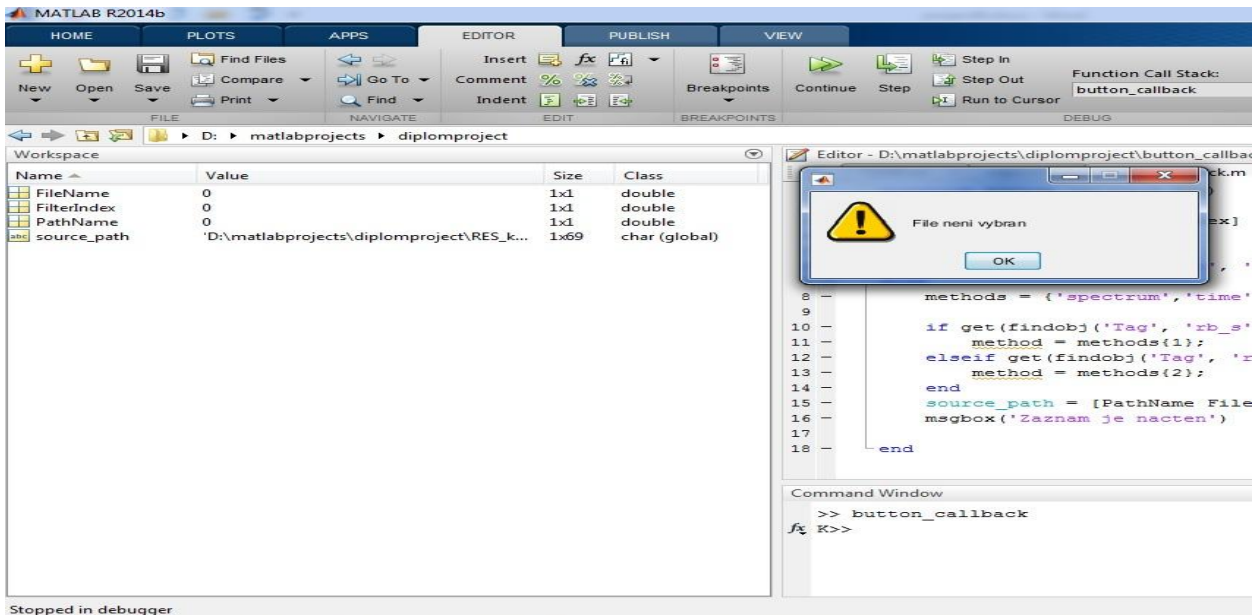


Obrázek č. 14. Okno výběru záznamů

Pro případ, když záznam nebude vybrán nebo operace bude zamítnuta a vrátí se 0 v proměně `FilterIndex` je připraven následující kousek kódu:

```
if FilterIndex==0
    msgbox('File není vybrán', '', 'warn')
```

V okamžik, když Matlab dostane 0 v `FileIndex`, objeví se upozorňující zpráva, že file není vybrán.



Obrázek č. 15. Výstražní okno



Považujeme, že záznam je vybrán a Matlab dostal 1, tedy začíná se kontrola metody výpočtu SNR:

```
else  
    methods = {'spectrum', 'time'};
```

 - matice se dvěma elementy typu string.

```
if get(findobj('Tag', 'rb_s'), 'Value')  
    method = methods{1};
```

 – pomocí tagu hledáme radiobutton **Spectrum**  
– a ptáme se jestli je zmáčknutá pomocí logické 1 nebo 0

To znamená, že když program po dotazu dostane 1 (tj. vybraná spektrální metoda výpočtu SNR) do proměnné **method** bude uložena spektrální metoda

Pokud přijde 0 program se zeptá na časovou metodu výpočtu :

```
elseif get(findobj('Tag', 'rb_t'), 'Value')  
    method = methods{2};  
end
```

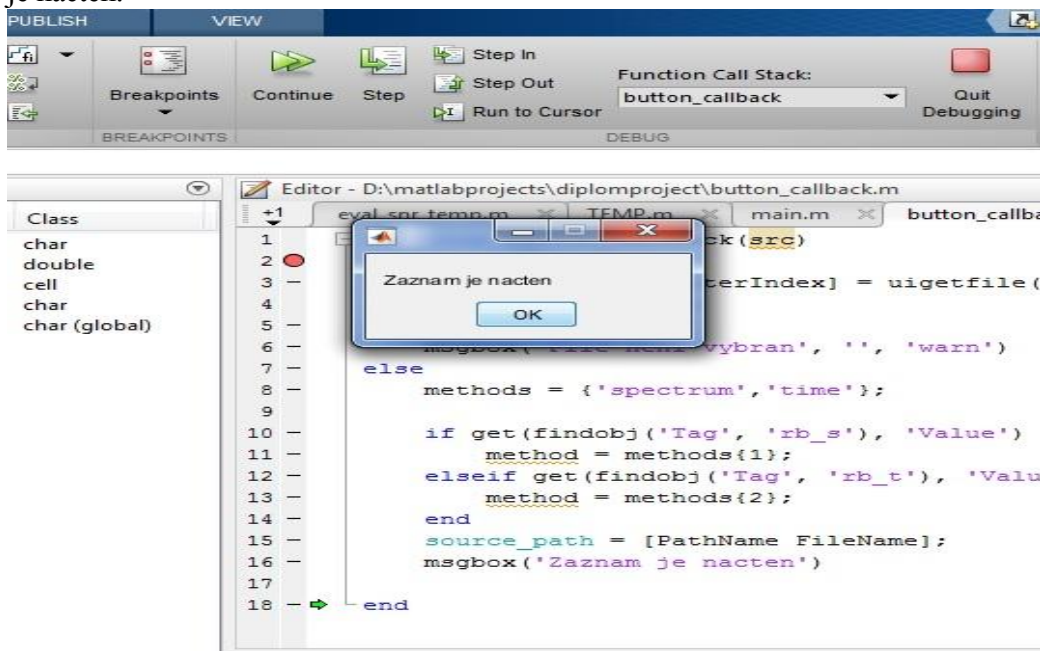
 – pomocí tagu program se ptá na hodnotu časové metody.

Pak se zavádí proměnná, do které se zapisuje název a adresa vybraného záznamu.

```
source_path = [PathName FileName];
```

 - dvourozměrná matice, kde na prvním místě je adresa vybraného záznamu a na druhém jeho jméno.

```
msgbox('Zaznam je nacten')
```

 – po ukončení celé operace se objeví zpráva o tom, že záznam je načten.

Obrázek č. 16. Výsledek práce button\_callback



### 2.2.3 Funkce `button_callback2`.

V této funkci se probíhá výběr metody výpočtu SNR.

Volá se proměnná `source_path` z paměti Matlab pro práci s touto proměnnou ve funkci `button_callback2`. `global source_path`

Opakuje se celý cyklus s kontrolou zmáčknutí tlačítka metody výpočtu SNR. Děla se pro případ, že by uživatel během práce programu změnil svůj výběr, a aby metoda, která nebyla vybraná byla uložena jako 0 hodnota.

```
methods = {'spectrum', 'time'};

if get(findobj('Tag', 'rb_s'), 'Value')
    method = methods{1};
elseif get(findobj('Tag', 'rb_t'), 'Value')
    method = methods{2};
end
```

Následujícím řádkem odesíláme adresu záznamu v proměnné `source_path` a vybranou metodu v proměnné `method` pro výpočet ve funkci `eval_params`. Odesílání se provádí tak, že se do vstupních parametrů dávají dvě proměnné, které je potřeba odeslat a na výstupu budou uloženy do pole `[params, spec]`

```
[params, spec] = eval_params(source_path, method);
```

Pak se zavádí tzv. `cell array` matice, kam se ukládají výsledky počítání:

```
data = {params spec};
```

Příkaz `setappdata` je speciální příkaz pro ukládání dat v uživatelském rozhraní.

```
setappdata(findobj('Tag', 'MainWin'), 'data', data);
```

Standartní syntax pro `setappdata` je (GUI, jméno, hodnota).

`findobj('Tag', 'MainWin')` – příkazem `findobj`, podle tagu, voláme hlavní okno.

`'data'` – je název, se kterým bude všechna informace uložena.

`data` – informace, která bude uložena, tj. v tomto případě proměnná `data`.

### IV. Funkce `button_callback3`.

Při zmáčknutí tlačítka Grafické zobrazení, začne pracovat funkce `button_callback3`.

Prvním krokem se volají z paměti programu všechny parametry a vypočtené hodnoty.

Nejprve se volá `data` příkazem `getappdata` (syntax je stejný jako u `setappdata`)

```
data = getappdata(findobj('Tag', 'MainWin'), 'data');
```

Pak se elementy pole `data` - `params` a `spec` mezi sebou rozdělují:

```
params = data{1}; spec = data{2};
```

Pokud je uživatelem vybraná časová metoda výpočtu SNR, z pole `params` se vola nízkofrekvenční filtr, navržený dřív v `eval_params`:

```
if get(findobj('Tag', 'rb_t'), 'Value')
    y1 = params.signal.sig_LF;
end
```



Pak z pole `params` se volají jednotlivé hodnoty:

```
mag_dB = spec.signal;
Y = params.signal.source;
L = length(Y);
fs = params.fs;
f0 = spec.f0;
t = (0:L-1)/fs;
h_1 = params.hfilt{1}; % A-weighting filter
h_2 = params.hfilt{2}; % band-stop filter
freq = params.freq;
peaks = params.signal.fund.pks;
locs = params.signal.fund.lcs;
THD_N = params.THD_N;
SNR_dB = params.SNR_dB;
```

Grafické zobrazování:

Přípravou pro další grafické zpracování je příkaz `cla`, který vymezuje jakékoliv data na osách.

Funkce `subplot` se používá v případě, že se v jednom rozhraní bude kreslit zadané množství grafických zobrazení. Zakládání prvního grafického okna:

```
hSA(1) = subplot(211);
set(hSA, 'Tag', 'Time_scope')
```

Pokud je uživatelem vybraná časová metoda výpočtu SNR, kromě samotného signálu v prvním grafickém okně se zobrazí taky nízkofrekvenční filtr. Jinak bude zobrazován jenom signál:

```
if get(findobj('Tag', 'rb_t'), 'Value')
plot(t, Y, t, y1); grid on
else
plot(t, Y); grid on
end
```

```
xlabel('time, s')
```

```
ylabel('Amplitude, V')
```

```
xlim([min(t) max(t)])
```

-označíme, že na ose X je čas a jednotky jsou sekundy.  
-označíme, že na ose Y je amplituda a jednotky jsou volty.  
-hranice osy X se při přiblížení mohou měnit, tedy se změní podle času `t`.

Zakládání druhého grafického okna:

```
hSA(2) = subplot(212);
set(hSA, 'Tag', 'Spec_scope')
```

Vytvořené grafické rozhraní a informace o osách se pak ukládá do pole `hGUI`:

Nejprve se čte aktuální nastavení všech parametrů v pole uživatelského rozhraní `hGUI`:

```
hGUI = getappdata(findobj('Tag', 'MainWin'), 'hGUI');
```

Do pole `hGUI` se zapisují proměnné s vytvořeným grafickým rozhraním uvnitř. Pak se nová data ukládají zpátky do stejného pole, přičemž je potřeba upřesnit pro Matlab, že práce probíhá v aktuálním okně – pomocí příkazu `gcf` (current figure handle).

```
hGUI.axes(1) = hSA(1);
hGUI.axes(2) = hSA(2);
setappdata(gcf, 'hGUI', hGUI)
```



Kreslení v druhém grafickém okně probíhá s použitím tzv. `handlePlot` – ruční kreslení:

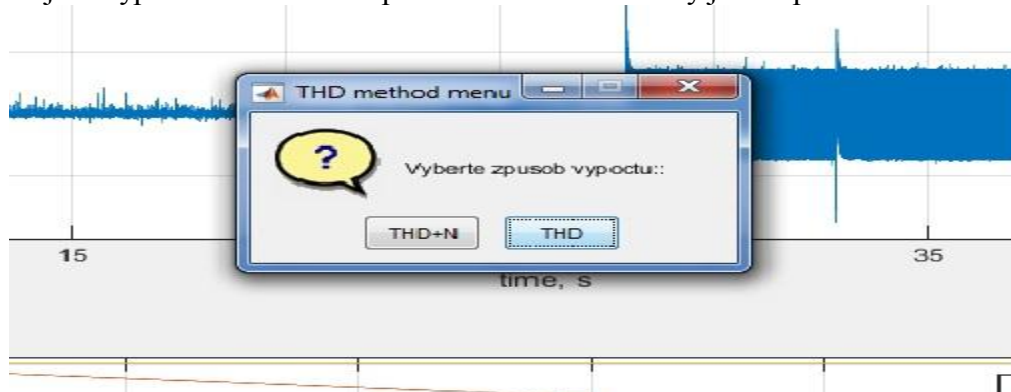
```
hP = plot(freq/1e3, fftshift(mag_dB), freq(locs)/1e3, (peaks), 'or', ...
         freq/1e3, [20*log10(abs(h_1(end:-1:1)))' 20*log10(abs(h_1))'], ...
         freq/1e3, [20*log10(abs(h_2(end:-1:1)))' 20*log10(abs(h_2))']);
grid on
legend(hP([1,3,4]), 'Spektrum signalu', ...
       'Kmitočtová charakteristika A-filtru', ...
       'Kmitočtová charakteristika filtru pasmove zadri')
xlabel('frequency, kHz')
ylabel('Magnitude, dB')
axis([0 fs/2]/1e3 -150 3])
```

Ruční kreslení je realizováno tak, že

1. Osa X, tj. kmitočtová osa, se dělí 1000, aby ty čísla mněly o 3 nuly méně.
  2. Kreslí se spektrum signálu **y** (**mag\_dB**)
  3. Označují se na grafu harmonické z pole **freq(locs)** (pozice na X), (**peaks**) (pozice na Y)
  4. Přidávají se kmitočtové charakteristiky dvou filtrů **h\_1** a **h\_2** v logaritmickém tvaru
  5. Pak se píše legenda grafu. V grafu jsou 4 objekty s pořádkovými čísly 1,2,3,4. Dvojkou se označí hlavní harmonické, na spektru jsou vyznačené červeným kolečkem, proto se v legendě o nich nepíše.
  6. Udává se jméno pro X souřadnici - je to frekvence a jednotky jsou kHz. Y souřadnice je rozsah (Magitude), v decibelech .
  7. Označují se maximum a minimum na osách Y a X. (funkce **axis**)
  8. Všechny vypočtené hodnoty během měření jsou vypsané a naznačena jejich pozice na grafu
- Po přepravění celého grafického bloku, se program zeptá uživatele, jakou hodnotu chce zobrazit – THD nebo THD+N.

```
choice = questdlg('Vyberte způsob vypočtu:', ...
                 'THD method menu', 'THD+N', 'THD', 'THD');
```

Objeví se okno, ve kterém budou předloženy dva způsoby výpočtů, ale v podstatě oba parametry už jsou vypočtené a uloženy v paměti. Používá se to tedy jenom pro zobrazení:



Obr.č.17. Pop-up okno výběru

V následujícím cyklu probíhá kontrola toho, jaký výběr udělal uživatel – metoda při výpočtu SNR a THD/THD+N . Pokud délka vektoru **SNR\_dB** je 1, byla vybraná časová metoda, protože pro časovou metodu nepočítáme SNR pro A-filtr. Pak se bude proměnná volat **choice** – výběr mezi



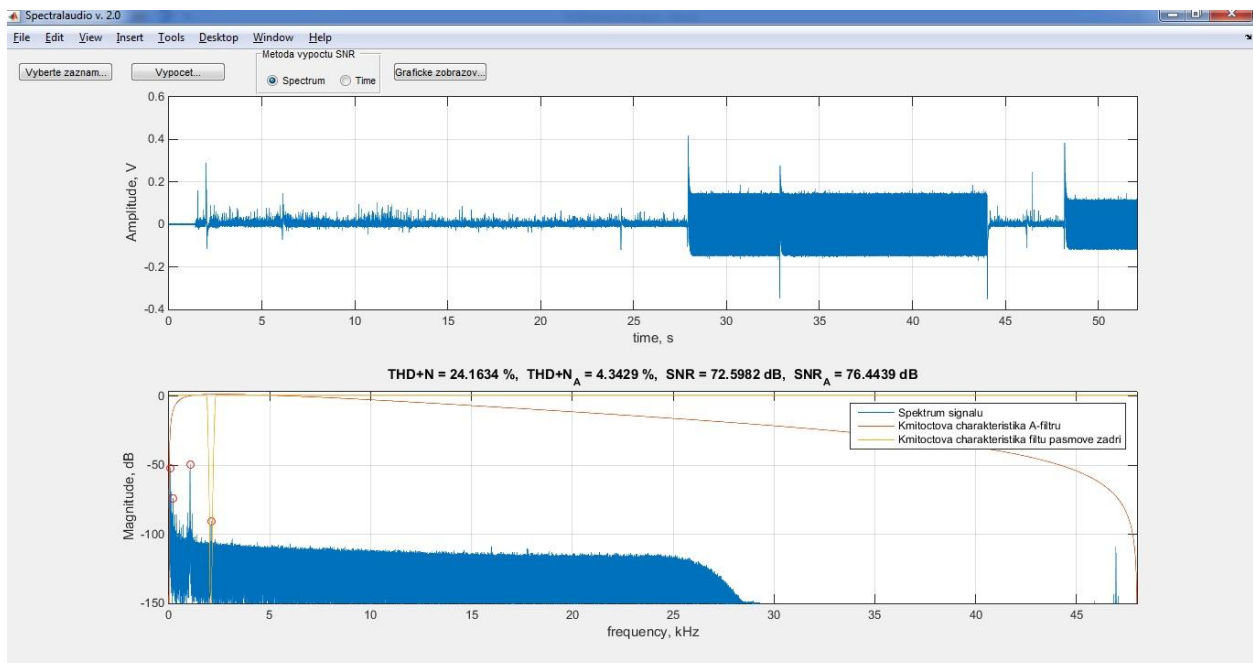


THD a THD+N. Když ověření skončí, vytvoří se pole `str` s příslušnými hodnotami. Celkem můžou nastat 4 případy zobrazování hodnot:

```
if length(SNR_dB)~=1 && strcmp(choice, 'THD+N')
    str = ['THD+N = ' num2str(THD_N(1)) ' %, ', ...
          'THD+N_A = ' num2str(THD_N(2)) ' %, ', ...
          'SNR = ' num2str(SNR_dB(1)) ' dB, ', ...
          'SNR_A = ' num2str(SNR_dB(2)) ' dB'];
elseif length(SNR_dB)~=1 && strcmp(choice, 'THD')
    str = ['THD = ' num2str(THD(1)) ' %, ', ...
          'THD_A = ' num2str(THD(2)) ' %, ', ...
          'SNR = ' num2str(SNR_dB(1)) ' dB, ', ...
          'SNR_A = ' num2str(SNR_dB(2)) ' dB'];
elseif length(SNR_dB)==1 && strcmp(choice, 'THD+N')
    str = ['THD+N = ' num2str(THD_N(1)) ' %, ', ...
          'THD+N_A = ' num2str(THD_N(2)) ' %, ', ...
          'SNR = ' num2str(SNR_dB(1)) ' dB'];
elseif length(SNR_dB)==1 && strcmp(choice, 'THD')
    str = ['THD = ' num2str(THD(1)) ' %, ', ...
          'THD_A = ' num2str(THD(2)) ' %, ', ...
          'SNR = ' num2str(SNR_dB(1)) ' dB'];
End
```

Posledním krokem je určení místa zobrazování hodnot z pole `str`:

`title(str)` – hodnoty se zobrazí jako titulky nad aktuálním grafickým oknem, což je hSA(2).



Obrázek č. 18. Konečné zobrazení všech výsled



## 2.3 Výsledky měření.

### 2.3.1 Popis určitých záznamů.

Posledním krokem bylo měření parametrů analogových záznamů. Celkově bylo analyzováno 9 záznamů. Pro porovnání experimentálních výsledků byl použit software na analýzu záznamů, který byl k dispozici volně, a to SigView v2.8.0. Pomocí tohoto programu byly vypočteny stejné parametry záznamů, přičemž pro porovnání byly použity stejné úseky záznamů jako pro výpočet v Matlab. Měření bylo provedeno pro impulsy o kmitočtu:

analog\_TEST KMIT DKG 1 1940 (optický záznam) – 1044 Hz

profi\_tape(Magnetická páska) – 1 kHz

RES DEFA1PF35\_U\_a – 1141 Hz

RES\_kmitoctovy\_50-10000\_test\_E-UV(optický záznam 40. let) – 1045 Hz

Track 1(gramofonová deska) – 1032 Hz; Track 2,3,4,5(gramofonové desky) – 315 Hz

Výsledky měření parametrů záznamů jsou v následujících tabulkách:

Matlab:

Název nahrávky	THD[%]	THD <sub>A</sub> [%]	SNR[dB]		THD+N[%]	THD + N <sub>A</sub> [%]	SNR <sub>A</sub>
			Metoda výpočtu: Spectr.	Time			
RES_kmitoctovy_50-10000_test_E-UV	0,07	0.04	75.78	14.27	20.8	3.06	61.55
RES DEFA1PF35_U_a	5.4	6.25	71.72	22.1	2.6	1.96	74.43
Profi_tape	0	0.002	72.68	25.88	0.9	2.5	75.39
analog_TEST KMIT DKG 1 1940	0.25	0.31	73.2	20.1	4.1	2.2	75.35
Track01	12.77	9.2	67.8	29	12.2	13.48	59
Track02	0.29	0.003	88.1	20.71	4.05	1.4	90.19
Track03	0.4	0.04	90.94	21.1	4.7	1.69	90.61
Track04	0.04	1.02	91.7	20.76	4.31	2.15	91.5
Track05	0.1	0.5	82.45	21.47	4.2	1.9	83.4

SigView:

Název nahrávky	THD[%]	THD <sub>A</sub> [%]	SNR[dB]	THD+N[%]	THD + N <sub>A</sub> [%]	SNR <sub>A</sub>
RES_kmitoctovy_50-10000_test_E-UV	0.36	0.71	51.88	1.23	1.22	50.29
RES DEFA1PF35_U_a 24-50	14.47	15.19	47.83	14,56	15,4	50.81
Profi_tape	0.13	0.15	48.83	1.03	0.63	50.66
analog_TEST KMIT DKG 1 1940	0.71	6.7	48.15	3.04	8.35	34.3
Track01	14,6	15,87	47,87	15,12	16	50,87
Track02	0.33	0.46	59.51	0.35	0.51	58.21
Track03	0.44	0.35	59.77	0.46	0.38	58.17
Track04	0.81	1.47	60.11	0.83	1.48	59.17
Track05	2.84	2.85	47.6	3.34	3.36	47.57





### 2.3.2 Hodnoty získané v Matlab.

Matlab úspěšně spočítal parametry pro všechny analogové záznamy (tabulka Matlab) a zobrazil kmitočtovou charakteristiku pro každý ze záznamů (příklad obrázek č. ). Při čtení Track 2,3 a 4 programem bylo potřeba změnit počet vzorků, protože samotný záznam obsahoval 2.75 milionů, což je méně než nastavených standardně 5-6 milionů. Kromě toho, je standardně nastaven jenom levý kanál, proto při práci s Track 4 bylo potřeba v řádku 15 a 22 ( $Y = Y(: ; 1)$ ) změnit hodnotu v závorkách z 1 na 2.

THD: výsledné hodnoty THD byly velmi závislé na procesu hledání harmonických signálů v části 2.1.3. Parametry hledání harmonických byly nastaveny tak, aby program hledal harmonické univerzálně ve všech záznamech, tj. pro některé záznamy (například RES DEFA1PF35\_U\_a), kde je velký počet harmonických, program nebral v úvahu ty nejmenší. Opakované měření stejných záznamů ukázalo, že maximální směrodatná odchylka měření činila 1,27 (počet opakování měření je 4).

SNR: tato hodnota byla vypočítaná dvěma metodami – časovou a spektrální. Základním problémem bylo, že výsledky mezi dvěma metodami na stejných úsecích signálů se velmi od sebe lišili, což bylo způsobeno časovou metodou. Detailním pohledem na tuto metodu se dá konstatovat, že návrh této metody nebyl přesný - při sestavování metody výpočtu byla za dominantní hodnotu na intervalu, kde je impuls (užitečný signál), považována úroveň užitečného signálu, tedy pro výpočet byla použita jenom hodnota užitečného signálu a byla porovnána se šumem mimo impulsů. Spektrální metoda byla založena na výpočtu SNR pomocí amplitudy užitečného signálu ( $mag\_dB$ ), a za šum byly považovány nízké hodnoty amplitudy signálu ( $mean(mag\_dB)$ ), což v praxi odpovídá logice. Dá se taky říct, že u záznamů, kde bylo slyšet a vidět ze spektra víc šumu, byly hodnoty SNR menší a naopak, což taky odpovídá logice. Zlepšení přesnosti výpočtů by se mohlo uskutečnit změnou postupu při realizaci pro časovou metodu a upřesnění při výpočtech spektrální metodou. Maximální pozorovaná odchylka pro každou metodu zvlášť: časová metoda – 0,21; spektrální metoda - 0,48

A-filtr: da se konstatovat, že A-filtr byl navržen úspěšně a tvar odfiltrovaného signálu odpovídal předpokladům. Filtr, podle své kmitočtové charakteristiky (obrázek č.) tlačil příliš nízké a vysoké kmitočty a měl nejvyšší úroveň propusti u 1kHz. Porovnáním s A-filtrem se SigView programem bylo možné pozorovat podobnost změn při filtraci.



### 2.3.3 Porovnání s výsledky z SigView.

Tabulka SigView obsahuje výsledky měření pomocí tohoto programu. Pro každý ze záznamů byl vybrán stejný úsek a kmitočet, po čemž byly vypočítány stejné parametry.

Hodnoty THD se neliší o víc než 3 %, kromě zvláštních případů měření, kde jeden z programu udával příliš velkou nebo malou hodnotu. Vyhnout se tomu bylo možné vybráním jiného úseku signálu pro případ chyby v SigView anebo změnou parametrů hledání harmonických pro přesnější výsledek.

Hodnoty SNR se řádově víc liší. Je to způsobeno hlavně nedokonalým přístupem při výpočtech v Matlab. Kromě toho, SigView s největší pravděpodobností počítá tento parametr úplně jinou metodou (matematickým očekáváním)[34],[37]. Jako alternativu pro získání přesnějších výsledků a prodloužení analýzy navrhuji použití a porovnání obou metod (časové a spektrální) jiným, volně dostupným, softwarem.

## 3. Závěr

V průběhu práce jsem byl obeznámen s vlastnostmi analogových záznamů. Pomocí programu Matlab byl vytvořen kód, neboli nástroj, na analýzu předložených analogových záznamů. Po de-



tailním prozkoumání jsem došel k závěru, že každá z těchto nahrávek má odlišné vlastnosti od ostatních. Samotný signál v každém záznamu má různou úroveň šumu a strukturu, dokonce byl každý záznam nahráván jiným způsobem a v různém časovém období, což samozřejmě ovlivnilo kvalitu nahrávání. Toto mělo za důsledek velký počet modifikací výsledného programu, protože cílem práce byl návrh univerzálního kódu. Cíl byl nakonec splněn a univerzality kódu bylo dosaženo.

Při čtení záznamů byla zjištěna jedna ze základních vlastností Matlabu při zpracování dat, a to, že čte data najednou, proto práce tohoto programu s velkým objemem informací může vést k časové náročnosti. Byla pozorována přímá závislost mezi objemem dat a časem potřebným pro výpočet. Každý záznam má určitý počet diskretizačních vzorků a při velkém objemu vzorků nemohl program výpočet dokončit, proto bylo experimentální metodou, neboli metodou zkoušení, vybráno optimální rozmezí počtu vzorků pro daný program, a to 3 až 6 milionů. Rovněž byly experimentální metodou zjištěny hodnoty pro hledání harmonických signálů (výška, šířka a výčnělek), typ a řád filtrů, šířka pásma pro filtr s pásmovou zádrží, a celkové metody realizace tohoto programu. Navržený program bude schopen vypočítat THD, THD+N pro záznam vybraný uživatelem, filtrovat signál pomocí A-filtru a spočítat odstup signál-šum (SNR) dvěma metodami - spektrální a časovou. Kromě toho bylo navrženo základní uživatelské rozhraní, které umožňuje výpočet a grafické zpracování ovládat intuitivně. Při práci s vypočteným a zobrazeným signálem je možné přiblížit část z toho signálu, například jeden impulz, pomocí funkce zoom, a vypočítat hodnoty pouze pro jeden přiblížený kus záznamu.

Hodnoty THD a THD+N jsou závislé na tom, kolik a jaké harmonické najde program během práce a jaká je úroveň šumu signálu. Obecně se dá konstatovat, že hodnoty, které byly získány programem, nejsou přesné, ale odpovídají logice, a navíc jsou srovnatelné s hodnotami, které byly získány programem SigView. Nulové nebo velmi malé hodnoty THD a THD+N byly způsobeny malým počtem získaných harmonických, anebo v případě nulové hodnoty, program našel jenom jednu harmonickou odpovídající požadavkům hledání.

Vypočtené hodnoty SNR se liší v závislosti na použité metodě. Příčinou je použití dvou absolutně různých postupů při výpočtu SNR - realizace se provádí dvěma matematickými operacemi, které ve výsledku dávají odlišné hodnoty. Z matematického pohledu je realizace obou metod správná, ale předpokládanou příčinou odlišností je řada faktorů, např. vlastnosti záznamů a nahodilé chování signálů a šumů. Ve výpočtech chybí flexibilita, tedy matematický model výpočtů, který by zohlednil všechny detaily záznamu. U porovnání hodnot SNR s výsledky z programu SigView jsou odlišnosti způsobeny díky tomu, že profesionální programové nástroje většinou počítají SNR přes jiné metody, například tzv. matematické očekávání. Program tedy nerozlišuje, kde je užitečný signál a kde je šum, ale počítá podle „očekávání“.

Chyby v práci kmitočtová charakteristika v logaritmickém měřítku, ale da se to nakreslit lineárně použitím závislosti amplitudy na kmitočtu.

Navržený program může být použit pro výpočet THD a THD+N, kreslení kmitočtové charakteristiky a filtrace signálů A-filtrem. SNR se počítá taky, ale přesnost výpočtu má být zlepšená v budoucnu. Program je zcela funkční a může být použit pro nalezení parametrů analogových záznamů.

## 4. Seznam použité literatury :



- [1]. Audio products for professionals “Audio specifications” Dennis Bohn, Rane Corporation. Dostupný z : <http://www.rane.com/note145.html>
- [2]. THD, Wikipedie otevřená encyklopedie, dostupné z <http://cs.wikipedia.org/wiki/THD>
- [3]. A-weighting, Science and engineering encyclopedia Dirac Delta Consultants Ltd. Dostupné z [http://www.diracdeltaco.uk/science/source/a/w/aweighting/source.html#.VODSJ\\_mG8dz](http://www.diracdeltaco.uk/science/source/a/w/aweighting/source.html#.VODSJ_mG8dz)
- [4]. Ф.Е.Евдокимов. Теоретические основы электротехники М., Академия 2004 стр. 262  
F.E. Jevdokimov. Teoretické základy elektrotechniky M., Akademie 2004, str.262
- [5]. Политех в сети, спектральный анализ периодических сигналов  
Poitech v síti , spektrální analýza periodických signálů. Dostupné z <http://www.webpoliteh.ru/subj/ore/388-2-2-spektralnyj-analiz-periodicheskih-signalov.html>
- [6]. Noise figure and SNR. Jim Stiles, University of Kansas. 10/26/2006. Dostupné z <http://www.ittc.ku.edu/~jstiles/622/handouts/Noise%20Figure%20and%20SNR.pdf>
- [7]. Основы спектрального анализа. Základy spektrální analýzy. Autor není ukázán.
- [8]. Analog devices, MT-003 Tutorial, by Walt Kester, dostupné z <http://www.analog.com/media/en/training-seminars/tutorials/MT-003.pdf>
- [9]. Антонию А. - Цифровые фильтры. Анализ и проектирование. – 1983  
Antonju.A – digitální filtry. Analýza a design – 1983.
- [10]. Мирский Г.Я. - Радиоэлектронные измерения, 3-е изд. – 1975  
Mirský.G.J. – radioelektrické měření, 3-verze – 1975.
- [11]. Букингем М. - Шумы в электронных приборах и системах – 1986  
Bukingem.M – šum v elektrických přístrojích a systémech – 1986.
- [12]. Webová stránka podpory Matlabu mathworks . Dostupné z <http://www.mathworks.com/>
- [13]. В.П.Дьяконов. Матлаб 7,самоучитель.  
Matlab 7, Kniha pro samouky.DMK press, 2008.
- [14]. Цифровые фильтры. Digitální filtry. Autor není ukázán.
- [15]. Total Harmonic Distortion and Effects in Electrical Power Systems. Associated Power Technologies. Dostupné z <http://www.aptsources.com/resources/pdf/Total%20Harmonic%20Distortion.pdf>
- [16]. Harmonic distortion in the electric supply system, Prof. V.J. Gosbell, University of Wollongong. Dostupné z <http://sound.westhost.com/lamps/technote3.pdf>
- [17]. Recommendations for Successful VoIP Surveys. Dostupné z [http://www.cisco.com/c/en/us/td/docs/wireless/technology/7920/site\\_survey/guide/7920ssg/survov r.html](http://www.cisco.com/c/en/us/td/docs/wireless/technology/7920/site_survey/guide/7920ssg/survov r.html)
- [18] Баскаков С.И. Радиотехнические цепи и сигналы. М.: Высшая школа, 2-е издание  
Baskakov S.I. Radiotechnické obvody a signály. M. Vyšší škola , 2-é vydání.
- [19] Классификация сигналов. Аналоговые сигналы. Klasifikace signálů. Analogové signály. Dostupné z <http://www.globalinformatics.ru/gloins-1132-1.html>
- [20] А.А.Ханова, И.Г. Макарова. Спектральный анализ и синтез.А.А.Chanova, I.G.Makarova  
Spektrální analýza a syntéza. Dostupné z <http://www.exponenta.ru/educat/systemat/hanova/lab/LR8/LR8.asp>
- [21] Sound recording and reproduction , Dostupné z [https://en.wikipedia.org/wiki/Sound\\_recording\\_and\\_reproduction](https://en.wikipedia.org/wiki/Sound_recording_and_reproduction)



- [22] Запись и воспроизведение звука . Оптический , механический и магнитный методы записи звука. Natáčení a reprodukce zvuku. Optické, mechanické a magnetické natáčení zvuku. Dostupné z [http://meduniver.com/Medical/otorinolaringologia\\_bolezni\\_lor\\_organov/231.html](http://meduniver.com/Medical/otorinolaringologia_bolezni_lor_organov/231.html)
- [23] Princip optického záznamu zvuku. .Dostupné z <http://fyzika.jreichl.com/main.article/view/1346-princip-optickeho-zaznamu-zvuku>
- [24] Запись и воспроизведение информации. Natáčení a reprodukce informace. Dostupné z [http://dic.academic.ru/dic.nsf/enc\\_tech/435/%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D1%8C](http://dic.academic.ru/dic.nsf/enc_tech/435/%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D1%8C)
- [25] Theory of magnetic recording, H. Neal Bertram. Cambridge University, 1994.
- [26]. Титце У., Шенк К. - Полупроводниковая схемотехника. Том I 2007  
Titce U., Šenk K. – polovodičová obvodotechnika, Č.1 [2007]
- [27] Физика шумов и флуктуаций параметров. А.В. Якимов.2013 Fyzika šumů a fluktuací parametrů, A.V. Jakimov. 2013
- [28] Амплитудно-частотная характеристика(АЧХ). Kmitočtová charakteristika . Dostupné z <http://radiolubitel.net/index.php/elektronika/265-amplitudno-chastotnaya-karakteristika-achkh>
- [29] The difference between measuring DR and THD+N .Dostupné z <https://blogs.msdn.microsoft.com/audiofool/2006/12/06/the-difference-between-measuring-dr-and-thdn/>
- [30] Tepelný šum. Dostupné z [https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BF%D0%BB%D0%BE%D0%B2%D0%BE%D0%B9\\_%D1%88%D1%83%D0%BC](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BF%D0%BB%D0%BE%D0%B2%D0%BE%D0%B9_%D1%88%D1%83%D0%BC)
- [31] Vystřelový šum. Dostupné z [https://ru.wikipedia.org/wiki/%D0%94%D1%80%D0%BE%D0%B1%D0%BE%D0%B2%D0%BE%D0%B9\\_%D1%88%D1%83%D0%BC](https://ru.wikipedia.org/wiki/%D0%94%D1%80%D0%BE%D0%B1%D0%BE%D0%B2%D0%BE%D0%B9_%D1%88%D1%83%D0%BC)
- [32] A/D převodník. Dostupné z [https://cs.wikipedia.org/wiki/A/D\\_p%C5%99evodn%C3%ADk#Kvantiza.C4.8Dn.C3.AD\\_.C5.A1um](https://cs.wikipedia.org/wiki/A/D_p%C5%99evodn%C3%ADk#Kvantiza.C4.8Dn.C3.AD_.C5.A1um)
- [33]. Шумы квантования. Kvantizační šумы. Dostupné z <http://studopedia.org/2-115472.html>
- [34] Signal-to-noise ratio. Dostupné z [https://en.wikipedia.org/wiki/Signal-to-noise\\_ratio](https://en.wikipedia.org/wiki/Signal-to-noise_ratio)
- [35] Fast Fourier Transform. Dostupné z [https://en.wikipedia.org/wiki/Fast\\_Fourier\\_transform](https://en.wikipedia.org/wiki/Fast_Fourier_transform)
- [36] Discrete Fourier Transform. Dostupné z [https://en.wikipedia.org/wiki/Discrete\\_Fourier\\_transform](https://en.wikipedia.org/wiki/Discrete_Fourier_transform)
- [37] Цифровая обработка сигналов .А.Б.Сергиенко. Digitalní zpracování signlů. А.В.Sergienko 2002.