

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

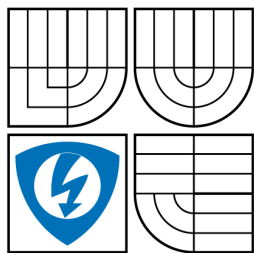
ANALÝZA AKUSTICKÉHO SIGNÁLU PRO DETEKCI RYCHLOSTI
VOZIDEL

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

TEREZA KONÍČKOVÁ

BRNO 2009



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Studentka: Tereza Koníčková

ID: 72964

Ročník: 3

Akademický rok: 2008/2009

NÁZEV TÉMATU:

Analýza akustického signálu pro detekci rychlosti vozidel

POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce bude seznámení se s problematikou měření a analýzy akustického signálu jedoucích vozidel k určení jejich rychlosti. Práce předpokládá rešerši v této oblasti, návrh a seznámení s vhodným měřicím zařízením, naměřením a následnou analýzu potřebných dat. Jednotlivé kroky konzultujte s vedoucím práce.

DOPORUČENÁ LITERATURA:

Dle doporučení vedoucího práce.

Termín zadání: 9.2.2009

Termín odevzdání: 1.6.2009

Vedoucí práce: Ing. Stanislav Klusáček

prof. Ing. Pavel Jura, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

ABSTRAKT

Cílem bakalářské práce je vytvoření programu pro analýzu zvukového signálu projíždějícího automobilu s cílem určení rychlosti. Práce popisuje jednotlivé metody měření rychlosti automobilu a předkládá novou metodu měření rychlosti – využití Dopplerova principu při analýze zvukového signálu. Práce ověřila, že je tuto metodu k určení rychlosti vozidla možno použít.

Klíčová slova : rychlost, měření, Dopplerův princip, FFTW

ABSTRACT

The aim of this Bachelor thesis is to create software for sound signal analysis of a passing car with the aim to determine the driving speed. This work describes single methods of a car velocity measurement and it is submitting a new method of velocity measurement with using the Doppler principle at sound signal analysis. My work proved, that it is possible to use this method to determine the speed of vehicles.

Keywords: speed, measurement, Doppler principle, FFTW

Bibliografická citace: KONÍČKOVÁ, T. *Analýza akustického signálu pro detekci rychlosti vozidel*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 45 s. Vedoucí bakalářské práce Ing. Stanislav Klusáček.

PROHLÁŠENÍ

„Prohlašuji, že svou bakalářskou práci na téma *Analýza akustického signálu pro detekci rychlosti vozidel* jsem vypracovala 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 neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne: **1. června 2009**

.....

Tereza Koničková

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce Ing. Stanislavu Klusáčkovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce. Dále děkuji pracovníkům firmy RAMET C.H.M. a.s. za možnost provedení kontrolních měření.

V Brně dne: **1. června 2009**

.....

Tereza Koníčková

OBSAH

1. ÚVOD	7
2. MĚŘENÍ RYCHLOSTI VOZIDEL	8
2.1 Typy používaných měření	9
2.1.1 RAMER	9
2.1.2 Laser	10
2.1.3 Mýtné brány, úsekové měření	12
2.1.4 Analýza akustického signálu	13
3. DOPPLERŮV PRINCIP	14
4. AKUSTICKÝ SIGNÁL	16
5. FOURIEROVA TRANSFORMACE	18
5.1 Knihovna FFTW	18
6. ALGORITMUS ANALÝZY ZVUKOVÉHO ZÁZNAMU	21
6.1 Blokové schéma programu	23
7. POPIS PROGRAMU	28
7.1 Pracovní verze	28
7.2 Měřicí verze	29
7.3 Odhad chyby výpočtu	31
8. VÝSLEDKY MĚŘENÍ	32
9. ZÁVĚR	35
10. LITERATURA	36
11. SEZNAM ZKRATEK	37
12. SEZNAM OBRÁZKŮ	38
13. PŘÍLOHA	39
13.1 Výpis programu Rychlost – měřicí verze	39

1. ÚVOD

Stránky novin i internetových diskusí zaplňují úvahy o měření rychlosti automobilů. Čtenáři diskutují, zda tuto pravomoc svěřit městské polici, nebo jen státní, zda povolit antiradary a co je vůbec možno považovat za antiradar, jak a kde vůbec rychlost automobilů měřit.

Svou práci bych chtěla ověřit, zda je možné zjistit rychlost projíždějícího automobilu pouze a jen ze zvukového signálu, který vysílá každý jedoucí automobil. Takže žádný radar, žádný laser, jen pasivní „poslouchač“ někde u cesty, který analýzou zvuku projíždějícího automobilu určí jeho rychlost.

Výsledkem práce by měl být počítačový program, který zaznamená a vyhodnotí zvuk projíždějícího automobilu a vypočítá jeho rychlost.

2. MĚŘENÍ RYCHLOSTI VOZIDEL

Základní měření rychlosti vozidla provádí každý řidič sám, rychloměrem, který je součástí vozidla. Pro přesnost měření tímto rychloměrem odkazuje český právní řád, jak je uvedeno v [4], na mezinárodní předpisy, a to konkrétně na předpis EHK/OSN č. 39 a evropskou směrnici 75/443/EHS, ve znění pozdějších směrnic.

Z uvedených norem vyplývá, že:

- Rychlost udávaná rychloměrem nesmí být nikdy nižší než skutečná rychlost. Je stanovena i maximální odchylka, která určuje, že např. při údajích rychloměru 50 km/h je minimální skutečná rychlost 41 km/h.
- Je přesně stanoven vzorec, podle kterého se přípustná odchylka rychloměru určuje

$$0 \leq v_1 - v_2 \leq \frac{v_2}{10} - 4 \frac{km}{h}$$

v_1 – údaj rychloměru
 v_2 – skutečná rychlost

Zákonem č. 361/2000 Sb. o silničním provozu je stanovena v § 18 maximální rychlost, kterou se vozidlo může pohybovat, např. pro vozidla do 3500 kg jsou tato omezení v obci 50 km/h, mimo obec 90 km/h a na dálnici 130 km/h.

Každé zařízení, které Policie ČR, vojenská policie nebo obecní police používá k měření rychlosti vozidel, musí odpovídat zákonu č. 505/1990 Sb., o metrologii, a jeho prováděcím vyhláškám. Radary musí mít platné typové schválení a každý jednotlivý kus musí být pravidelně ověřován.

Radary jsou v ČR schvalovány s jednotnou odchylkou ± 3 km/h, pokud je zjištěná rychlost do 100 km/h, respektive ± 3 procenta při rychlostech nad 100 km/h. Tato odchylka není promítnuta do zobrazené rychlosti na radaru, zohlednit ji musí sám policista nebo strážník. Teoreticky může u řádně fungujících radarů nastat situace, že vozidlu, které objektivně jede rychlostí 50 km/h, ukáže jeden radar rychlost 47 km/h a jiný 53 km/h.

2.1 TYPY POUŽÍVANÝCH MĚŘENÍ

2.1.1 RAMER

Silniční radary měří rychlost pomocí Dopplerova jevu s využitím odražených radiových vln v mikrovlnném pásmu. Paprsek vysílaný parabolickou anténou se odráží od karoserie projíždějících vozidel zpět k radaru, je zachycen anténou a zesílen. Zpracováním odraženého signálu elektronickými obvody je vypočtena rychlost projíždějícího vozidla.

Mikrovlnné záření vysílané radarem se šíří přímo a chová se podobně jako světlo. Prochází sklem a plastickými hmotami. Radarové vlny se odrážejí od kovových částí karosérie a některých terénních překážek, jako jsou například svodidla, sloupy veřejného osvětlení a železobetonové konstrukce.

Policie České republiky používá typ radarů RAMER (radarový měřič rychlosti) od firmy RAMET C.H.M. a.s. [7] Radar pracuje v mikrovlnném pásmu s frekvencemi 34.0 GHz a 34.3 GHz. Paprsek vysílaný radarem je velmi úzký (úhel 5°), radary mají velmi nízký vysílací výkon (0,3 mW, max. 2 mW), který ztěžuje možnost odhalení. Ve srovnání s ostatními světovými výrobci mají české radary špičkové technické parametry. Dosah radaru je 60 metrů; obvykle se ale měří na vzdálenost do 35-ti metrů. Překročení rychlosti je zdokumentováno videokamerou a snímek je přenesen do počítače. Pro měření v noci je radar doplněn bleskem.

Radar typu RAMER je nejčastěji zabudován do civilních neoznačených automobilů. Radarová hlava je zasazena do přední masky automobilu, počítač s displejem a řídicí panel jsou před sedadlem spolujezdce. Takzvané stacionární (pevně stojící) radary jsou umístěny v kovových skříních podél silnic, jiná varianta může být umístěna u silnice pouze na stativu.

↓135km/h 0000013316 09:01:22 04.05.2009



Obrázek 1 Dokumentační snímek z radaru AD 9 C

Na obrázku 1 je výstup z měřicího zařízení RAMER AD 9 C, se kterým jsem měla možnost ověřovat fungování svého programu. Na spodním okraji snímku jsou nastavení kamery, 20 km/h znamená, že budou zaznamenávána všechna vozidla, která překročí tuto rychlost. Na horním okraji snímku je šipkou naznačen směr jízdy vozidla (vozidlo přijíždějící k radaru). Radar může měřit podle nastavení buď oba směry, nebo jen jeden zvolený. Následuje změřená rychlost. Dalším údajem je číslo snímku. Číslo tvoří souvislou vzestupnou řadu a není možné přestupek ze systému vymazat bez zanechání „prázdného místa“. Nezbytnými údaji jsou čas a datum měření. Ze snímku byla odstraněna poznávací značka vozidla, na originálu je samozřejmě čitelná.

2.1.2 Laser

Jiný typ zařízení se anglicky nazývá LIDAR - LIght Detection And Ranging – (měření vzdálenosti pomocí odrazu světelného paprsku).

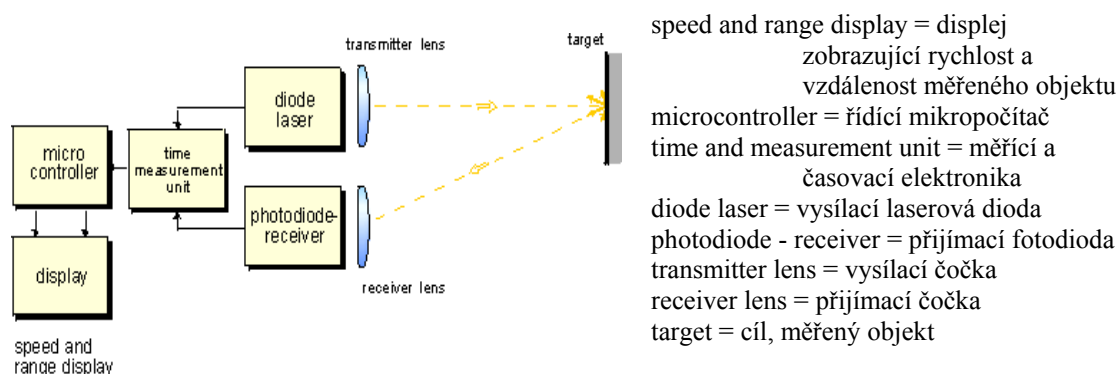


Obrázek 2 Ruční laserový měřič rychlosti ProLaser II od firmy Kustom Signals, Inc. – převzato z [5]

Jak vlastně laserový měřič rychlosti pracuje? Základem je obyčejný polovodičový laser, který září v neviditelné – infračervené části spektra. Laser nepracuje spojitě, ale paprsek je vysílán ve velmi krátkých intervalech – pulzech.

Pokud paprsek laseru namíříme na nějaký lesklý objekt, odrazí se od něj zpět. Odražený paprsek je zachycen pomocí optiky a světlocitlivý prvek - fotodioda jej přemění na elektrický signál.

Světelný záblesk se pohybuje rychlostí 300 000 km/s. Čím je vzdálenost objektu, od kterého se laserový paprsek odrazí, větší, tím delší je doba mezi vysláním a příjmem pulzu. Vzdálenost objektu je přímo úměrná změřenému intervalu mezi dvěma pulzy – vysílacím a přijímacím. Co se stane, když se měřený objekt – automobil – začne pohybovat. Vzdálenost a tím i intervaly mezi pulsy se postupně zkracují, jak se k nám cíl přibližuje. Není pak nic jednoduššího, než změřit vzdálenost, zlomek vteřiny počkat a znovu změřit vzdálenost. Pak už snadno vypočítáme rychlost, ale to za nás udělá malý počítač, který je součástí každého laserového měřiče rychlosti.



Obrázek 3 Princip činnosti laserového měřiče rychlosti – převzato z [5]

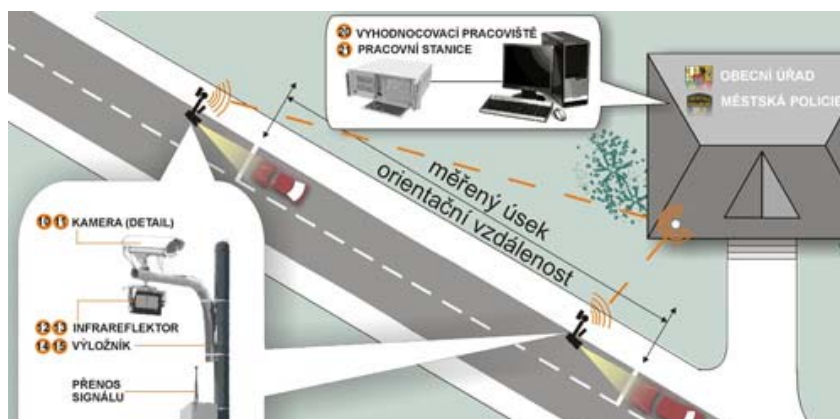
Laserový měřič rychlosti má tvar pistole nebo pušky (může být ale také umístěn na stativu), kterou operátor namíří na vybraný automobil tak, aby jej měl v záměrném bodu hledáčku. Měřicím místem je lesklá část karoserie – nejčastěji reflektor nebo SPZ. Stiskem spouště je zařízení uvedeno v činnost a během zhruba půl sekundy vozidlu změřena rychlost. Údaj o rychlosti a někdy i vzdálenosti se

vzápětí objeví na displeji. Policista si pak na řidiče, který překročil rychlost, jednoduše počká a přestupek vyřeší na místě.

Laserové měřiče rychlosti jsou velmi přesné – běžná tolerance při měření rychlosti je lepší $\pm 1,5$ km/h. Také vzdálenost, na kterou je možné měřit rychlost je velmi vysoká. Závisí na konkrétním typu zařízení: většinou se měří mezi 50 až 400 metry. Při dostatečně výkonném laseru a podepření stativem může být vzdálenost klidně až 1 km.

Ve většině případů se měří rychlost přijíždějících vozidel, to znamená zpredu. Úhel, pod kterým se měří rychlost je velmi blízký nule. Při měření rychlosti laserem je nutné, aby operátor našel cíl ručně. Standardní lasery není možné použít v automatických zařízeních, které snímkuji neukázněné řidiče na místech se sníženou rychlostí.

2.1.3 Mýtné brány, úsekové měření



Obrázek 4 Princip úsekového měření – převzato z [9]

Pro měření rychlosti tímto způsobem je využíváno statických kamer, rozmístěných ve známých vzdálenostech od sebe. Není měřena okamžitá rychlost, ale průměrná rychlost v daném úseku (od několika desítek metrů – v městě, po několik kilometrů – mýtné brány na dálnici). Kamerový systém zaznamená čas

a identifikuje vozidlo na obou branách a podle jednoduchého vzorce $v = \frac{s}{t}$ vypočítá ze vzdálenosti kamer (s) a dobu průjezdu vozidla (t) jeho průměrnou rychlost. Přesnost měření je velká, zvláště u delších měřených úseků.

2.1.4 Analýza akustického signálu

V této práci se pokouším použít úplně odlišnou metodu, založenou na analýze záznamu akustického signálu projíždějícího automobilu s využitím Dopplerova jevu.

3. DOPPLERŮV PRINCIP

Dopplerův princip je popsán v každé základní učebnici fyziky [8]. Byl objeven v roce 1842 profesorem pražské techniky Christianem Dopplerem a týká se vlivu pohybu zdroje a přijímače vlnění na jeho frekvenci. Pohybuje-li se zdroj zvuku, přijímač vnímá vysílané frekvence jinak, než v případě, že se zdroj nepohybuje.

Na obrázku 5 je případ, kdy se zdroj pohybuje směrem v přijímači, dochází z „zhušťování vln“, frekvence vlnění se zvětšuje. Při vzdalování se zdroje je naopak

frekvence vlnění menší, zvukové vlny jsou „naředěny“.

Pro výpočet frekvence přijímaného vlnění platí

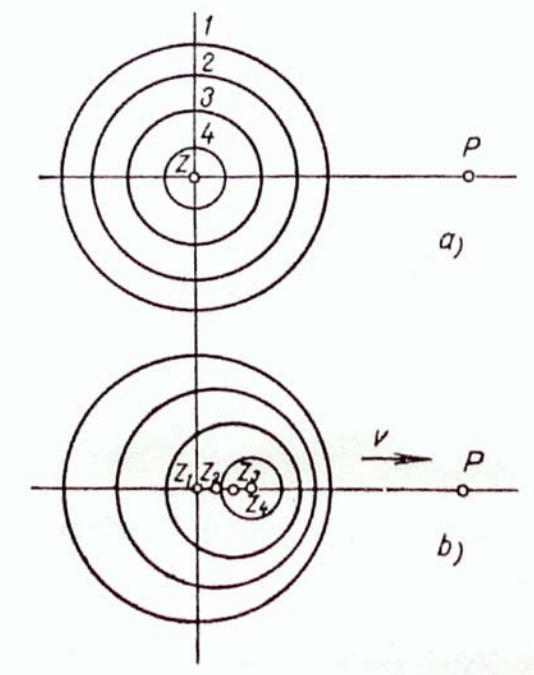
$$f_1 = \frac{f_0}{1 - \frac{v}{c}} \quad (1)$$

pro přibližující se zdroj,

$$f_2 = \frac{f_0}{1 + \frac{v}{c}} \quad (2)$$

pro vzdalující se zdroj,

kde f_0 je frekvence zdroje, v je rychlost zdroje (v našem případě automobilu) a c je rychlost zvuku ve vzduchu.



Obrázek 5 Šíření vln z klidného a pohybujícího se zdroje – převzato z [8]

V ideálním plynu pro rychlost zvuku platí pro závislost na teplotě t vzorec z [8]

$$c = \sqrt{\kappa \frac{p_0}{\rho_0}} \left(1 + \frac{1}{2} \gamma \cdot t \right) \quad (3)$$

kde p_0 je tlak plynu při teplotě 0°C , ρ_0 příslušná hustota, κ Poissonova konstanta a γ součinitel teplotní rozpínivosti plynu.

Ze vzorce pro rychlost zvuku v ideální plynu vyplývá, že pro rychlost zvuku v suchém vzduchu platí následující vztah:

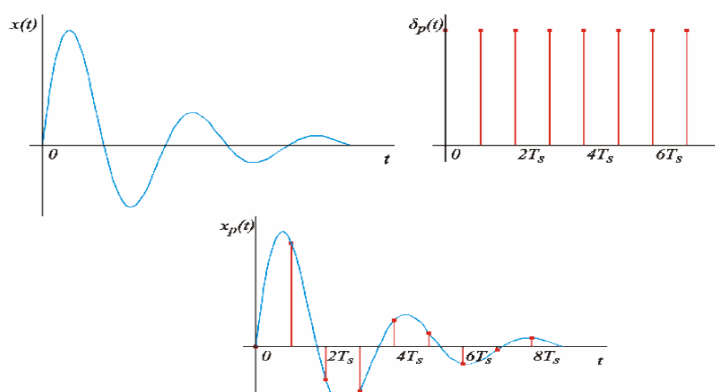
$$c = (331,82 + 0,61 t) \text{ m.s}^{-1} \quad (4)$$

Pro teplotu 15 stupňů Celsia je možno počítat s rychlostí asi 341 m/s.

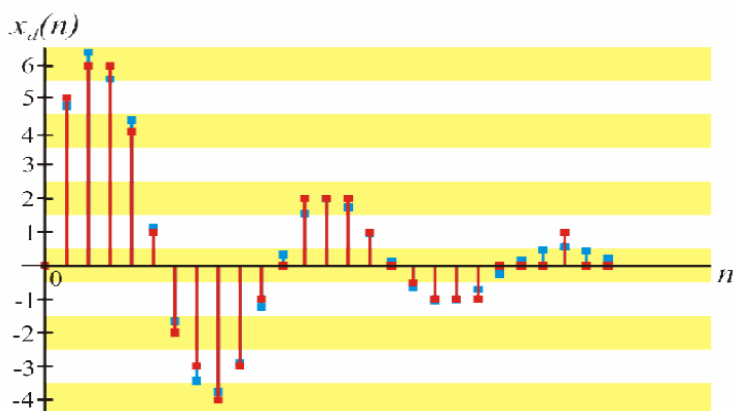
Rychlost zvuku závisí kromě na teplotě také na vlhkosti vzduchu, protože hustota vlhkého vzduchu je menší než hustota suchého vzduchu. Protože ale také Poissonova konstanta κ vlhkého vzduchu je menší, není závislost rychlosti zvuku na vlhkosti příliš výrazná. Při padesátiprocentní vlhkosti je o 0,1% větší a při vlhkosti 100% o 0,2% větší než v suchém vzduchu téže teploty. V programu jsem závislost rychlosti zvuku na vlhkosti neuvažovala.

4. AKUSTICKÝ SIGNÁL

Akustický signál vysílaný jedoucím automobilem je signálem spojitým. Pro zpracování v počítači je tento signál vzorkován a kvantován.[2] Vzorkování je proces, kdy ve stanovených časových okamžicích přiřazujeme signálu $x_p(t)$ určité funkční hodnoty $x(t)$. Na obrázku 6 je znázorněno vzorkování jako násobení spojitého signálu posloupností Dirackových impulsů.



Obrázek 6 Vzorkování spojitého signálu - převzato z [2]



Obrázek 7 Kvantování signálu - převzato z [2]

Kvantování je proces, při kterém se v amplitudě spojitý signál převádí na diskrétní. Při převodu diskretizovaného signálu $x_p(t)$ na jeho digitální reprezentaci

$x[n]$ je nutno převést všechny funkční hodnoty na menší, konečný, počet úrovní.
Např. v případě 8-mi bitového WAV souboru je ukládáno 256 úrovní.

Na obrázku 7 je příklad kvantování signálu, modře je původní signál, červeně signál kvantovaný. Žluté a bílé pruhy udávají jednotlivé kvantizační intervaly.

5. FOURIEROVA TRANSFORMACE

Každý signál je možno reprezentovat v časové oblasti, ale také v oblasti frekvenční. Podstatou Fourierovy transformace je, že funkci v časové oblasti přetransformuje do oblasti frekvenční a získáme tak frekvenční spektrum. Fourierova transformace vychází z předpokladu, že každý signál lze vyjádřit jako superpozici nekonečně mnoha sinusových signálů. Výsledkem Fourierovy transformace jsou Fourierovy koeficienty, které nám říkají, jak daný sinusový průběh přispívá k celkovém signálu.

Také pro diskrétní signál je možno použít Fourierovu transformaci a jak je popsáno např. v [1], existuje několik metod pro výpočet Fourierovy transformace. Ve svém programu jsem použila již existující knihovnu FFTW. [6]

5.1 KNIHOVNA FFTW

Jedná se o soubor funkcí pro výpočet rychlé Fourierovy transformace, byla naprogramována na Massachusetts Institute of Technology. Název je zkratkou **F**astest **F**ourier **T**ransform in the **W**est, takže je zbytečné pokoušet se naprogramovat něco lepšího. Knihovna umožňuje pracovat s komplexními čísly a s vícerozměrnými daty, já používám jen jednorozměrnou DFT s reálnými daty. Knihovna je šířena pod GPL licenci, je ji možno použít za podmínek této licence. Na použití knihovny FFTW mě navedla práce [2].

Použití knihovny je jednoduché, v programu je nutné definovat jen vstupní a výstupní proměnné (in, out), fftw_plan a velikost N, o vše ostatní se už postará knihovna sama. Příklad použití je v následujícím textu:

```
#include "fftw3.h"

int main (int argc, char *argv[]){
    int N;                                // pocet hodnot
    double *in, *out;                     // vstupni a vystupni hodnoty
```

```
fftw_plan p;                                // pracovní promenna

in = (double*) fftw_malloc(sizeof(double) * N);    // alokace pameti pro vstup
out = (double*) fftw_malloc(sizeof(double) * N);    // alokace pameti pro vystup

/* zde je nutno načíst vstup - in */

p = fftw_plan_r2r_1d(N, in, out, FFTW_FORWARD, FFTW_ESTIMATE); // nastavení
                                                                    parametru
fftw_execute(p);                                // provedení výpočtu

/* zde je k dispozici výstup - out */

fftw_destroy_plan(p);                          // uvolnění pameti
fftw_free(in);
fftw_free(out);

return(0);

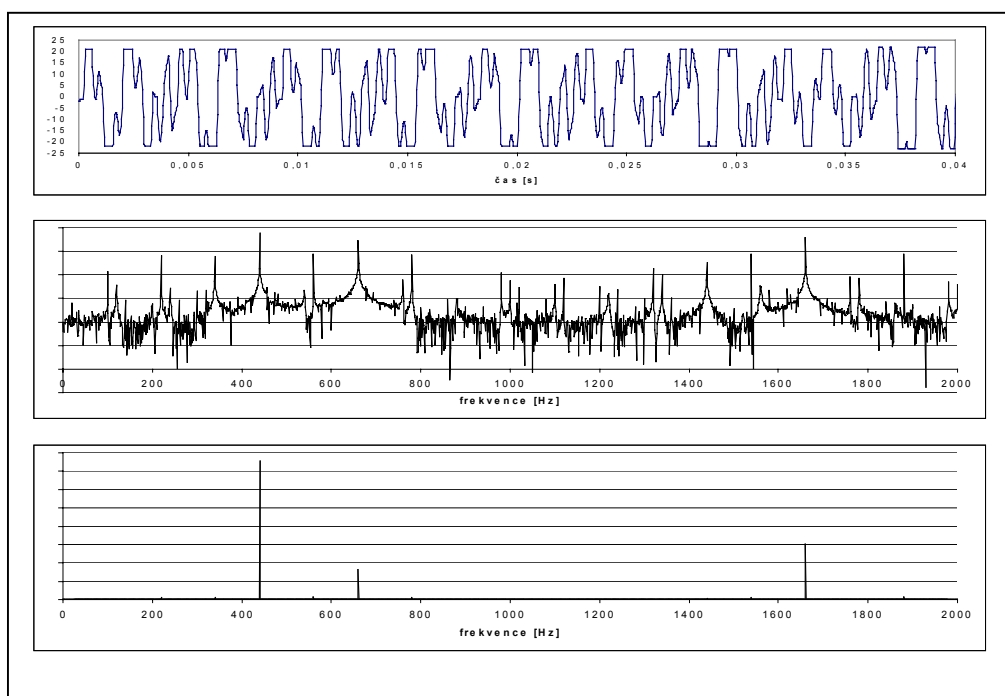
}
```

Použité funkce:

fftw_malloc – alokuje paměť pro proměnnou použitou pro výpočet, podporuje zarovnávání pro rychlejší výpočet s využitím SIMD instrukcí, paměť je nutno uvolňovat funkcí **fftw_free**

fftw_plan_r2r_1d – tato funkce inicializuje proměnnou `fftw_plan`, jako vstupní parametr je počet vzorků `n` a proměnná `in`, `r2r` v názvu znamená, že jak vstupní tak výstupní hodnoty jsou reálná čísla (knihovna obsahuje i funkce `r2c`, `c2c` pro výpočty s komplexními čísly), `1d` v názvu funkce znamená jednorozměrnou funkci, k dispozici jsou i `2d` a `3d` funkce.

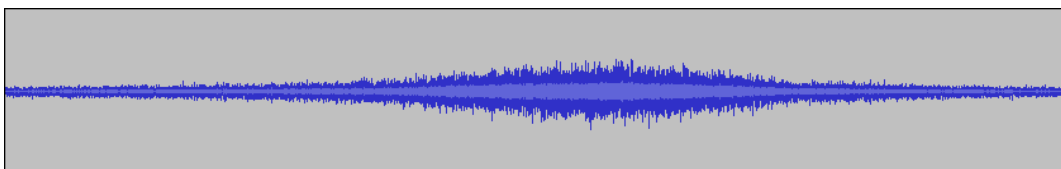
fftw_execute – provádí samotný výpočet transformace



Obrázek 8 Fourierova transformace

Příklad použití je na obrázku 8. V horní části obrázku je ukázka uměle vytvořeného signálu. Smíchala jsem v programu Audacity tři sinusové průběhy o frekvencích 440, 660 a 1660 Hz. Výsledný signál má ještě trochu ořezaná maxima, aby to měl program s analýzou trochu těžší. Na prostředním obrázku je zobrazeno vypočítané frekvenční spektrum, na ose y jsou hodnoty vynášeny v logaritmické stupnici. Situace se zpřehlední, vyneseme-li hodnoty v lineární stupnici (spodní třetina obrázku). To na nás „vyskočí“ právě jen ty tři frekvence – 440, 660 a 1660 Hz. Na podobných souborech jsem si ověřovala, že můj program „dobře“ počítá, že umí nalézt frekvence, které se v signálu opravdu vyskytují.

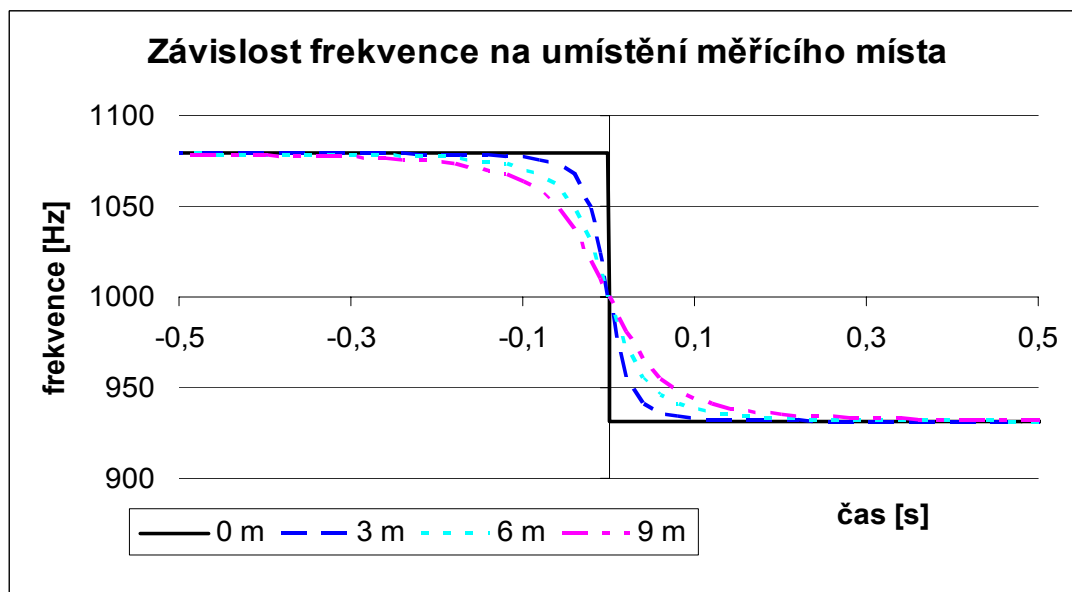
6. ALGORITMUS ANALÝZY ZVUKOVÉHO ZÁZNAMU



Obrázek 9 Zvukový záznam projíždějícího automobilu

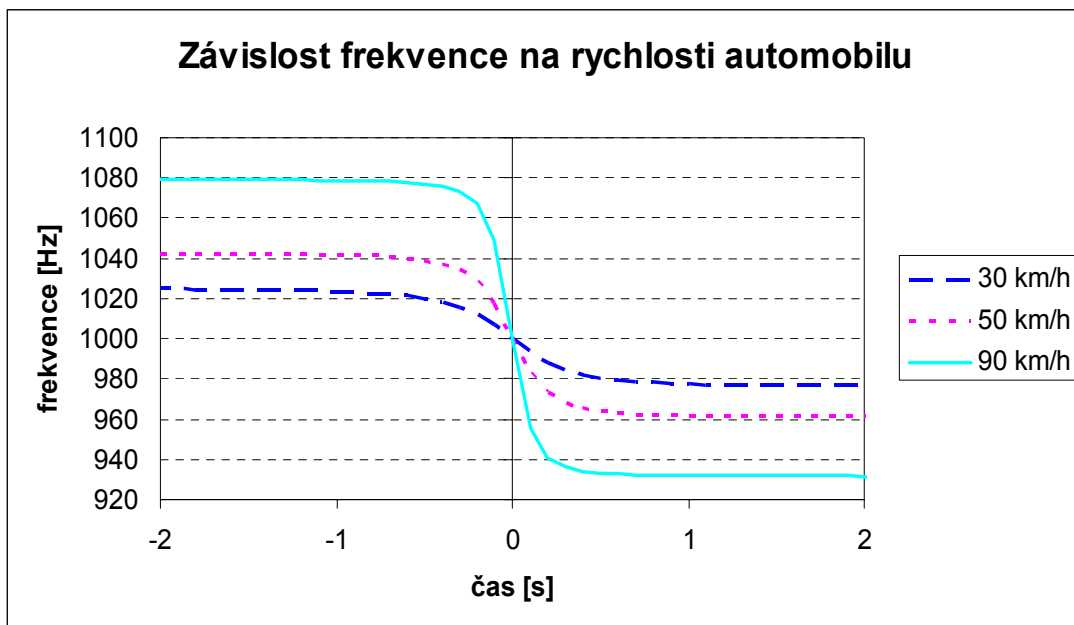
Na obrázku 9 je ukázka zvukového záznamu projíždějícího automobilu. Intenzita zvuku je maximální, když automobil míjí měřicí stanoviště. Nejdříve je nutné určit ze záznamu maximum intenzity záznamu, protože to určuje okamžik průjezdu automobilu kolem měřicího bodu.

V případě, že by mikrofon zaznamenávající průjezd automobilu byl umístěn v dráze vozidla, byla by změna frekvencí před průjezdem a po průjezdu skoková. Protože ale v reálné situaci je měřicí bod (mikrofon) vždy vzdálen několik metrů od dráhy vozidla, projeví se to „zhlazením“ průběhu. Pro několik vzdáleností a klidovou frekvenci 1000 Hz je situace vynesena v grafu na obrázku 10.



Obrázek 10 Průběh změny frekvence pro různé vzdálenosti měřicího bodu

Na průběh změny frekvence má také vliv rychlost vozidla, několik variant je zakresleno na obrázku 11.

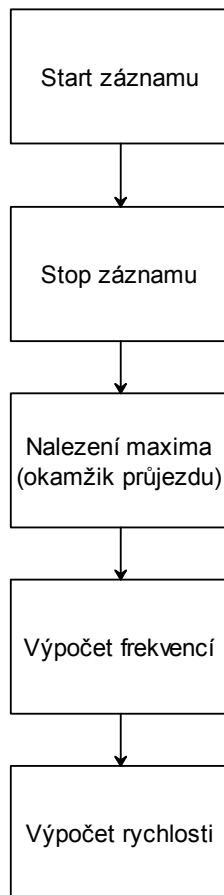


Obrázek 11 Průběh změny frekvence pro různé rychlosti

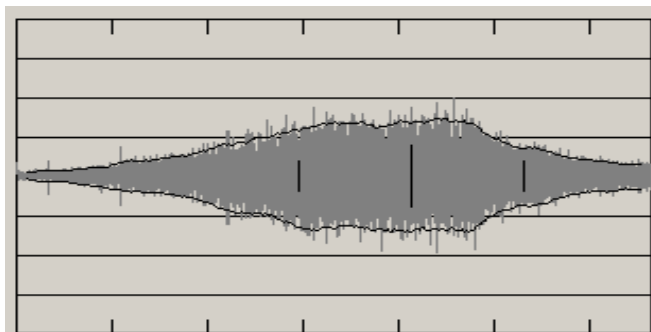
Pro rychlost vozidla 30 km/h se mění frekvence v rozmezí ± 2 sekundy od maxima, pro rychlost 90 km/h je to jen $\pm 0,5$ sekundy. Graf je vynesen pro vzdálenost pozorovatele od projíždějícího vozidla 3 metry.

Kombinací obou vlivů (rychlost vozidla a vzdálenost měřícího bodu od silnice) obdržíme průběhy s nejrůznější strmostí. Při výpočtech rychlosti není oblast kolem okamžiku průjezdu použita a předpokládám, že v úseku před a po průjezdu (mimo tuto oblast) jsou frekvence konstantní.

6.1 BLOKOVÉ SCHÉMA PROGRAMU



Akustický signál od automobilu je zaznamenán pomocí mikrofonového vstupu zvukové karty, s využitím API funkce `mcisendString` jsou nastaveny parametry záznamu na vzorkovací frekvenci 44100 Hz, mono, 8bit.

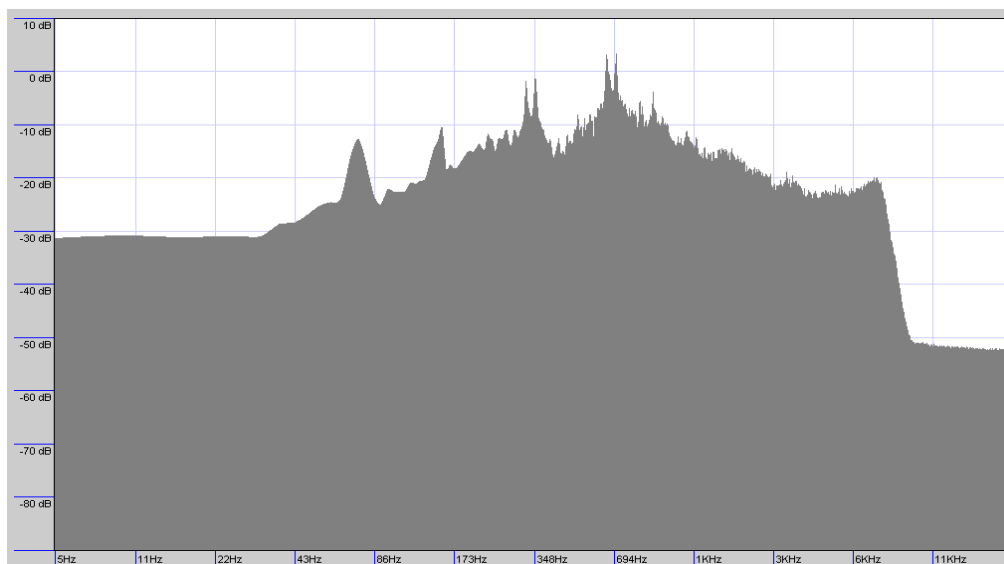


Obrázek 12 Vykreslení klouzavých průměrů

Pro nalezení maxima intenzity je záznam rozdělen na 300 úseků (velikost grafu pro zobrazení záznamu), v každém úseku je zjištěna minimální a maximální hodnota a pro tyto hodnoty jsou pro vyhlazení vypočítány klouzavé průměry s délkou 11. Na obrázku 12 je takový záznam s vykreslením klouzavých průměrů.

Z průběhu minim a maxim je určen časový okamžik minima minim a maxima maxim a jako průměr vypočítán okamžik průjezdu, na obrázku 12 vyznačen delší svislou čarou. Na obrázku jsou ještě dvě další svislé čáry vyznačující časovou vzdálenost od průjezdu $\pm$ sekunda.

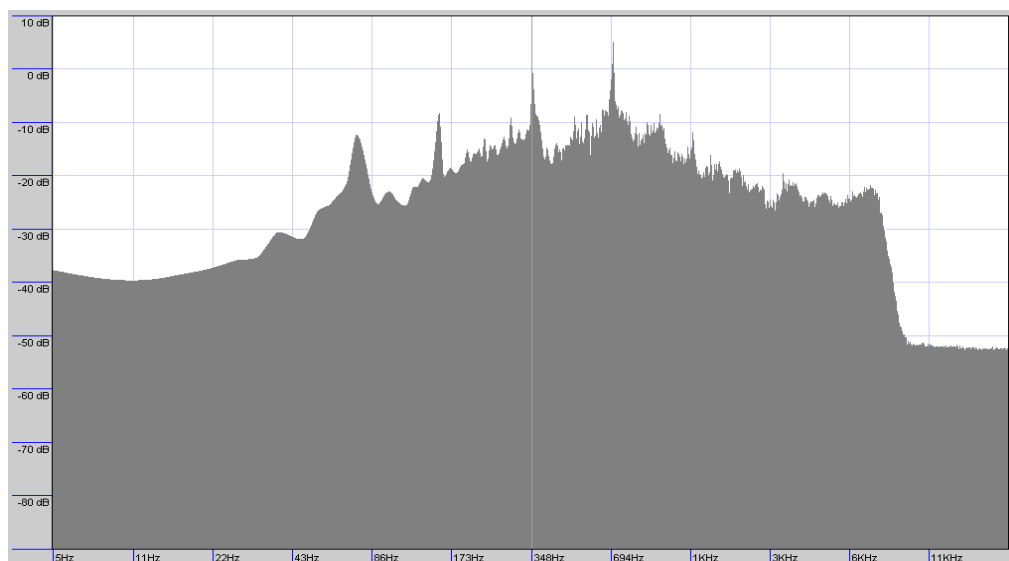
Poté je provedena frekvenční analýza záznamu. Na obrázku 13 je frekvenční spektrum celého záznamu z obrázku 12 zobrazené programem Audacity.



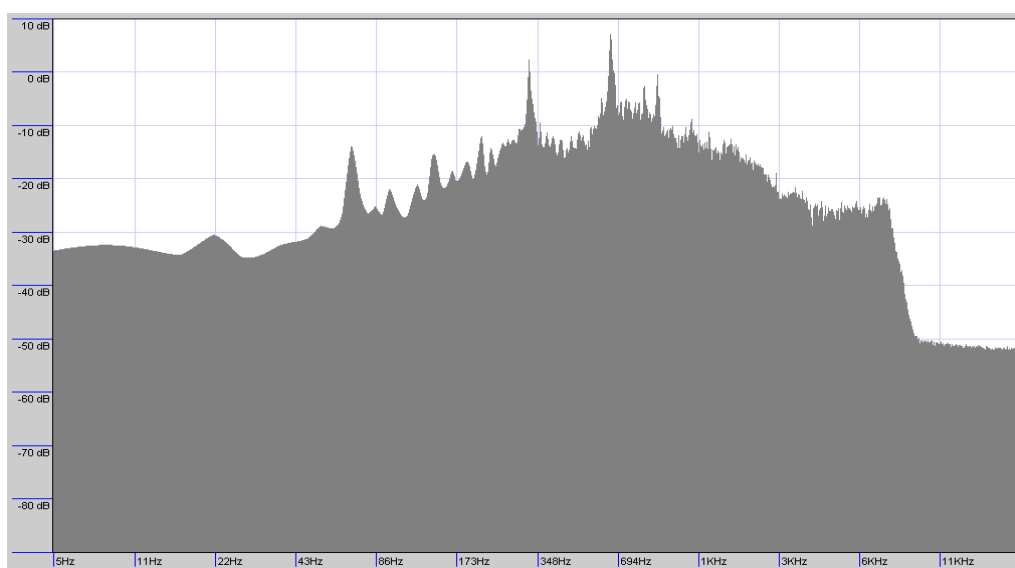
Obrázek 13 Frekvenční spektrum zvukového záznamu

Měřítka na ose x je logaritmické a v grafu jsou patrné čtyři výrazné piky, odpovídající frekvence jsou 323 Hz, 350 Hz, 649 Hz a 704 Hz. Z obrázku je patrné, že v záznamu dominují frekvence do 1000 Hz, vyšší jsou v záznamu potlačeny.

Provedeme-li analýzu části záznamu před okamžikem průjezdu, obdržíme spektrum na obrázku 14. Na obrázku 15 spektrum po průjezdu kolem měřicího místa.



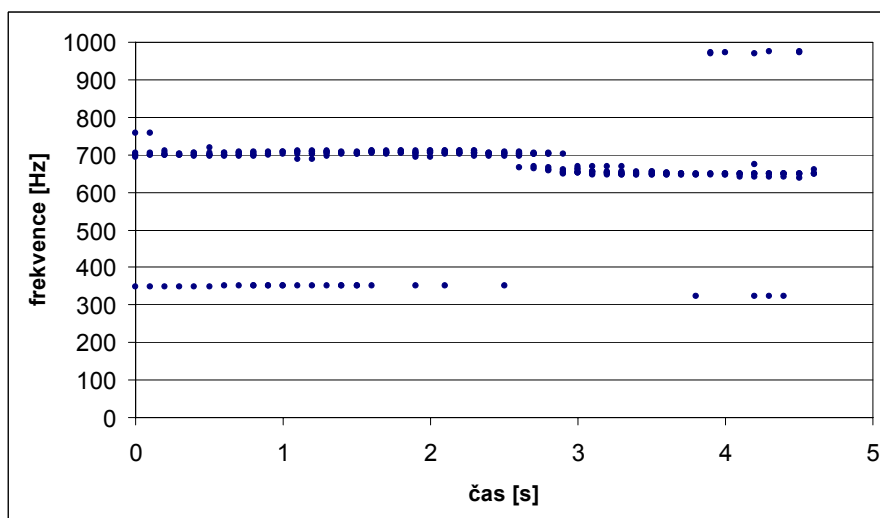
Obrázek 14 Frekvenční spektrum záznamu před průjezdem



Obrázek 15 Frekvenční spektrum záznamu po průjezdu

Oproti obrázku 13 zmizely dvojité vrcholy, na obrázku 14 jsou jen vrcholy pro frekvence 705 Hz a 350 Hz, na obrázku 15 vrcholy pro 649 Hz a 322 Hz.

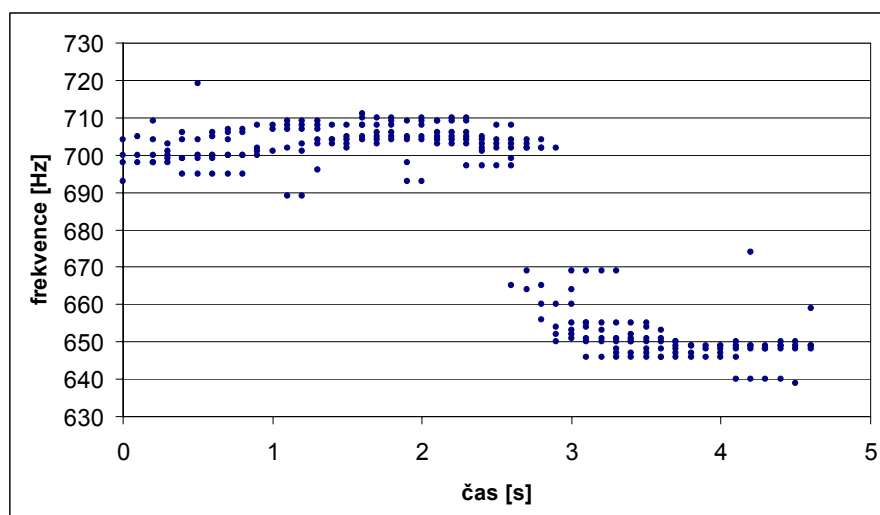
Program rozdělí záznam na sekundové úseky s posunem 0,1 sekundy a ty jsou postupně analyzovány, je vypočítáno frekvenční spektrum, zjištěny vrcholy a jim odpovídající frekvence. Na obrázku 16 jsou zjištěné frekvence vyneseny do grafu.



Obrázek 16 Významné frekvence v závislosti na čase

Z grafu je patrné, že se i při mých výpočtech vyskytují ve spektru před okamžikem průjezdu hodnoty dominantních frekvencí kolem 700 Hz a 350 Hz, za okamžikem průjezdu se hodnoty těchto frekvencí snížily.

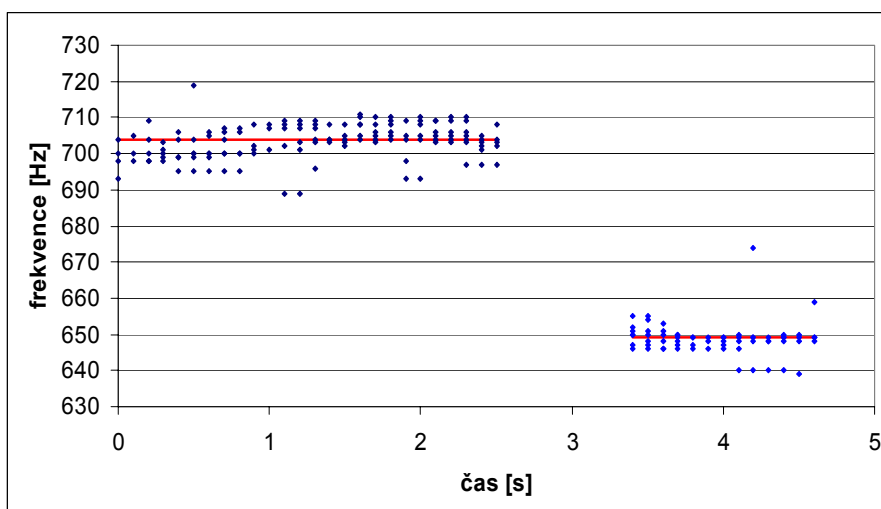
Na obrázku 17 je výřez z předcházejícího grafu jen pro frekvence od 630 do 730 Hz, z grafu je dobře patrný esovitý průběh předpokládaný na obrázcích 10 a 11.



Obrázek 17 Významné frekvence (výřez obr. 16)

Frekvence z okolí okamžiku průjezdu nejsou při výpočtu uvažovány, hodnotami před průjezdem je proložena přímka rovnoběžná s osou x, stejný postup je

uplatněn pro hodnoty frekvencí po okamžiku průjezdu. Výsledek je zobrazen na obrázku 18.



Obrázek 18 Frekvence před a po okamžiku průjezdu

Hodnota frekvence před okamžikem průjezdu je 704 Hz, hodnota po průjezdu je 649 Hz. Z těchto dvou frekvencí je možno s použitím vztahů (1) a (2) vypočítat rychlost okolojedoucího vozidla.

Rychlost je dána vztahem

$$v = c \frac{f_{pred} - f_{po}}{f_{pred} + f_{po}} \quad (5)$$

kde c je rychlost zvuku, f_{pred} frekvence zvuku před průjezdem okolo měřicího bodu, f_{po} frekvence po průjezdu.

Pro hodnoty z obrázku 18 s použitím vztahu (5) vypočítáme rychlost projíždějícího vozidla – 49,9 km/h.

Program opakuje postup také pro frekvence kolem 350 Hz a pokud by se vyskytovalo na obrázku 16 více linií, tak i pro ně (do maximálního počtu 10).

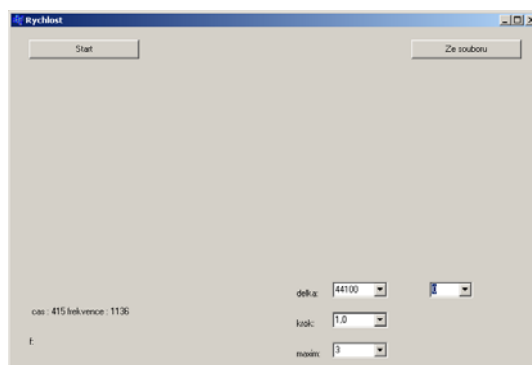
Z takto vypočtených rychlostí vyřadí rychlosti s chybou výpočtu větší než 10% a ze zbylých vypočítá průměrnou rychlost.

7. POPIS PROGRAMU

Program **Rychlost** je napsán v jazyce C pomocí Borland C++ Builderu v6 [3] pro operační systém Windows, testován byl na systému Windows XP. Existuje ve dvou verzích Pracovní a Měřicí. První slouží k vyladování algoritmu a má velké množství parametrů, které je možno měnit, druhá slouží jen k určení rychlosti.

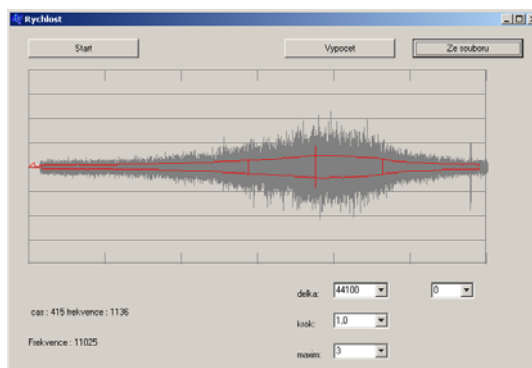
7.1 PRACOVNÍ VERZE

Aplikaci tvoří jedno okno, na počátku je možno zvolit, zda bude měřeno přímo záznamem zvuku – tlačítko Start, nebo je možno použít k analýze už zaznamenaný zvuk a nahrát jej pomocí tlačítka Ze souboru.



Obrázek 19 Základní okno aplikace

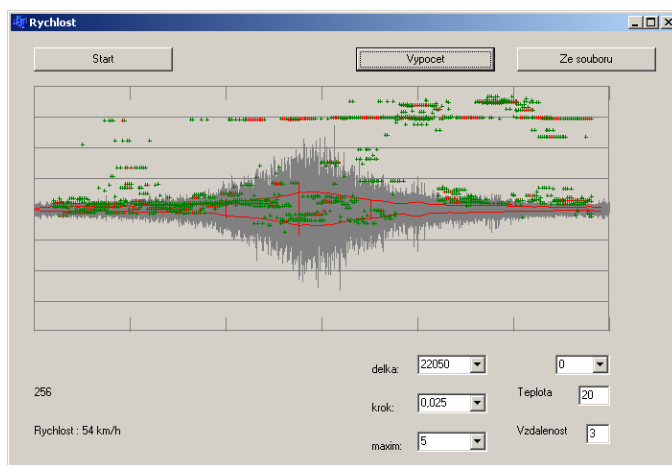
Po načtení záznamu je nalezeno maximum intenzity a vše vykresleno v okně.



Obrázek 20 Vykreslení záznamu s určením okamžiku průjezdu

Ve spodní části okna jsou umístěny prvky, kterými je možno volit parametry, které ovlivňují výpočet. Délka výpočetního okna, krok posunu okna a počet maxim, která se při výpočtu berou v úvahu. Tyto volby slouží k vyladění algoritmu pro výpočet rychlosti.

Na následujícím obrázku je vykreslen záznam i s křížky, které označují frekvence vypočítané pro příslušný úsek záznamu.



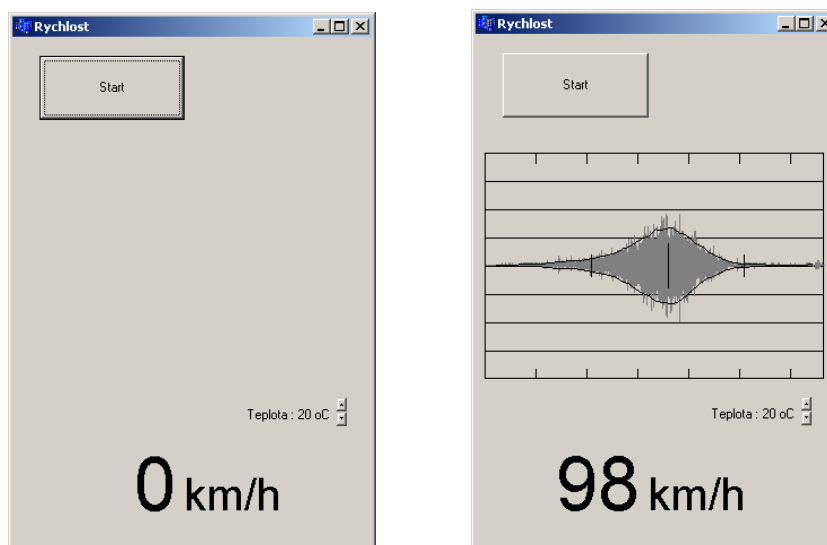
Obrázek 21 Záznam s vyznačením význačných frekvencí

Z obrázku je patrné, že opravdu dochází ke změně v hodnotě frekvencí v okolí okamžiku průjezdu, průběh je ve tvaru zhlazeného skoku, jak předpokládají grafy na obrázcích 10 a 11.

Pro výpočet rychlosti je body reprezentující frekvence před okamžikem průjezdu proložena přímka, body po okamžiku průjezdu je proložena druhá přímka a poté vypočítána rychlost pomocí vzorce (5).

7.2 MĚŘICÍ VERZE

Aplikaci tvoří jedno okno, jako parametr pro výpočet se nastavuje pouze teplota okolního vzduchu. Tlačítkem Start započne nahrávání záznamu, tlačítkem Stop (objeví se až po zahájení nahrávání) ukončíme nahrávání.



Obrázek 22 Okno aplikace před záznamem a po výpočtu

Program provede analýzu signálu stejně jako Pracovní verze, jen s pevně nastavenými parametry – krok 0,1 sekundy, délka vzorku 44100, počet frekvencí, které se vyhodnocují v každém kroku 5.

V obou verzích programu jsou použity následující funkce:

StartZaznam

Funkce otevře soubor pro záznam zvuku a nastaví parametry tohoto záznamu :
vzorkovací frekvence 44100, jeden kanál (mono), 8 bitů na vzorek.

StopZaznam

Funkce zastaví záznam a uloží ho do WAV souboru.

NactiWAV

Funkce načte zvukový záznam ze souboru.

NakresliPrazdnyGraf

Funkce vykreslí do okna prázdný graf, vyznačí osy.

VykresliKIPrumer

Funkce vykreslí záznam zvuku, pomocí klouzavých průměrů stanoví maximum v průběhu záznamu a vykreslí značku pro okamžik průjezdu.

Vypocet

Funkce provede s použitím funkcí knihovny FFTW výpočty Fourierovy transformace pro vzorek dané délky.

NajdiMaxima

Funkce nalezne daný počet frekvencí, které odpovídají maximům ve frekvenčním spektru.

AnalyzaMaxim

Funkce proloží hodnotami frekvencí před a po okamžiku průjezdu přímkou a z hodnot frekvencí vypočítá podle vztahu (5) rychlost vozidla.

7.3 ODHAD CHYBY VÝPOČTU

Výsledná rychlost je vypočítávána podle vztahu (5)

$$v = c \frac{f_{pred} - f_{po}}{f_{pred} + f_{po}} .$$

Rychlost c je určena podle vztahu (4) $c = (331,82 + 0,61 t) \text{ m.s}^{-1}$, teplotu t měříme s přesností na stupně Celsia, vnáší tedy do vztahu chybu přibližně 0,2%. Nezapočítáním vlivu vlhkosti další chybu asi 0,1%.

Frekvenci f_{pred} a f_{po} určíme s přesností 2 Hz. Pro frekvenci 100 Hz je tedy chyba 2%, pro 1000 Hz 0,2%.

Relativní chyba výsledku je dána součtem relativních chyb jednotlivých veličin, protože se ve vztahu (3) vyskytují jen v první mocnině.

Vyskytují-li se v signálu frekvence kolem 100 Hz, je výsledná rychlost určena s přesností do 5 %, pro frekvence okolo 1000 Hz je rychlost určena s relativní přesností do 1 %.

8. VÝSLEDKY MĚŘENÍ

Zvukové záznamy, které jsem použila ve své práci jsem nahrávala přímo samotným programem (program vytváří pracovní soubor Trec.wav). Další variantou byl kontinuální záznam na počítač pomocí nějakého zvukového programu nebo digitální záznamník. Záznam jsem zpracovávala v programu Audacity (šířen pod licencí GNU GPL). Vyřízla jsem ze záznamu jen část v okolí okamžiku průjezdu s dostatečným časem před a po průjezdu. Tím jsem mohla eliminovat záznamy např. s dvěma projíždějícími automobily. Samotné nahrávání s programem **Rychlost** chce trochu cviku, hlavně v odhadu počátku nahrávání podle rychlosti přibližování se automobilu, ale zejména vyčkat dostatečně do průjezdu, aby byl delší časový úsek pro výpočet.

Při odladování programu jsem používala možnost Pracovní verze, ve které je možné použít WAV soubor přímo z disku. Při simulování reálné činnosti jsem záznamy přehrávala v MediaPlayeru, zvukový výstup a vstup jsem měla propojeny kablíkem (JACK 3,5 – JACK 3,5) a programem **Rychlost** jsem zaznamenávala zvukový signál, stejně jako bych byla někde u silnice.

Ověřování jsem nejprve prováděla porovnáním hodnot naměřených stacionárním informačním radarem (jeden takový máme umístěn při vjezdu do obce). Nevýhodou tohoto ověřování je skutečnost, že tyto radary jsou umístěny při vjezdu do obcí a v měřeném úseku dochází u vozidel k výrazné změně rychlosti. Metoda, kterou ověřuji, naopak vyžaduje v celém měřeném úseku konstantní rychlost, jinak dochází k výraznému zkreslení výsledků a nemožnosti takovou rychlost vůbec určit. Většina záznamů získaných tímto způsobem je proto nepoužitelná. Další nevýhodou je pouze získání rychlostí kolem 50 km/h.

Další kontrolní záznamy jsem získala ve spolupráci s pracovníky firmy RAMET C.H.M. a.s.. Prováděli jsme měření pomocí mobilního radaru zabudovaného do osobního automobilu. Zároveň jsem zaznamenávala zvukový signál pomocí digitálního záznamníku a také notebooku. Výsledkem měření radarem je protokol, jehož ukázka je na obrázku 23. Z něj jsou patrné časové rozdíly mezi

měřeními, mezi jednotlivými průjezdy automobilů. Tyto časové rozdíly mně posloužily k synchronizaci mezi protokolem a zvukovými záznamy. Z nich jsem poté vyřízla záznamy jednotlivých průjezdů, nyní již se znalostí odpovídající rychlosti. Několik takových záznamů je přiloženo na CD.

Start, km/h, 04.05.2009, 08:52:55,	KUNOVICE,	OVERENI, 20, 20, 20, 20, da, 60
□dT, 79, km/h, 04.05.2009, 08:52:56, S		
□dC, 83, km/h, 04.05.2009, 08:53:01, S		
□aT, 77, km/h, 04.05.2009, 08:53:17, S		
□aC, 83, km/h, 04.05.2009, 08:53:18, S		
□dC, 83, km/h, 04.05.2009, 08:53:20, S		
□aT, 81, km/h, 04.05.2009, 08:53:34, S		
□aT, 83, km/h, 04.05.2009, 08:53:36, S		
□aC, 80, km/h, 04.05.2009, 08:53:38, S		
□aT, 87, km/h, 04.05.2009, 08:53:54, S		
□aT, 85, km/h, 04.05.2009, 08:53:57, S		
□aT, 85, km/h, 04.05.2009, 08:54:21, S		
□dC, 108, km/h, 04.05.2009, 08:54:23, S		
□dT, 88, km/h, 04.05.2009, 08:54:51, S		
□aT, 126, km/h, 04.05.2009, 08:54:56, S		
□aC, 80, km/h, 04.05.2009, 08:55:00, S		

Obrázek 23 Protokol z radaru AD 9 C

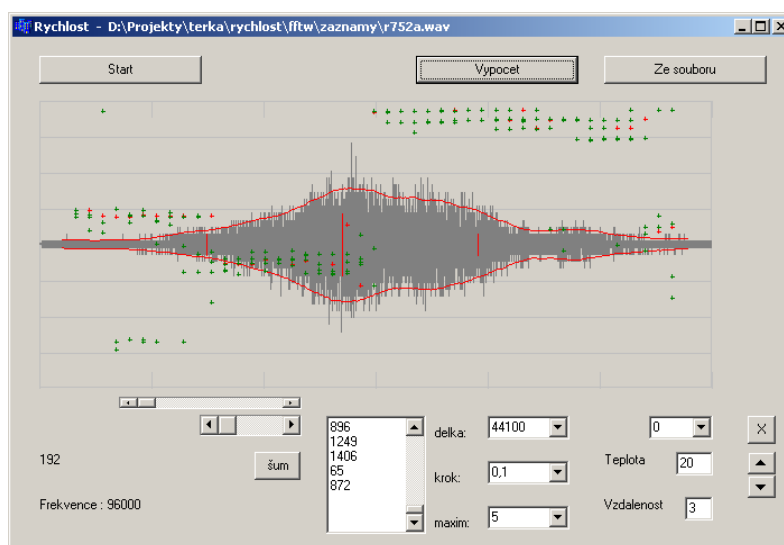
V následující tabulce jsou uvedeny pro porovnání rychlosti zjištěné radarem AD 9 C a programem Rychlost.

záznam	radar AD C 9	program Rychlost
R211.wav	88 km/h	100 km/h
R330.wav	85 km/h	87 km/h
R511.wav	109 km/h	117 km/h
R535.wav	78 km/h	102 km/h
R551.wav	100 km/h	95 km/h
R556.wav	116 km/h	110 km/h
R752.wav	80 km/h	92 km/h

Tabulka 1 Porovnání měření radarem a určení rychlosti programem Rychlost

Z tabulky je patrné, že ne vždy je rychlost určená programem odpovídající skutečné. V polovině případů je program v toleranci $\pm 5\%$. Hlavní příčinou je

nemožnost nalezení nějaké závislosti v určených frekvencích u některých záznamů. Na obrázku 21 jsou zobrazeny pomocí křížků frekvence a je možné zde vysledovat linie, z kterých je možno vypočítat frekvence potřebné k výpočtu rychlosti. Některé záznamy ale vypadají jako ten na obrázku 24, kdy před průjezdem a po průjezdu se vyskytují úplně odlišné frekvence a s takovými závislostmi si dosavadní algoritmus nedokáže poradit. V tom případě je zobrazena rychlost 0 km/h.



Obrázek 24 Záznam s vyznačením frekvencí II

9. ZÁVĚR

Cílem práce bylo ověřit, zda je možno měřit rychlost projíždějícího automobilu jen analýzou zvuku motoru.

Byl vytvořen program pro prostředí Windows, který je schopen analyzovat zvukový záznam a z vypočítaných frekvencí zvuku pohybujícího se automobilu stanovit rychlost automobilu.

Prozatímni úspěšnost v určení rychlosti není příliš velká, ale dalším rozvíjením, hlavně algoritmu při analýze frekvencí je možné program vylepšit.

Také vzhled aplikace, tj. pouze jednoduché okno se dvěma tlačítky vybízí k přenosu této aplikace na nějaký kapesní počítač (PDA), popřípadě na některý z „chytrých“ telefonů s operačním systémem. V tomto směru je možno práci ještě rozvíjet.

↑89km/h 0000013233 08:47:18 04.05.2009



Obrázek 25 Někteří řidiči dokonce dodržují stanovenou rychlost

10. LITERATURA

- [1] JAN Jiří : *Číslíková filtrace, analýza a restaurace signálů*, VUTiUM Brno 2002, ISBN 80-214-1558-4
- [2] DOLEŽAL Aleš: *Programové vybavení pro frekvenční analýzu zvukového signálu*, bakalářská práce, Univerzita Tomáše Bati, Zlín, 2006
- [3] MATOUŠEK David: *C++ Builder, vývojové prostředí*, BEN, 2003
- [4] BESIP, *Měření rychlosti vozidel* [online], [cit. 2009-2-25],
URL <<http://www.ibesip.cz/Rychlost/Mereni-rychlosti-vozidel>>
- [5] Antiradary Beltronic, *Lasery Pro Laser III*, [online], [cit. 2009-5-1], URL
<http://www.antiradar.com/radary/main.php3?text=novezbra.htm&sel_id=6>
- [6] FRIGO Mateo, JOHNSON Steven G.: *FFTW manual*, MIT 2008, [online],
[cit. 2009-5-1], URL <<http://www.fftw.org/fftw3.pdf>>
- [7] RAMET - *Policejní radary řady AD9*, [online] [cit. 2009-4-20]
URL <<http://www.rametchm.cz/index.php?typ=RMA&showid=13>>
- [8] HORÁK Zdeněk, KRUPKA František: *Fyzika, Příručka pro vysoké školy technického směru*, SNTL/ALFA, Praha, 1976
- [9] antiradarynet *Úsekové měření - další forma skrytého zdanění*, [online], [cit. 2007-1-8], URL <<http://www.antiradary.net/usekove-mereni>>

11. SEZNAM ZKRATEK

DFT	discrete Fourier transform
EHK	Evropská hospodářská komise
EHS	Evropské hospodářské společenství
FFTW	Fastest Fourier Transform in the West
GNU	GNU's Not UNIX
GPL	General Public License
LIDAR	Light Detection And Ranging
OSN	Organizace spojených národů
PDA	personal digital assistant
RAMER	radarový měřič rychlosti
SIMD	Single Instruction Multiple Data
WAV	Waveform audio format

12. SEZNAM OBRÁZKŮ

Obrázek 1 Dokumentační snímek z radaru AD 9 C.....	10
Obrázek 2 Ruční laserový měřič rychlosti ProLaser II od firmy Kustom Signals, Inc. – převzato z [5].....	10
Obrázek 3 Princip činnosti laserového měřiče rychlosti – převzato z [5].....	11
Obrázek 4 Princip úsekového měření – převzato z [9]	12
Obrázek 5 Šíření vln z klidného a pohybujícího se zdroje – převzato z [8]	14
Obrázek 6 Vzorkování spojitého signálu - převzato z [2].....	16
Obrázek 7 Kvantování signálu - převzato z [2]	16
Obrázek 8 Fourierova transformace.....	20
Obrázek 9 Zvukový záznam projíždějícího automobilu	21
Obrázek 10 Průběh změny frekvence pro různé vzdálenosti měřicího bodu.....	21
Obrázek 11 Průběh změny frekvence pro různé rychlosti	22
Obrázek 12 Vykreslení klouzavých průměrů.....	23
Obrázek 13 Frekvenční spektrum zvukového záznamu.....	24
Obrázek 14 Frekvenční spektrum záznamu před průjezdem	25
Obrázek 15 Frekvenční spektrum záznamu po průjezdu	25
Obrázek 16 Významné frekvence v závislosti na čase	26
Obrázek 17 Významné frekvence (výřez obr. 16)	26
Obrázek 18 Frekvence před a po okamžiku průjezdu.....	27
Obrázek 19 Základní okno aplikace.....	28
Obrázek 20 Vykreslení záznamu s určením okamžiku průjezdu	28
Obrázek 21 Záznam s vyznačením význačných frekvencí	29
Obrázek 22 Okno aplikace před záznamem a po výpočtu	30
Obrázek 23 Protokol z radaru AD 9 C	33
Obrázek 24 Záznam s vyznačením frekvencí II.....	34
Obrázek 25 Někteří řidiči dokonce dodržují stanovenou rychlost.....	35

13. PŘÍLOHA

13.1 VÝPIS PROGRAMU RYCHLOST – MĚŘICÍ VERZE

Rychlost.cpp

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
//-----  
USEFORM("UnitRychlost1.cpp", Form1);  
//-----  
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)  
{  
    try  
    {  
        Application->Initialize();  
        Application->CreateForm(__classid(TForm1), &Form1);  
        Application->Run();  
    }  
    catch (Exception &exception)  
    {  
        Application->ShowException(&exception);  
    }  
    catch (...)  
    {  
        try  
        {  
            throw Exception("");  
        }  
        catch (Exception &exception)  
        {  
            Application->ShowException(&exception);  
        }  
    }  
    return 0;  
}  
//-----
```

UnitRychlost1.cpp

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
  
#include "UnitRychlost1.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
  
#pragma comment(lib, "fftw3.lib")  
  
#include <mmsystem.h>  
#include <stdio.h>  
#include "fftw3.h"  
  
TForm1 *Form1;  
  
#include "proc.cpp"
```

```
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    StartZaznam(); // spusti zaznam do wavu
    NakresliPrazdnyGraf();
    Button2->Visible = true;
    Button1->Visible = false;
}
//-----

void __fastcall TForm1::Button2Click(TObject *Sender)
{
    // stopne zaznam a ulozi ho do trec.wav a do promenne pom
    StopZaznam();
    NactiWAV(JmenoSouboru);
    VykresliZaznam();
    VypoctiRychlost();

    Button1->Visible = true;
    Button2->Visible = false;
}
//-----

void __fastcall TForm1::UpDown1Click(TObject *Sender, TUDBtnType Button)
{
    Teplota = UpDown1->Position;
    Label3->Caption = "Teplota : " + IntToStr( Teplota ) + " °C" ;
}
//-----

proc.cpp

//-----
long frekv = 44100; // vzorkovací frekvence
double *pom; // odkaz na pole hodnot
double *maxima; // odkaz na pole maxim
long NN = 1; // delka pole zaznamu
long stred = 1; // stred zaznamu (maximum)
long DelkaN = 44100;
int Nmax = 5; // pocet maxim ktera se hledaji
float krok = 0.1; // casovy krok 0.1 sekundy
int krokI;
char *JmenoSouboru = "Trec.wav";
int Teplota = 20; // pocatecni teplota
int NumMax = 30;
const MaxHz = 2000;
//-----

void NakresliPrazdnyGraf()
{
    Form1->PaintBox1->Canvas->Pen->Color=clBlack;
    Form1->PaintBox1->Canvas->Rectangle(0,0,300,200);
    for (int i=0; i<8; i++) {
        Form1->PaintBox1->Canvas->MoveTo(0,25*i);
        Form1->PaintBox1->Canvas->LineTo(300,25*i);

        Form1->PaintBox1->Canvas->MoveTo(45*i,0);
        Form1->PaintBox1->Canvas->LineTo(45*i,10);
        Form1->PaintBox1->Canvas->MoveTo(45*i,200);
    }
}
```

```

        Form1->PaintBox1->Canvas->LineTo(45*i,190);
    };
    Form1->PaintBox1->Canvas->MoveTo(0,100);
}
//-----

void StartZaznam()
{
    mciSendString("OPEN NEW TYPE WAVEAUDIO ALIAS mySound",NULL,0,NULL);
    mciSendString("SET mySound TIME FORMAT MS BITSPERSAMPLE 8 CHANNELS 1 SAMPLESPERSEC
        44100 BYTESPERSEC 44100 ",
        NULL,0,NULL);
    mciSendString("RECORD mySound",NULL,0,NULL);
}
//-----

void StopZaznam()
{
    mciSendString("STOP mySound",NULL,0,NULL);
    mciSendString("SAVE mySound ""Trec.wav",NULL,0,NULL);
    mciSendString("CLOSE mySound",NULL,0,NULL);
}
//-----

void NactiWAV(AnsiString Jmeno)
{
    FILE *souborVST;
    long i, ii;
    short bb;

    char *jm;
    jm = (char*) malloc(Jmeno.Length()+1);
    jm = Jmeno.c_str();

    souborVST = fopen(jm,"rb");
    fseek(souborVST, 0L, SEEK_END);
    NN = ftell(souborVST)- 44;

    pom = (double*) fftw_malloc(sizeof(double) * NN);

    fseek(souborVST, 24L, SEEK_SET); // nastaveni na pozici frekvence
    ii = 1; frekv = 0;
    for (i=1;i<=4;i++){
        bb = 0;
        fread( &bb, 1, 1, souborVST );
        frekv = frekv + ii * bb;
        ii = ii * 256;
    };

    fseek(souborVST, 44L, SEEK_SET); // nastaveni za hlavicku
    i=0;
    while (!feof(souborVST)) {
        fread( &bb, 1, 1, souborVST );
        pom[i]= bb-128;
        i++;
    }
    fclose(souborVST);

    krokI = (int) frekv * krok;
}
//-----

void VykresliKlPrumer()
{
    const Delka = 301;
    long plus [Delka], minus [Delka];

```

```

long plusI [Delka], minusI [Delka];
long i, ii, Max0=0;

for (i=0; i<Delka; i++) {
    plus[i]=0; minus[i]=0; plusI[i]=1; minusI[i]=1;
};

for (i=1; i<NN-1; i++) {
    ii = i / ( NN / Delka );
    if (pom[i]>plus[ii]) plus[ii]=pom[i];
    if (pom[i]<minus[ii]) minus[ii]=pom[i];
    if (abs(pom[i])>Max0) Max0 = abs(pom[i]);
};

Form1->PaintBox1->Canvas->Pen->Color = clDkGray;
for (i=0; i<Delka; i++) {
    minus[i] = 50*minus[i]/Max0;
    plus[i] = 50*plus[i]/Max0;
    Form1->PaintBox1->Canvas->MoveTo(i,100+minus[i]);
    Form1->PaintBox1->Canvas->LineTo(i,100+plus[i]);
};

int kp = 5;
int kpP, kpM;
int kpMin=1000, kpMax=0, kpMinI=0, kpMaxI=0;

for (i=kp; i<Delka-kp; i++) {
    kpP=0; kpM=0;
    for (ii=i-kp; ii<=i+kp; ii++) {
        kpP = kpP + plus[ii];
        kpM = kpM + minus[ii];
    };
    plusI[i] = kpP / (2*kp+1);
    minusI[i] = kpM / (2*kp+1);
    if (plusI[i]>kpMax) {kpMax=plusI[i]; kpMaxI=i;};
    if (minusI[i]<kpMin) {kpMin=minusI[i]; kpMinI=i;};
};
// stred je nekde mezi maximem hodnim a maximem dolnim
stred = (NN/Delka) * (kpMaxI + kpMinI)/2;

Form1->PaintBox1->Canvas->Pen->Color=clBlack;
// nakresli caru u stredu
Form1->PaintBox1->Canvas->MoveTo((kpMaxI + kpMinI)/2,80);
Form1->PaintBox1->Canvas->LineTo((kpMaxI + kpMinI)/2,120);
// nakresli caru -1 sekunda
Form1->PaintBox1->Canvas->MoveTo((stred-frekv)/( NN / Delka ),90);
Form1->PaintBox1->Canvas->LineTo((stred-frekv)/( NN / Delka ),110);
// nakresli caru +1 sekunda
Form1->PaintBox1->Canvas->MoveTo((stred+frekv)/( NN / Delka ),90);
Form1->PaintBox1->Canvas->LineTo((stred+frekv)/( NN / Delka ),110);

// nakresli horni caru
Form1->PaintBox1->Canvas->MoveTo(kp,100+plusI[kp]);
for (i=kp; i<Delka-kp; i++) { Form1->PaintBox1->Canvas->LineTo(i, 100+plusI[i]);
};
// nakresli dolni caru
Form1->PaintBox1->Canvas->MoveTo(kp,100+minusI[kp]);
for (i=kp; i<Delka-kp; i++) { Form1->PaintBox1->Canvas->LineTo(i, 100+minusI[i]);
};

}
//-----

void VykresliZaznam()
{
    NakresliPrizdnyGraf();
    VykresliKlPrumer();
}
//-----

```

```
void NajdiMaxima(long j, long N, double *out)
{
    long i, ii, iii, Ffr;
    const NmaxM = 10;
    double max[NmaxM];
    long maxi[NmaxM];

    // vynulovani maxim
    for (i=0; i<Nmax; i++) { max[i]=0.0; maxi[i]=0; };

    // nalezeni maxim
    for (i=1; i<=N-2; i++)
        if (max[Nmax-1]<abs(out[i])){
            for (ii=0; ii<Nmax-1; ii++) {
                if (abs(out[i])>max[ii]) {
                    for (iii=Nmax-1; iii>ii-1; iii--) {
                        max[iii]=max[iii-1];
                        maxi[iii]=maxi[iii-1];
                    };
                    max[ii] = abs(out[i]);
                    maxi[ii] = i;
                    ii = Nmax;
                };
            };

            for (i=0; i<Nmax; i++) { // ulozeni maxim
                ii = maxi[i];
                if (((out[ii]-out[ii-1])*(out[ii+1]-out[ii]))<0) {
                    if (ii<N/2) Ffr=frekv*ii/N; else Ffr = frekv*(N-ii)/N;
                    maxima[j*NumMax+i] = Ffr;
                };
            };
        }
    //-----

float AnalyzaMaxim()
{
    int ii;
    int pocet [MaxHz];
    int mezera = 5;
    float pr = 0.0;
    float vZvuk = 331.82 + 0.61 * Teplota;
    float Spr = 0;

    for (ii=0; ii<MaxHz; ii++) pocet[ii]=0;

    int stredI = stred / krokI;

    for (int j=0; j<stredI-mezera; j++) { // stanoveni poctu jednotlivych frekvenci
        for (int i=0; i<Nmax; i++) {
            ii = maxima[j*NumMax+i];
            if ((ii>0)&&(ii<MaxHz)) {pocet[ii]++; };
        }
    }

    int PocetR = 0;
    float rr [10][10];

    for (int kk=0; kk<10; kk++){ // pro deset nejpocetnejsich frekvenci

        int predMax = 0, predMaxI=0;
        for (int i=0; i<MaxHz; i++) {
            if (pocet[i]>predMax) {
                predMax = pocet[i]; predMaxI = i;
            };
        };
        pocet[predMaxI]=0;
    }
}
```

```

int ctverec = 100;
int soucet=0;
int pocetI=0;
int maxSoucet = 100000;
int maxSoucetI = 0;
int pp = 0;
int pp2 = 0;
int j=0;

for (int i=predMaxI-5; i<predMaxI+5; i++) {
    soucet=0; pocetI=0;
    for (ii=0; ii<stredI-mezera; ii++) {
        int mm=ctverec;
        for (int il=0; il<Nmax; il++) {
            if ( (i-maxima[ii*NumMax+il])*(i-maxima[ii*NumMax+il]) < mm ) mm = (i-
            maxima[ii*NumMax+il])*(i-maxima[ii*NumMax+il]);
        };
        if (mm<ctverec) {
            soucet=soucet+ mm ;
            pocetI++;
        }
    };
    if (pocetI>0) { pp = pocetI;
        pocetI = 1000 * soucet/pocetI;
        if (pocetI<maxSoucet) {maxSoucet = pocetI; maxSoucetI = i; pp2 = pp;};
    };
    j++;
};

rr[kk][1] = maxSoucetI;
rr[kk][2] = pp2;

float pom = 150 / (3.6 * vZvuk);
int minR = (int) maxSoucetI * (1 - pom)/(1+pom);
if (minR < 1) minR = 1;
int maxPoI = maxSoucetI;
j=0;
maxSoucet = 100000;

for (int i=minR; i<maxPoI; i++) {
    soucet=0; pocetI=0;
//    for (ii=stredI+mezera; ii<NN/krokI ; ii++) {
    for (ii=stredI+1; ii<NN/krokI ; ii++) {
        int mm=ctverec;
        for (int il=0; il<Nmax; il++) {
            if ( (i-maxima[ii*NumMax+il])*(i-maxima[ii*NumMax+il]) < mm ) mm = (i-
            maxima[ii*NumMax+il])*(i-maxima[ii*NumMax+il]);
        };
        if (mm<=ctverec) {
            soucet=soucet+ mm ;
            pocetI++;
        }
    };
    if (pocetI>0) {
        pocetI = 1000 * soucet/pocetI;
        if (pocetI<maxSoucet) {maxSoucet = pocetI; maxSoucetI = i;};
    }
    j++;
};
rr[kk][3] = maxSoucetI;

float d1,d2,dd;

if ((maxPoI+maxSoucetI)>0) {
    pr = 3.6 * vZvuk * ( maxPoI - maxSoucetI ) / ( maxPoI + maxSoucetI );
    d1 = 3.6*vZvuk*(maxPoI+2-maxSoucetI) / (maxPoI + maxSoucetI);
    d2 = 3.6*vZvuk*(maxPoI-2-maxSoucetI) / (maxPoI + maxSoucetI);
    if (pr>0) {dd = 100.0 * ((d1-d2)/2)/pr;} else dd=100;
}
else { pr = 0; d1 = 0; d2 = 0; dd =100;};

```

```

        rr[kk][4] = pr;
        rr[kk][5] = dd;

    };

    for (int a1=0; a1<10; a1++) {
        if (rr[a1][5]>10.0) rr[a1][4]=0;    // vynuluj rychlost s chybou > 10%
    };

    Spr=0; PocetR=0;    // vypocet prumerne rychlosti
    for (int a1=0; a1<10; a1++) {
        if (rr[a1][4]>0.0) { Spr=Spr+rr[a1][2]*rr[a1][4]; PocetR=PocetR+rr[a1][2]; };
    };

    if (PocetR==0) return 0;

    return Spr/PocetR;

}

//-----

float Vypocet() {

    double *in, *out;    // promenne pro FFTW
    fftw_plan p;

    in = (double*) fftw_malloc(sizeof(double) * DelkaN);
    out = (double*) fftw_malloc(sizeof(double) * DelkaN);
    int pocetMaxim = NumMax * NN / krokI;
    maxima = (double*) malloc(sizeof(double)* pocetMaxim);
    for ( int i=0; i<pocetMaxim; i++) maxima[i]=0;    // vynulovani maxim
    long j = 0, ii =0;
    while (i < (NN - DelkaN) ) {
        for (int i=0; i<DelkaN; i++) { in[i]=pom[i+ii]; };
        p = fftw_plan_r2r_1d(DelkaN, in, out, FFTW_DHT, FFTW_ESTIMATE);
        fftw_execute(p);

        NajdiMaxima(j,DelkaN,out);

        fftw_destroy_plan(p);
        ii = ii + krokI;
        j++;
    };

    fftw_free(in);
    fftw_free(out);

    return AnalyzaMaxim();
}

//-----

void VypoctiRychlost() {

    Form1->Labell->Caption = IntToStr( (int) Vypocet() );

}

//-----

```