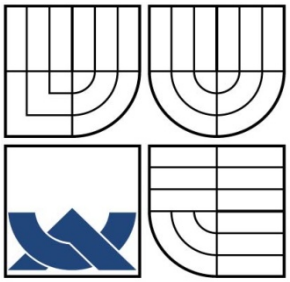
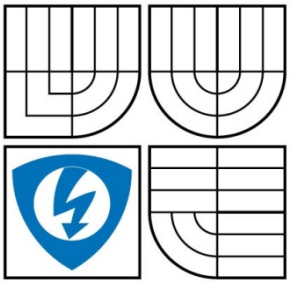


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

VYUŽITÍ MASKOVACÍCH EFEKTŮ PRO VODOZNAČENÍ AUDIO DAT

Using masking effects for audio data watermarking

DIPLOMOVÁ PRÁCE

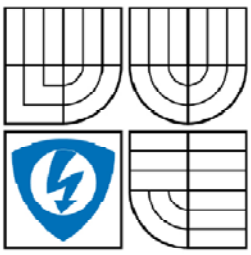
MASTER´S THESIS

AUTOR PRÁCE Bc. jiří kabourek

AUTHOR

VEDOUCÍ PRÁCE ING. RADEK ZEZULA.PHD
SUPERVISOR

BRNO 2008



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student:	Kabourek Jiří Bc.	***ID:***	54049
Ročník:	2	***Akademický rok:***	2007/2008

NÁZEV TÉMATU:

Využití maskovacích efektů pro vodoznačení audio dat

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte, popište a implementujte v prostředí Matlab metodu digitálního vodoznačení audio signálů využívající minimální maskovací práh psychoakstického modelu ucha pro vkládání vodoznaku do audio signálů. Implementovanou metodu vodoznačení otestujte na její robustnost a transparentnost vloženého vodoznaku.

DOPORUČENÁ LITERATURA:

[1] ARNOLD, M; SCHMUCKER, M.; WOLTHUSEN, S. D. Techniques and Applications of Digital Watermarking and Content Protection. Artech House, inc., 2003. 296 p. ISBN 1-58053-111-3

[2] FASTL, H.; ZWICKER, E. Psychoacoustics: Facts and Models. 3rd edition: Springer, 2006. 462 p. ISBN 3540231595

Termín zadání: 11.2.2008 ***Termín odevzdání:*** 28.5.2008

Vedoucí práce: Ing. Radek Zezula, Ph.D.

prof. Ing. Kamil Vrba, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Licenční smlouva

poskytovaná k výkonu práva užít školní dílo

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Bc. Jiří Kabourek

Bytem: Trstěnice 33

Narozen/a (datum a místo): 28.12.1983/Znojmo

(dále jen „autor“)

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií

se sídlem Údolní 244/53, 602 00, Brno

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

prof. Ing. Kamil Vrba, CSc.....

(dále jen „nabyvatel“)

Článek. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
   - ☐ disertační práce
   - ☐ diplomová práce
   - ☐ bakalářská práce
   - ☐ jiná práce, jejíž druh je specifikován jako .....

   (dále jen VŠKP nebo dílo)

Název VŠKP:	Využití maskovacích efektů pro vodoznačení audio dat
Vedoucí/ školitel VŠKP:	Ing. Radek Zezula, Ph.D.
Ústav:	Ústav telekomunikací
Datum obhajoby VŠKP:	

VŠKP odevzdal autor nabyvateli v*:

- ☐ tištěné formě – počet exemplářů ……1……..
- ☐ elektronické formě – počet exemplářů ……1……..

* hodící se zaškrtněte

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizovaní výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
   - ihned po uzavření této smlouvy
   - 1 rok po uzavření této smlouvy
   - 3 roky po uzavření této smlouvy
   - 5 let po uzavření této smlouvy
   - 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: …….

…….. ……

Nabyvatel Autor

ABSTRAKT

V této práci je prezentována technika pro vkládání digitálních vodoznaků v digitálních audio signálech. Digitální vodoznak musí být nepostřehnutelný a měl by být robustní proti útokům a různým rušením. Algoritmus pro vkládání vodoznaku využívá techniky rozprostřeného spektra a psychoakustického modelu ISO-MPEG I layer I. Robustnost vodoznaku byla testována na filtraci signálu, MP3 kompresi a na změnu vzorkovací frekvence.

KLÍČOVÁ SLOVA

Digitální vodoznačení audio signálu, psychoakustický model ISO-MPEG I layer I, rozprostřené spektrum

ABSTRACT

In this work is presented technique for embedding digital watermark in digital audio signals. Digital watermark must be imperceptible and should be robust against attacks and other types of distortion. Algorithm is implemented for embedding digital watermark using technique spread-spectrum and psychoacoustic model ISO-MPEG I layer I. Robustness was tested for filtering signal, MP3 compression and resample method.

KEYWORDS

Digital audio watermarking, psychoacoustic model ISO-MPEG I layer I, spread-spectrum

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma **Využití maskovacích efektů pro vodoznačení audio dat** jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne ....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce Ing. Radku Zezulovi.Phd, za jeho užitečné rady a pomoc při vypracování diplomové práce.

V Brně dne .....

OBSAH

Seznam obrázků

1 Úvod

Všestranné a jednoduché použití software a klesající ceny digitálních přístrojů (CD,MP3 přehrávače,PC, videokamery,DVD,laptopy,PDA….) mají lidé díky těmto přístrojům nepřeberné množství možností kopírování dat. Vysokorychlostní připojení k internetu usnadňuje lidem to, aby distribuovali velké množství multimediálních souborů a z nich tak vytvářeli identické digitální kopie. Výhodou těchto digitálních souborů je v tom, že kvalita při přehrávání nebo kopírování pořád stejná, což neplatí u analogové techniky (VHS pásky). Tyto výhody digitálních prostředků vedou k neomezenému kopírování bez ztráty věrnosti, a tím vznikají významné finanční ztráty pro autora. Snadnost obsahové změny a dokonalé rozmnožování v digitální oblasti vede k podpoře ochrany vlastnických práv jedince a prevence neautorizovaného falšování multimediálních dat. Tradiční metody pro ochranu autorských práv mediálních dat začínají být nevyhovující. Hackerství digitálních mediálních systémů se stává dokonalejší kvůli velké dostupnosti multimediálních procesů, díky internetu a výpočetní technice. Jednoduché ochranné mechanizmy, které byly založené na vložené informaci do hlavičky digitálního souboru jsou nepoužitelné, protože informace hlavičky může být snadno odstraněna jednoduchou změnou formátu dat, která nepůsobí na věrnost média.To směruje ke zvýšení nezákonného kopírování a distribuce digitálních dat bez ohledu na autorská práva. Digitální vodoznačení dat se snaží tyto problémy odstranit. Cílem digitálního vodoznačení je vývoj algoritmů, které umí vložit zprávu do digitálního souboru. Tato zpráva by měla být nepostřehnutelná pro uživatele a zároveň odolná proti pokusům na její odstranění. Ve zvukových signálech se snažíme, aby digitální vodoznak byl odolný proti zkreslení vzniklé při kompresi zvukových signálů nebo přenosu přes analogové prostředí. Musí být také odolná proti odstranění vodoznakové zprávy užitím psychoakustického modelu. Digitální vodoznačení akustických signálů je systém založen na psychoakustických principech lidského ucha. Protože lidské ucho funguje jako analyzátor spektra, tzn., že rozloží akustickou tlakovou vlnu na spektrum. Proto využíváme psychoakustického modelu, kde jeho výstupem je spektrum vnímatelné pro lidský sluch, který je nazýván minimální maskovací práh.

Využívá efektu maskování akustických signálů a detekce prahu slyšení. Psychoakustických modelů se využívá zejména pro kompresi zvukových signálů, ale zároveň se může využívat i pro digitální vodoznačení audio dat**.** Navržený algoritmus by měl generovat digitální vodoznak, který je upraven podle minimálního maskovacího prahu a vložen do zvukového signálu. Takto vložený upravený digitální vodoznak by měl být neslyšitelný a robustní. Pro vyšší robustnost digitálních vodoznaků využíváme algoritmů pro zpracování akustických signálu jako je rychlá Fourierova transformace (FFT), Vlnková transformace (Waveletova),diskrétní kosinova transformace(DCT), časově frekvenční analýza, kepstrální analýza, lineárně predikční algoritmy, různých metod pro přenos signálů jako je metoda rozprostřeného spektra a mnoho dalších metod.

2 Digitální vodoznačení

Digitální vodoznačení a stenografie jsou metody pro vkládání nepostřehnutelné informace k nosnému signálu. Stenografie je metoda popisující ukrývání zpráv, to jeví, jako kdyby komunikace neprobíhala, ale ve skutečnosti probíhá tajná komunikace. Toho se může využít ve spojeních bod-bod (point-to-point). U metod digitálního vodoznačení může mít protivník znalosti o tom, že komunikace je tvořena metodou digitálního vodoznačení. První pokusy digitálního vodoznačení se objevily v 90. letech minulého století, kdy digitální vodoznačení začalo budit pozornost významného množství vědců. Začalo to od jednoduchých základních principů až po sofistikované algoritmy používající znalosti z teorie komunikace.

2.1 Základní princip vodoznačení

Základním úkolem digitálního vodoznačení je ochrana autorských práv. Může nastat problém, kdy nejsme schopni určit vlastníka zvukové nahrávky. V případě takového konfliktu je vytažen skrytý podpis ze zvukové nahrávky, kterým může být například osobní identifikační číslo autora (ID).

Základní princip aktuálních systémů vodoznačení je srovnatelný se symetrickým šifrováním, kde používáme stejný klíč pro zakódování a dekódování vodoznaku. Každý systém vodoznačení se sestává ze dvou subsystémů: kodéru vodoznaku a dekodéru vodoznaku. Formální systém vodoznaku může být popsán $(\mathcal{O}, W, K, E_k, D_k, C\tau)$, kde $\mathcal{O}$ je soubor originálních dat, W se soubor vodoznačených dat, a K je soubor klíčů. Dvě funkce

$$E_k : \mathcal{O} \times W \times K \longrightarrow \mathcal{O} \tag{2.1}$$

$$D_k : \mathcal{O} \times K \longrightarrow \mathcal{O} \tag{2.2}$$

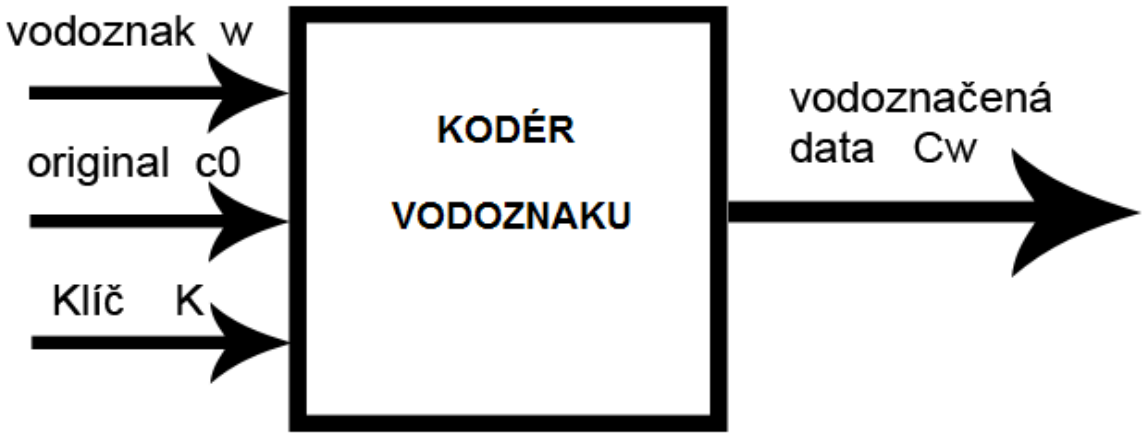
popisující vkládání a detekci vodoznaku. Komparační funkce je dána vztahem:

$$C\tau : W^2 \longrightarrow \{0,1\} \tag{2.3}$$

porovnávající extrahovaný se skutečným vodoznakem užitím citlivosti τ pro porovnání. Vstupní parametry vkládacího procesu jsou nosný signál (originální c_0), vodoznak w určený pro vložení a tajný a veřejný klíč K pro zakódování vodoznaku:

$$E_k(c_0, w) = c_w \tag{2.4}$$

Výstupem kodéru je soubor dat upravený s vloženou vodoznakovou zprávou viz.obr.2.1:



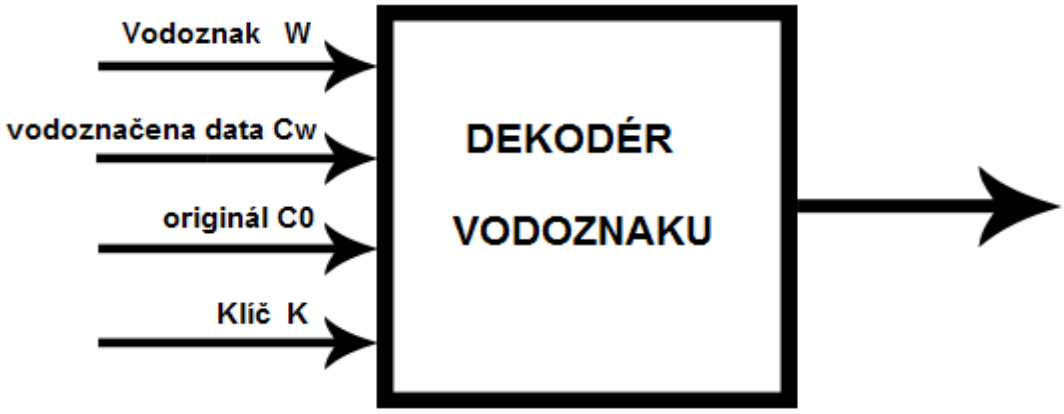
Obr.2.1 Blokové schéma kodéru

V detekčním procesu pro extrakci vodoznakové zprávy je užito vodoznačených $\hat{c}_w$, které mohli být během přenosu zkreslena nebo úmyslně poškozena, vodoznaku w a klíče K užitých během vkládacího procesu tvoří maximální sadu vstupních parametrů viz. obr.2.2.

Existují různé typy vodoznakových systémů lišící se v počtu vstupních parametrů v kódovacím a dekódovacím procesu. Extrahovaný vodoznak $\widehat{w}$ se liší obvykle od vloženého vodoznaku w kvůli různým manipulacím. K tomu aby znalec korespondující oběma vodoznaky určil jejich totožnost, využívá srovnávací funkce $C\tau$, která srovnává originální vodoznak s extrahovaným vodoznakem. To je dáno mezi citlivosti τ:

$$C_\tau(\widehat{w,w}) = \begin{cases} 1, & c \geq \tau \\ 0, & 0 < \tau \end{cases} \quad (2.5)$$

Práh τ závisí na vybraném algoritmu a v dokonalém systému by měl být schopen zřetelně identifikovat vodoznak.



Obr.2.2 Blokové schéma dekodéru.

2.2 Terminologie užívaná ve vodoznakových aplikacích

2.2.1Robustnost

Robustnost algoritmu je definována jako schopnost detektoru vodoznakového systému, který je schopen extrahovat vložený vodoznak po běžném zpracování signálu. Systém by měl být navržen tak, aby byl odolný proti různým rušícím vlivům. Například v rádiovém vysílání, kdy vložený vodoznak musí být zachován i po zkreslení, které vzniklo během přenosu, včetně dynamické komprese a dolní propusti, protože detekce vodoznaku je udělána přímo z vysílaného signálu. Na druhé straně, v některých algoritmech je robustnost úplně nežádoucí a tyto algoritmy jsou označeny jako křehké vodoznaky (angl. fragile watermark). [2]

2.2.2 Křehký vodoznak

Jedná se vodoznak, který je vložen s velmi nízkou robustností. V takovém případě může být tento typ vodoznaku zničen dokonce i sebemenšími manipulacemi s vodoznačeným signálem. Takové vodoznaky jsou srovnatelné se skrytými zprávami ve stenografických metodách. Mohou být užívány pro kontrolu integrity objektů. [2]

2.2.3 Veřejný a soukromý vodoznak

Veřejný a soukromý vodoznak je rozlišován podle požadavku utajení pro klíč užívaný pro vložení a znovu získání.Podle základního principu vodoznačení, je stejný klíč užíván pro kódující a dekódující proces. Pokud je klíč známý, tento typ vodoznačení se nazývá veřejné vodoznačení, a pokud je klíč tajný tak se jedná o soukromé vodoznačení. Veřejné vodoznačení může být použito v aplikacích které nemají vysoké bezpečnostní požadavky (např. pro vložení meta informace)

2.2.4 Nevnímatelnost

Změny způsobené vložením vodoznaku by neměly přesáhnout práh citlivosti sluchu člověka. Je tedy důležité dobře zvolit práh, pod kterým vzorky vodoznaku nezpůsobí vnímatelné zvukové změny. Vodoznak je tedy považován za nevnímatelný, pokud je nepostřehnutelný lidskými smysly. Toto rozhodování a volba prahu je založeno na vlastnostech lidského sluchu (HAS). Nevnímatelnost si můžeme představit jako vnímavostní podobu mezi originální zvukovou nahrávkou a vodoznačenou.

2.2.5 Bitová rychlost vodoznaku

Bitová rychlost vkládaného vodoznaku je počet vložených bitů za jednotku času a je obvykle dáno v b/s (bps). Některé vodoznakové aplikace, jako je například kontrola kopií, vyžádaní vloženého výrobního čísla nebo ID autora, užívají průměrnou bitovou rychlost až 0.5bps. U některých aplikací, jako je tomu například u zvýraznění řeči ve zvukových nebo komprimovaných audio signálech, používají tyto algoritmy pro vložení vodoznaku přenosovou rychlost až 150kbps, tím tvoří významnou část v hostitelském zvukovém signálu. [1]

2.2.6 Odolnost

Nemělo by být možné bez znalosti použité metody a tajného klíče odstranit vodoznak nebo jej učinit nečitelným. Odolnost je myšlena také ve smyslu
odolnosti proti různým modifikacím zdrojových dat. Modifikace mohou být úmyslné (útoky) nebo neúmyslné (komprese, filtrace šumu, změna velikosti hlasitosti, atd.). Odolnost proti kompresi je velmi důležitá zejména v oblasti statických obrazů a
videa. Odolnost je jedna z velice důležitých vlastností vodoznaku.

2.2.7 Složitost

Složitost popisuje úsilí vynaložené na odstranění vodoznaku. Používaným parametrem pro vyhodnocení složitosti je množství času. Obecně je doporučeno navrhovat algoritmy vkládání vodoznaku tak náročné, aby jejich prolomení trvalo útočníkovi takovou dobu, po které by se odstranění vodoznaku stalo již bezvýznamné.

2.2.8 Spolehlivost detekce

Vodoznak by měl představovat dostatečný a spolehlivý důkaz o vlastnických právech testovaných dat.

2.2.9 Statistická nedetekovatelnost

Neautorizovaná osoba by neměla být schopna na základě statistických metod odstranit vodoznak. To znamená, že ani vlastnictví velkého počtu digitálních dat označených stejným vodoznakem, by nemělo umožnit jeho detekci. Možným řešením je použití obsahově nebo časově závislých vodoznaků.

2.2.10 Kapacita

Kapacita udává množství informace, která může být uložena do zdrojových dat. Obecně je množství informace, kterou je možné vložit do zdrojových dat omezené, proto se ve většině systémů s vodoznaky zvolí kompromis, který je závislý na typu zdrojových dat a konkrétní aplikaci.

Kapacita vodoznaku je velmi důležitá vlastnost a úzce souvisí s odolností. Pokud totiž zdrojová data obsahují velké množství vložených informací, stává se vodoznak v případě útoku snáze detekovatelným. Naproti tomu při vložení minimálního počtu informačních bitů, která jsou obsažena jen ve velmi malé oblasti zdrojových dat, je vodoznak prakticky odstraněn jakoukoli modifikací zdrojových dat. Je tedy vždy důležité dobře rozhodnout jaké množství vložené informace je vhodné pro konkrétní případ.

2.2.11 Detekce vodoznaku

V některých aplikacích, může detekční algoritmus využívat originální zvukové nahrávky, k tomu aby byl schopen extrahovat vodoznak ze sekvence vodoznačeného signálu. Tímto se výrazně zlepší detekční parametry, ve kterém původní zvuková nahrávka může být odečtena od kopie vodoznačené zvukové nahrávky. Tímto získáme sekvenci samotného vodoznakového signálu. Pokud detekční algoritmus nemá přitup k originální zvukové nahrávce, tak se tím výrazně zhorší detekční schopnosti algoritmu. Celý proces vkládání a extrahování vodoznaku si můžeme představit jako přenosový kanál. Při přenosu dat přes tento přenosový kanál může dojít ke zkreslení vodoznačených dat, to je způsobeno přítomnosti rušících vlivů jako jsou aditivní šum .

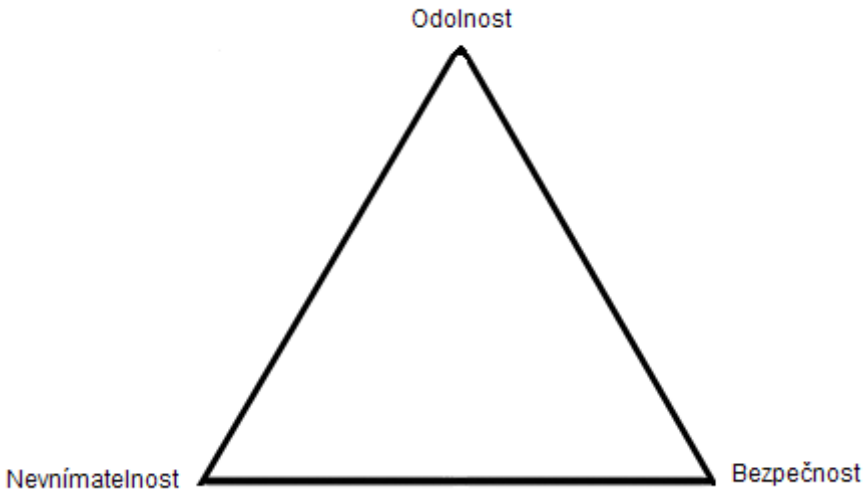
2.2.12 Bezpečnost

Vodoznačící systém musí zabezpečit vodznačený signál, tak aby protivník nemohl být schopen detekovat přítomnost vložených dat a nemohl tato data odstranit. Bezpečnost vodoznakových systémů je založena na stejném principu šifrovacích technik. Vodoznačená data nemohou být narušený, pokud ovšem neoprávněný uživatel nemá přístup k tajnému klíči, kterým byl vodoznak vložen. Dnešní systémy jsou však založeny na používání více kryptografických klíčů, což ztěžuje přístup a následné odstranění vodoznaku. Tímto neoprávněný uživatel nemůže být schopen extrahovat data v rozumném čase i když si je vědom, že hostitelský signál obsahuje vodoznak a je obeznámen s algoritmem vkládání. [4]

2.2.13 Výpočetní složitost

Implementace systému digitálního audio vodoznačení je obtížný úkol. Hlavní problém z technického hlediska je jeho výpočetní složitost vkládajících a detekčních algoritmů a počet vložení a detekcí vodoznaku užitého v systému. Například při vysílání musí být vkládání a detekce udělána v reálném čase, zatímco u autorskoprávních ochranných aplikacích není čas rozhodující faktor . [4]

Žádná ze současných technik digitálních vodoznaků však zatím nedokázala naplno splnit všechny výše popsané požadavky. Mezi tři nejdůležitější patří: nevnímatelnost, odolnost a bezpečnost, tyto požadavky jsou zobrazeny na „trojúhelníku požadavků“ (viz.obr. 2.3). Z Trojúhelníku požadavku je patro, že pokud jeden požadavek převažuje, tak zbylé dva jsou oslabené. Například požadavek na vysokou odolnost vodoznaku způsobí viditelné změny ve výsledných datech a naopak.



Obr.2.3 Trojúhelník požadavků na digitální vodoznak

2.4 Zkoumané problémy

Základní proces v každém systému vodoznačení může být modelován jako forma komunikace, kde zpráva je přenášena s vloženým vodoznakem do

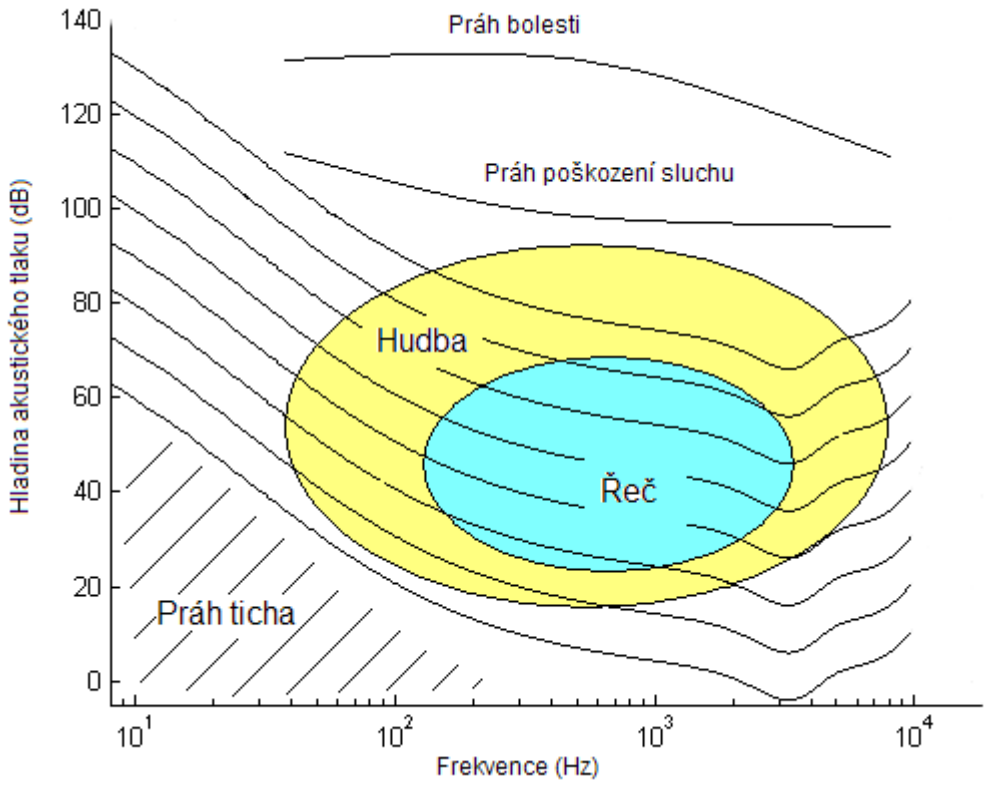
přijímače vodoznaku. Proces vodoznačení je vnímán jako přenosový kanál skrz, který je zpráva vodoznaku poslaná společně s hostitelským signálem. Při manipulaci s vodoznačenými daty jsou obvykle tyto data zkresleny nebo úmyslně poškozeny.

Hlavním zkoumaným předpokladem je proces vkládání a extrakce vodoznaků, který může být modelován jako komunikační systém, kde vkládání vodoznaku je modelováno jako vysílač, zkreslení vodoznaku je modelováno jako šum zahrnutý při přenosu a extrakce vodoznaku modelována jako detektor. Zde musíme také započítat vliv lidského sluchového systému HAS (human auditory system) a určení vnímavostních prahů, které jsou vypočítány užitím modelů ze zvukového kódování, jako je např. MPEG komprese. [1]

3 Fyziologie lidského ucha

3.1 Oblast slyšení

Zvuk má mnoho objektivně měřitelných vlastností, to však nemusí podávat dobrý obraz o tom, co člověk doopravdy slyší. Zvuk se šíří ve spojitém prostředí, to jsou molekuly, které teoreticky přenáší libovolnou směs frekvencí, každou o určité amlitudě a fázi. Avšak lidské ucho není schopno všechny tyto frekvence vnímat, proto při záznamu zvuku tedy není potřebné zaznamenávat všechny frekvence. Z hlediska zvukového vjemu je zvuk popsán prahovými hodnotami v čase, v úrovni akustického tlaku a kmitočtu. Na obr1 je popsána oblast slyšení, tj závislost úrovně akustického tlaku SPL (Sound Pressure Level)na kmitočtu.



Obr.3.1 Oblast slyšení

Oblast slyšení se nachází mezi prahem slyšitelnosti a prahem bolesti. Tato oblast je složena z oblasti řeči a hudby. Práh poškození sluchu je oblast, při jejímž překročení může dojít k poškození sluchu. Práh bolesti je hranice, při

jejímž překročení už lidské ucho není schopno vnímat zvuky a přitom dochází k trvalému poškození lidského sluchu. Křivka prahu slyšitelnosti je oblast zvuku, která udává, kdy je lidské ucho ještě schopno vnímat minimální hodnoty akustického tlaku. Tato křivka je u každého jedince odlišná, křivka se mění s věkem. Lidské ucho je schopno zachytit zvukový signál v rozsahu 20-20000 Hz. Nejcitlivěji zvuky vnímáme jak je patrno z obr.3.1 přibližně od 500Hz-5kHz. [2]

3.2 Člověk a vnímání zvuku

Člověk vnímá zvuk celým tělem, nejvíce však hlavou a uchem. Sluchové ústrojí je složeno ze tří částí. Vnější a střední ucho slouží pro zachycení a vedení zvukových vln, vnitřní ucho tyto zvukové vlny zpracovává pro mozek.

3.2.1 Hlava a vnější ucho

Zejména hlava, vnější ucho a zvukovod ovlivňují akustický tlak, který rozkmitává bubínek. Tělo a hlava ovlivňuje zvuky většinou blízké frekvenci 1500Hz. Vnější ucho provádí dvě činnosti: chrání bubínek a střední ucho proti poškození a umožňuje vnitřnímu uchu velmi blízce určit pozici. Zvukovod slouží jako rezonátor. Je dlouhý přibližně 2 cm; to odpovídá čtvrtině vlnové délky, což jsou frekvence blízké 4kHz.

3.2.2 Střední ucho

Zvuky ovlivňující vnější ucho se stávají z oscilací vzduchu. Tyto oscilace ovlivňují vnitřní ucho, které obsahuje tekutinu, tak že obklopuje nervové buňky. Střední ucho mění charakter vlnění a jeho výchylka je úměrná vzrůstajícímu tlaku. Toto vlnění je schopno uvést do pohybu tekutinu ve vnitřním uchu. Střední ucho je štěrbinová dutina mezi vnějším a vnitřním uchem. Jeho části tvoří bubínkový prostor, který je v přední části spojen Eustachovou trubicí a nosohltanem. Na straně zvukovodu tvoří její stěnu bubínek a následuje kónická membrána, která je vtažená směrem do středního ucha. Plocha bubínku činí asi 55 mm^2. Na druhém konci bubínkové dutiny se v kostěné stěně labyrintu nacházejí dvě okénka krytá pružnou blankou: horní oválné a spodní okrouhlé okénko. Uvnitř dutiny je řetěz tří středoušních kůstek, od bubínku v pořadí kladívko, kovadlinka a třmínek. [3]

3.2.3 Vnitřní ucho

Rozhraní mezi vnitřním a středním uchem tvoří blanka oválného okénka. Mechanické kmitání bubínku je převáděno pákovým systémem středního ucha na oválné okénko, které své kmitání přenáší na tekutinu vnitřního ucha. Na oválné okénko navazuje hlemýžď (clochlea). Hlemýžď si můžeme představit jako trubici, která je stočena do spirály a u člověka tvoří 2,75 závitu. Trubice je rozdělena na 3 části. Oblast scala vestibuli a scala timpani je vyplněna tekutinou, která je nazývána perylimfa. Oblast mezi nimi, scala media, je vyplněna endolymfou. Tyto tekutiny mají jiné koncentrace draslíku a sodíku a z tohoto důvodu mezi nimi vzniká elektrický potenciál 80 mV. Endolymfa a perilymfa je oddělena na jedné straně Reissnerovou membránou, na druhé straně kochleární přepážkou. Nosní struktura kochleární přepážky je tvořena basilární membránou, kde je umístěn Cortiho orgán. Cortiho orgán je smyslový orgán, který přenáší mechanické chvění na nervové buňky. Okrouhlé okénko vytváří zvukové vlny a ty se šíří podélně celým hlemýžděm. Membrána oválného okénka rozkmitává tekutinu hlemýždě, tím dochází ke vzniku stojaté vlnění. Membrána oválného okénka se chová jako spektrální analyzátor, výška tónu je dána místem, ve kterém je membrána rozkmitána. [3]

3.3 Hlasitost zvuku

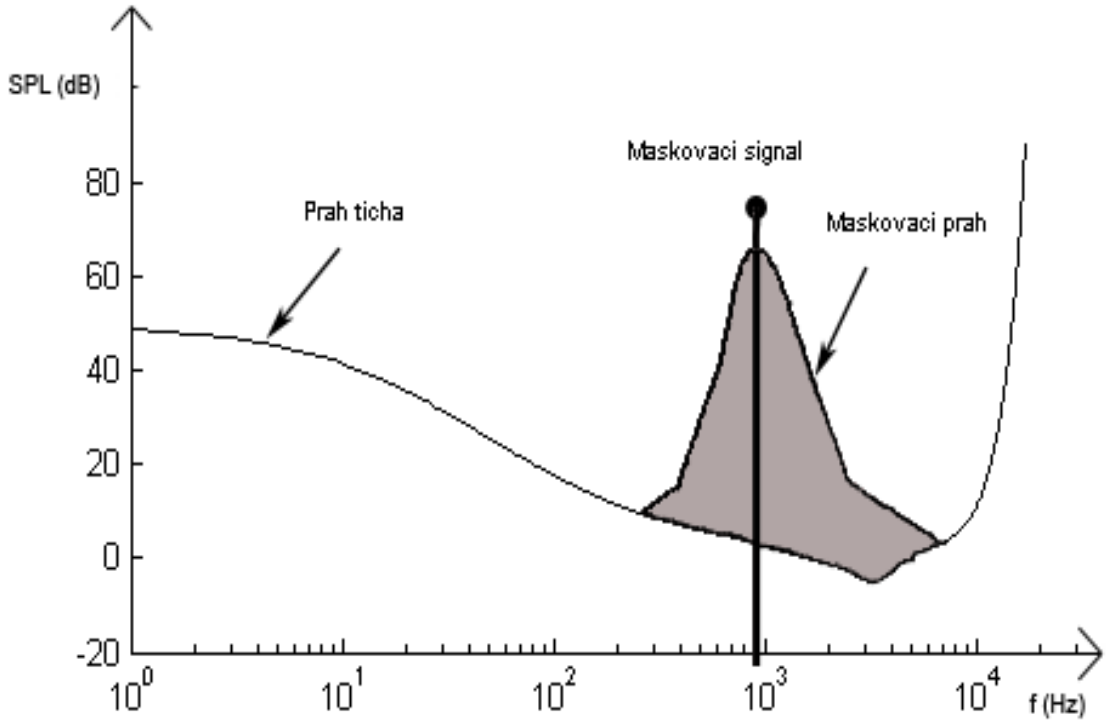
Změna tlaku vzduchu, která je zapříčiněna daným vlněním se označuje jako akustický tlak a měří se v Pascalech. Právě akustický tlak hraje hlavní roli při vnímání hlasitosti. Výsledky testů potvrdily, že člověk vnímá zvuk až od jistého (tzv. prahového) akustického tlaku. Pro frekvenci 1kHz je tento práh slyšitelnosti p_0 = 20,4μPa. Tato hodnota byla zvolena jako standard. Absolutní práh slyšitelnosti se od něj může lišit v závislosti kmitočtu vlnění a věku člověka.Intenzita zvuku je tedy množství zvukové energie, která projde danou plochou za jednotku času (W/m^2). Ta je vypočítána jako součin akustického tlaku a rychlosti částic vzduchu při šíření zvuku. Intenzita zvuku tedy klesá kvadraticky od zdroje zvuku. Celkové množství energie ve Wattech, kterou daný zdroj vyzáří do okolí se nazývá akustický výkon.

3.4 Maskování akustických signálů

Pokud do lidského ucha přicházejí současně dva zvuky, může sluchový vjem vyvolaný jedním z nich převládnout do té míry, že zeslabí nebo úplně potlačí vjem zvuku druhého. Zvuky s větší intenzitou zhoršují vnímání jiného přítomného zvuku s nižší intenzitou. Lidský sluch totiž zachytává zvuk jako součet všech zvukových signálů o určité frekvenci, amplitudě a fázi.

3.4.1 Frekvenční maskování (frequency masking)

Frekvenční maskování vzniká, pokud do lidského ucha přicházejí dva zvukové tóny s malým rozdílem frekvencí. Lidské ucho vnímá pouze tón o vyšší intenzitě a tón o nižší intenzitě je zamaskován (“viz. obr.3.2”). Avšak čistý sinusový signál těžko překrývá zvuk podobný šumu, zvuk podobný šumu snadno překrývá nevýraznou sinusovou složku. Důvodem je stavba lidského ucha, protože při rozkmitání bazilární membrány na jednom místě vzniká také chvění v jejím blízkém okolí. [1]



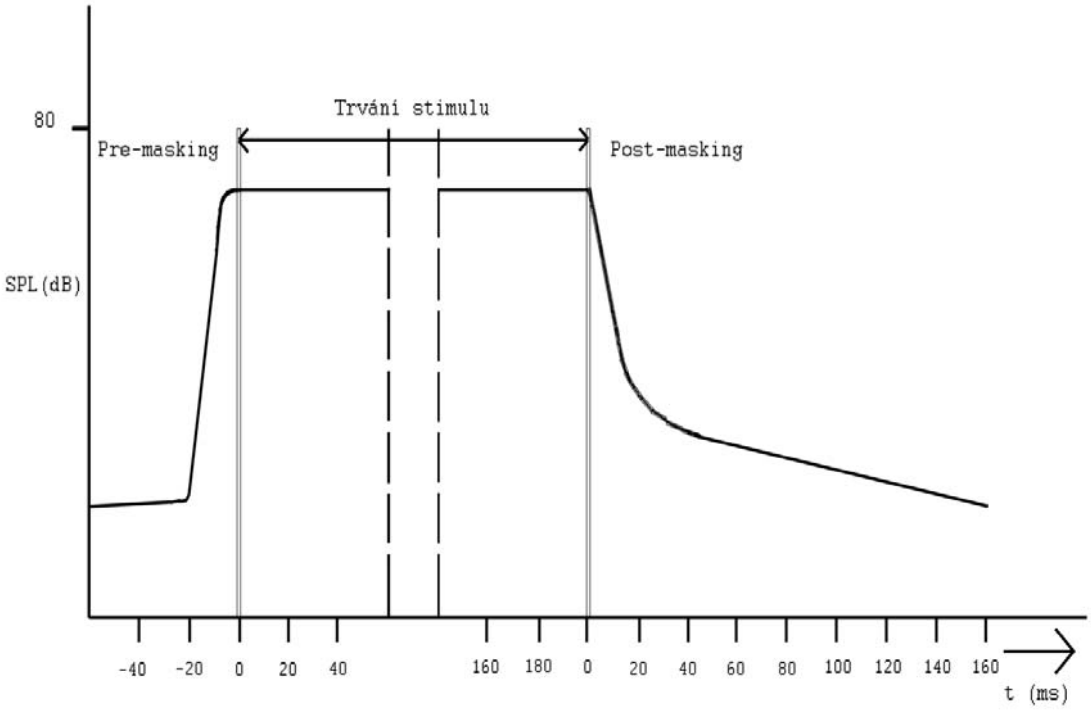
Obr.3.2 Frekvenční maskování

To je příčinou fyziologické detekce tónů s frekvencemi blízké dominantní frekvenci, které však mozek ignoruje. Pokud tedy do lidského ucha přicházejí

zvukové signály o frekvencích podobné dominantní frekvenci, tak je člověk na tyto frekvence málo citlivý. Jednou z mnoha měřítek této tolerance je Barkova stupnice dělící slyšitelné spektrum na 24 kritických pasem.

3.4.2 Časové maskování (temporal masking)

Maskovací efekt nastává v případě, když maskovaný krátkodobý signál určité hladiny přichází nejdéle do 5 ms před maskujícím signálem, déle trvajícím signálem, toto maskování se též nazývá premasking temporal. Podobně dochází k maskování v případě, objeví-li se maskovaný signál po maskujícím, opět déle trvajícím signálu, nazýván postmasking. Tohoto maskovacího efektu dosáhneme tehdy, jestliže maskovaný signál přichází do 10 ms po maskujícm, je-li tato doba delší, ne však delší jak 200 ms, dosáhneme částečného maskovacího efektu [1].



Obr.3.3 Časové maskování

3.5 Kritická pásma a Barkova stupnice

Lidské ucho funguje jako frekvenční analyzátor. To je způsobeno tím, že obsahuje bazilární membránu, která má svou charakteristickou frekvenci. Každý bod této membrány funguje jako pásmový filtr se střední frekvencí, šířkou pásma a sklonem. To byl důvod k vytvoření kritického pásma slyšení a vznikla tak Barkova stupnice (“viz. tab.3.1”), která koresponduje 24 kritickými pásmy reprezentujícími lidský sluchový systém. Tuto lze vypočítat podle následujícího vzorce

$$Bark = 13 \cdot arctan\left(\frac{0{,}0076f}{kHz}\right) + 3{,}5 \cdot arctan\left(\frac{f}{7500}\right)^2 \qquad (3.1) \qquad [3]$$

Tabulka 3.1: Rozdělení kritických pásem [3]

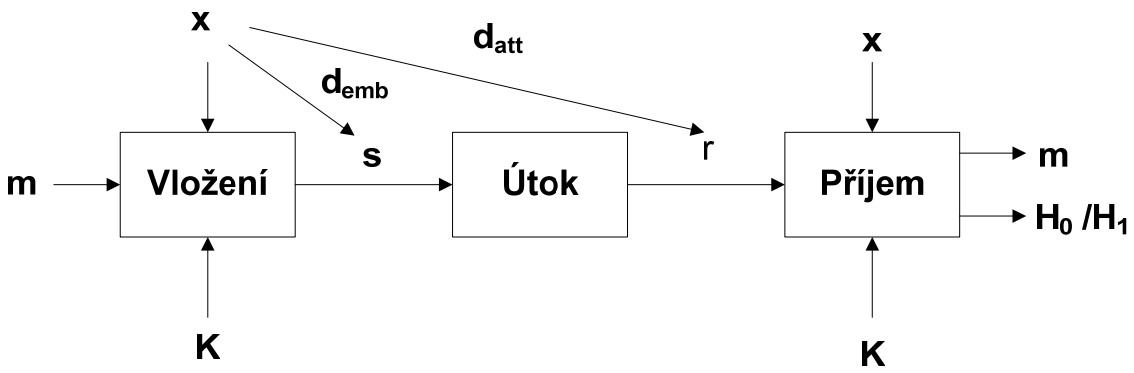
Bark	f (Hz)	fc(Hz)	ΔfG
0	0	50	100
1	100	150	100
2	200	250	100
3	300	350	100
4	400	450	110
5	510	570	120
6	630	700	140
7	770	840	150
8	920	1000	160
9	1080	1170	192
10	1270	1370	210
11	1480	1600	240
12	1720	1850	280
13	2000	2150	320
14	230	2500	380
15	2700	2900	450
16	3150	3400	550
17	3700	4000	700
18	4400	4800	900
19	5300	5800	1100
20	6400	7000	1300
21	7700	8500	1800
22	9500	10500	2500
23	12000	13500	3500
24	15500		

Kritická pásma pro lidské slyšení jsou pro nižší frekvence úzká a široká pro vyšší frekvence. K vnímání hlasitosti přispívají všechna spektrální maxima, ale ne všechny stejnou mírou. V rámci jedné frekvenční skupiny hlasité části do značné míry maskují slabší části. Velikost kritického pásma byla zjištěna v experimentech s maskováním signálního tónu šumem. Výsledkem těchto experimentů je šířka pásma šumu, při které již tón není slyšet. K maskování dochází v okamžiku, kdy se šířka pásma šumu rovná šířce pásma sluchového filtru, taková šířka pásma se nazývá kritické pásmo[3].

4 Všeobecný návrh vodoznaku

4.1 Hlavní model digitálního vodoznaku

Obr.4.1 ukazuje celkový pohled na hlavní model digitálního vodoznačení. Vodoznaková zpráva $\mathbf{m}$ je vložena k hostitelskému signálu $\mathbf{x}$ tím se vytvoří vodoznakový signál $\mathbf{s}$. Vkládající process je závislý na tajném klíči $\mathbf{K}$ a musí splňovat požadovanou vnímavostní transparentnost, tj. subjektivní rozdíl kvality mezi $\mathbf{x}$ a $\mathbf{s}$ spočítané jako vložené zkreslení ($\mathrm{d_{emb}}$), které musí být vloženo pod maskovacím prahem.



Obr.4.1 hlavní model digitálního vodoznačení

Před detekcí vodoznaku a dekódováním, je vodoznakový signál $\boldsymbol{s}$ obykle úmyslně modifikován. Úmyslné modifikace jsou obvykle uváděny jako útoky; tyto útoky jsou tvořeny útočným zkreslení d_{att} ve vnímavostní přijatelné úrovni. Po útocích, vodoznakový extraktor příjmá napadený signál $\boldsymbol{r}$. Proces vodoznakové extrakce se stává ze dvou sub-procesu, první, vodoznakové dekódování příjmané vodoznakové zprávy $\hat{m}$ používající klíč $\boldsymbol{K}$, a, druhý, vodoznakové detekce.

4.2 Vkládání vodoznaku podle psychoakustického modelu

Ve vodoznačných aplikacích musíme dbát na to, aby vložený vodoznak byl nevnímatelný. Proto využíváme psychoakustického modelu, který je založen na nedokonalosti lidského ucha. Audio signál je vodoznačen s unikátní pseudonáhodnou sekvencí mající určitý tvar podle psychoakustického modelu. Takto sestavený vodoznak reprezentován ve spektrální a časovém tvaru využívající maskujících efektů v lidském sluchovém systému (HAS - Human Auditory System). Délka N audio signálu je prvně segmentovaná na bloky si(k) o délce 512 vzorků, $i = 0,1,\dots,\left\lfloor\frac{N}{512}\right\rfloor$, a k=0,1,…,511. Z bloku vzorků je určen podle frekvenčního maskovacího modelu minimální maskovací práh. Můžeme použít i bloky o délce 1024 vzorků. Tento algoritmus pracuje následovně. Pro každý audio segment si(k):

- Vypočítat výkonové spektrum Si(k) audio segmentu si(k);
- Vypočítat frekvenční masky Mi(k) výkonového spektra Si(k);
- Použít masku Mi(k) k váhování šumové sekvence vodoznakové zprávy pro daný audio blok a vytvořit tak tvarovanou vodoznakovou zprávu Pi(k) = Yi(k)M(i);
- Takto upravený vodoznak připojit k nosnému audio signálu příslušného rámce
- Vypočítat inverzní FFT (IFFT) frekvenčně upraveného vodoznačeného signálu.

Vodoznakový zvukový signál je určen k přenesení přes rozmanitý počet kanálů. Přenosový kanál může představovat aditivní šum, převedený několikrát z digitálního do analogového a z analogového do digitálního, nebo

může být dokonce užito psychoakustického sluchového modelu. Vodoznakový signál by měl být tedy vložen s co nejvyšší možnou intenzitou, abychom dosáhli maximální robustnosti, aby se vodoznaková zpráva dochovala v přenosovém kanálu a mohli jsme správně určit vodoznakovou zprávu.

4.3 Detekce vodoznaku

Při extrakci vodoznaku může nastat výsledek detekčního systému:

- Hit – Systém deklarován tak, že existuje signatura, pokud je vstup vodoznačen s totožnou signaturou.
- Správné potlačení – systém rozhoduje tak, že správná signatura neexistuje, pokud vstup není vodoznačen nebo je vodoznačen s odlišnou signaturou.
- Miss – Systém rozhoduje tak, že správná signatura není, i když je vstup vodoznačen se správnou signaturou.
- False alarm – systém deklaruje existující signaturu, i když vstup není vodoznačen nebo je vodoznačen odlišnou signaturou. [8]

Rovnocennost mezi vodoznakem a přirozeným aditivním šumem ukazuje to, že charakteristická šifra z originálního signálu je rozdílná, její rozdíl je dán různým rušením nebo úmyslnými útoky, které vodoznakový signál zdeformují, aby se záměrně zabránilo detekci. Předpokládejme $r(n), 0 \leq n \leq N-1$ je segment o délce N vzorků, a náš cíl je zkontrolovat, zda tento segment obsahuje šifru. Předpokládejme přesné zarovnání testovaného segmentu a přístup k originálnímu segmentu. Pomocí korelační funkce vypočítáme mezi testovaným signálem a originálním správné zarovnání. V tomto případě může být $r(n)$ vyjádřeno jako $r(n) = s(n) + d(n), 0 \leq n \leq N-1$, kde $s(n)$ je identické k originálnímu signálu a d(n) obsahuje pouze šum, nebo šum se šifrou. Detekční mechanismus je založen na faktu, že během ověřování máme přístup k originálnímu signálu, a také máme přístup k tajnému klíči, který potřebujeme k dekódování šifry. Detekci provádíme zkoušením dvou předpokladů:

$$H0: u(n) = r(n) - s(n) = v(n) \tag{4.1}$$

- nenalezena vodoznaková zpráva

$$H1: u(n) = r(n) - s(n) = w'(n) + v(n) \tag{4.2}$$

- nalezena vodoznaková zpráva.

$w'(n)$ je šifra,která mohla být pozměněná a $v(n)$ je šum. Pro správný předpoklad je vybráno podle výpočtu korelace mezi $u(n)$ a originální signaturou $w(n)$ podle:

$$Corr(u,w) = \frac{\sum_{j=0}^{N-1} u(f) \cdot w(f)}{\sqrt{\sum_{j=0}^{N-1} u(f)^2 \cdot \sum_{j=0}^{N-1} w(f)^2}} \tag{4.3}$$

Pro ověření podobnosti může být užito následujícího vztahu:

$$Sim(u,w) = \frac{\sum_{j=0}^{N-1} u(f) \cdot w(f)}{\sum_{j=0}^{N-1} w(f)^2} \tag{4.4}$$

Hodnota podobnosti není omezena [-1 1] jako korelační funkce, ale může být jakákoliv hodnota v intervalu $(-\infty, +\infty)$. Korelační a podobnostní měření jsou odlišné. Jsou citlivé na dané zkreslení a proto musí mít dobré rozhodovací schopnosti. Hlavní otázkou detekce systému je nastavení vhodného prahu pro korelační a podobnostní funkci.

4.4 Metoda rozprostřeného spektra

Metoda rozprostřeného spektra (spread-spectrum) představuje originální metody pro maskování informace. Původně bylo určeno pro rádiový přenos a zvýšení odolnosti proti rušení, avšak tyto aplikace jsou spíše často užívány při přenosu digitálních informací. Aplikace založené na metodě rozprostřeného spektra se používají u bezdrátových sítí, rádiových modemů, digitální telegrafie,

mobilní telefonické komunikace, ale mohou být aplikovány i na multimediální data. Technika rozprostřeného spektra má dobré vlastnosti při potlačování rušení během přenosu. Využívá se také pro zamaskování signálu proti neoprávněnému posluchači. Uchování informace v soukromí je velmi podobná i ve vodoznakových aplikacích. Těchto metod je také často využíváno ve vývoji vodoznakových algoritmů. Rozprostřené spektrum originálního audio signálu může být považováno jako interferující rušení se signálem, který nese vodoznakovou informaci.

Rozprostřené spektrum pseudonáhodně rozprostře informační signál do širšího pásma frekvencí, tím ho lze obtížněji rušit a odposlouchávat. Obecný model rozprostřeného spektra (viz.obr.4.2) splňuje obecný model symetrické kryptografie a proto lze rozprostřeného spektra lze použít i pro utajování dat.

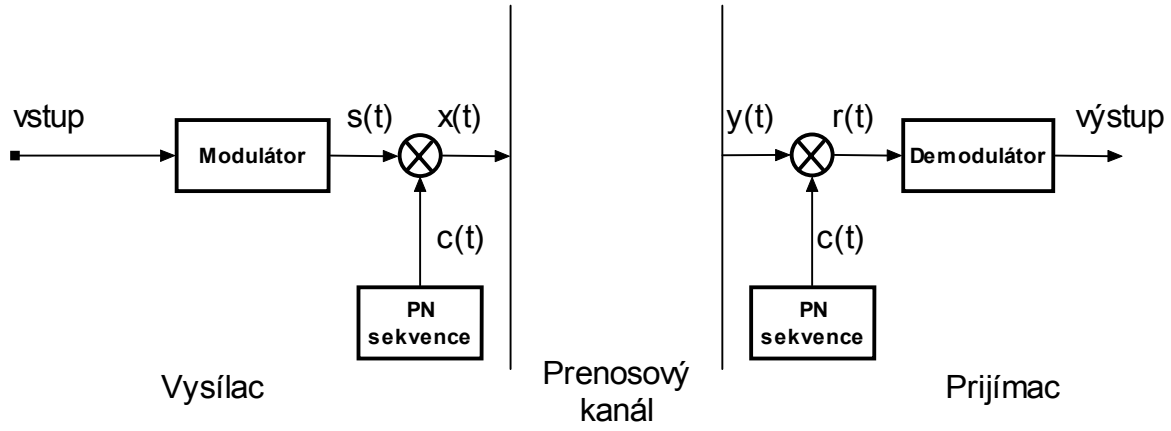
Rozprostřené spektrum dělíme na:

- Frequency hoping Spread Spectrum (FHSS)

Je to technika, která byla původně vyvinutá pro vojenské a zpravodajské použití na počátku 2. Světové války. Signál se vysílá náhodně se střídanou posloupností frekvencí. [11]

- Direct Sequence Spread Spectrum (DSSS)

Této techniky rozprostření využívají např. buňkové telefonní sítě při multiplexování technikou CDMA (Code division Multiple Access). Každý bit dat se reprezentuje ve vysílaném signálu více bity vysílanými za původní časový interval. [11]



Obr.4.2 Obecné schéma rozprostřeného spektra

Vstupní data jsou předány kanálovému kodéru, který generuje analogový signál s úzkou šířkou pásma. Tento signál je dále modulován pomocí posloupnosti číslic generovaných generátorem pseudonáhodných čísel. Modulací dojde ke zvětšení šířky pásma vysílaného signálu. Přijímač stejnou pseudonáhodnou posloupností číslic získaný signál demoduluje, takto získaný signál je předán kanálovému dekodéru, který obnoví původní data.

Modulace rozšířeného spektra je speciální metoda vodoznakové modulace. Modulace je vykonána na **C0**, kde je transformační blok vzorků **c0.** Tato transformace je užívána v modelu audio signálu s ortonormální základní funkcí zahrnující odstup signálu. V tomto případě Fourierova transformace je užívána jako základní funkce a transformované bloky jsou vypočteny z Fourierových koeficientů reprezentující vektor **C0**. Každý bit $k \in \{0,1\}$ je modulován pseudonáhodným **pn**k vektorem skládajícího se ze dvou stejně pravděpodobných elementů $\{-1,+1\}$ generované pomocí tajného klíče. Proto, očekávaná hodnota pseudonáhodné sekvence je $E\{\boldsymbol{pn}_k\} = 0$. Obvykle pseudonáhodné sekvence pro dva bity jsou invertovány $\boldsymbol{pn}_0 = -\boldsymbol{pn}_1 = \boldsymbol{pn}$**.** Originální signál $\boldsymbol{c_0}$ je rozdělen na části $M = \left\lfloor \frac{l(c_0)}{N} \right\rfloor$ bloků $\boldsymbol{c_{0j}}$, $0 \leq j \leq M-1$ o N vzorcích.

K ulehčení budeme uvažovat jeden blok $(\boldsymbol{c_0} := \boldsymbol{c_{0j}})$ nesoucí 1 bit vodoznaku.

1. Blok $\boldsymbol{c_0}$ je transformován ortogonální transformaci τ v odpovídající oblasti $\boldsymbol{C_0}$.

$$\boldsymbol{C_0} = \tau(\boldsymbol{c_0}) \tag{4.5}$$

2. PN sekvence $\boldsymbol{pn_k}$ je váhováno s α k přizpůsobení kvality a robustnosti.

$$\boldsymbol{W} = \boldsymbol{\alpha pn_k} \tag{4.6}$$

3. Modulovaný a váhovaný vodoznakový signál je přičten k zpracovanému signálu transformované oblasti.

$$C_{\boldsymbol{w}} = \boldsymbol{C_0} + \boldsymbol{W} \tag{4.7}$$

4. Vodoznačný signál je transformován zpět do časové oblasti.

$$\boldsymbol{c_w} = \boldsymbol{\tau^{-1}}(\boldsymbol{C_w}) \tag{4.8}$$

Během detekujícího kroku, musí být tentýž vektor $\boldsymbol{pn_k}$, $k = 0{,}1$ generovaný přes tajný klíč. Porovnávací funkce je užívána k tomu, aby bylo rozhodnuto o přítomnosti vloženého vektoru $\boldsymbol{pn}$. Tento požadavek je dokonalou synchronizací s vloženým blokem vzorků.

1. Synchronizace se začínajícím vloženým blokem $\boldsymbol{c_w}$;
2. Transformace $\boldsymbol{c_w}$ do vkládací oblasti $\boldsymbol{C_w} = \tau(\boldsymbol{c_w})$;
3. Korelace $\boldsymbol{C_w}$ s $\boldsymbol{pn}_k, k = 0{,}1$ podle použité komparační funkce C_τ

$$C_\tau(\boldsymbol{C_w}, \boldsymbol{pn}) = C_\tau(\boldsymbol{C_0}, \boldsymbol{pn}) + C_\tau(\alpha \boldsymbol{pn}, \boldsymbol{pn}) \tag{4.9}$$

4. Detekci přenášeného bitu, obvykle provádíme podle znaménka v komparační funkci

$$\boldsymbol{sign}\big(C_\tau(\boldsymbol{C_w}, \boldsymbol{pn})\big) \begin{cases} > 0, & pro\ \boldsymbol{pn_0} \\ < 0, & pro\ \boldsymbol{pn_1} \end{cases} \tag{4.10}$$

První hodně užívanou komparační funkcí C_τ je lineární korelace

$$C_\tau(\boldsymbol{X},\boldsymbol{Y}) = \langle \boldsymbol{X},\boldsymbol{Y}\rangle = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{X}[i]\boldsymbol{Y}[i] \tag{4.11}$$

Se signálem vektoru $\boldsymbol{X}$ a $\boldsymbol{Y}$**.** Výsledek korelace se sestává ze dvou příspěvků $C_\tau(\boldsymbol{C_0}, \boldsymbol{pn})$ a $C_\tau(\ \alpha\boldsymbol{pn}, \boldsymbol{pn})$. Druhý člen akumuluje příspěvky s pseudonáhodnou sekvencí vloženou v odlišné základní funkci, kde první člen reprezentuje korelaci nebo vzájemné působení nosného signálu s pseudonáhodnou sekvencí. Pokud pseudonáhodná sekvence je rozdělena na části do dvou sekvencí rovnající se kladným a záporným elementům, korelace $C_\tau(\boldsymbol{C_0}, \boldsymbol{pn})$ může tedy být zapsána takto

$$C_\tau(\boldsymbol{C_0}, \boldsymbol{pn}) = \sum_{i=1}^{N/2}\boldsymbol{C}_0^{+}[\boldsymbol{i}] - \boldsymbol{C}_0^{-}[\boldsymbol{i}] = \frac{(\boldsymbol{\mu}^{+} - \boldsymbol{\mu}^{-})}{2} \tag{4.12}$$

S $\boldsymbol{\mu}^{+}$ a $\boldsymbol{\mu}^{-}$ označuje průměrnou hodnotu. Podle nejdůležitější poučky o limitě, je rozdělení podle průměru normální pokud N je dostatečně velké. Kromě toho, odlišnost dvou normálních rozdělení je tedy normální s $N\left(\mu_{C_\tau}, \sigma_{C_\tau}\right)$. Potom $\boldsymbol{C_0}$ a $\boldsymbol{pn}$ jsou dvě nezávislé náhodné proměnné, průměr μ_{C_τ} a změna σ_{C_τ} může být spočítána podle

$$\mu_{C_\tau} = E\{C_\tau(\boldsymbol{C_0}, \boldsymbol{pn})\} = E\{\boldsymbol{C_0}\}\,E\{\boldsymbol{pn_k}\} = 0 \tag{4.13}$$

$$\sigma_{C_\tau}^2 \approx \hat{\sigma}^2_{\frac{(\boldsymbol{\mu}^{+} - \boldsymbol{\mu}^{-})}{2}} = \hat{\sigma}^2_{\frac{(\boldsymbol{\mu}^{+} - \boldsymbol{\mu}^{-})}{2}} = \hat{\sigma}^2_{\mu C_0} = \frac{\hat{\sigma}^2_{C_0}}{N} \tag{4.14}$$

Podle užitého modelu je distribuční funkce $N\left(0, \sigma\boldsymbol{c_0}/\sqrt{\boldsymbol{N}}\right)$ v nevodoznačeném případě a předpokládáme pevné váhování $\alpha := \{\alpha\}_{i=1}^{N}$ pseudonáhodné sekvence, pravděpodobnost distribučního rozložení funkce pro dvě odlišné sekvence je

$$f_{\boldsymbol{pn_1}}(t) = \frac{1}{\sqrt{2\pi\sigma c_r}} e^{-\frac{(t-\alpha)^2}{2\sigma^2 C_r}} \qquad f_{\boldsymbol{pn_0}}(t) = \frac{1}{\sqrt{2\pi\sigma c_r}} e^{-\frac{(t+\alpha)^2}{2\sigma^2 C_r}} \tag{4.15}$$

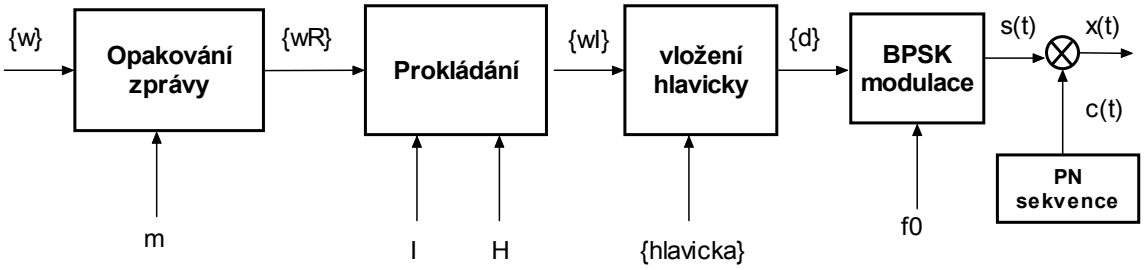
Vyskytované chyby detekované v bitech jsou pokud $C_\tau(\boldsymbol{C_0}, \boldsymbol{pn}) > C_\tau(\boldsymbol{\alpha C_0}, \boldsymbol{pn})$. Proto jsou tedy chyby pravděpodobně signalizovány získáním

$$P_{fa} = P_{01} + P_{10} = p_0 \int_\tau^{+\infty} f_{\boldsymbol{pn_0}}(t)dt + p_1 \int_{-\infty}^{\tau} f_{\boldsymbol{pn_1}}(t)dt \tag{4.16}$$

Kde P_{01} representuje chybu je vysílán 0 bit a 1 bit je detekován a P_{10}shodně.

5 Generace vodoznaku

Cílem generování vodoznaku je vytvoření šumového signálu x(t) obsahující data, která jsou upravena podle psychoakustického modelu a následně vkládána k originálnímu zvukovému signálu. K vytvoření vodoznakového signálu x(t) je použita technika rozprostřeného spektra využívající BPSK modulaci. Tento proces je zobrazen na obr.5.1.



Obr.5.1 Blokové schéma pro vytvoření vodoznakové informace

Kde:

- {*w*} originální vodoznak
- m je hodnota opakování bitů

- $\{w_R\}$ je upravený vodoznak po opakovacím kódování
- I, H jsou rozměry prokládací matice
- $\{W_I\}$ vodoznak vytvořený prokládací maticí
- $\{hlavička\}$= sekvence jedniček
- $\{d\} = \{hlavička\} + \{w_I\}$ vodoznaková sekvence určena pro použití rozprostřeného spektra
- $f_0 =$ frekvence použita pro BPSK modulaci

Mějme například zprávu která bude obsahovat 2 osmibitové znaky:

$\{w\} = \{$ 1 1 -1 1 -1 -1 1 -1 | 1 1 -1 1 1 1 -1 -1 $\}$

Pro větší robustnost vodoznaku použijeme opakovacího kódování. Opakovací kódování odesílá m bitů se stejnou hodnotou d pro každý bit vodoznakové zprávy. Zvolme například opakování $m = 3$, tím získáme bitovou posloupnost $\{w_R\}$.

$\{w_R\} = \{$
1 1 1 1 1 1 -1 -1 -1 1 1 1
-1 -1 -1 -1 -1 -1 1 1 1 -1 -1 -1
1 1 1 1 1 1 -1 -1 -1 1 1 1
1 1 1 1 1 1 -1 -1 -1 -1 -1 -1 $\}$

Prokládání neboli interleaving se používá jako doplněk kanálového kódování, kvůli ochraně proti skupinovým chybám (shlukům chyb), které vznikají pří přenosu sdělovacím kanálem (např. vlivem impulsního rušení).
Jednotlivé bity nebo i několikabitové symboly kódového slova o délce N, které přicházejí z kodéru pro dopředné potlačení chyb (FEC), jsou v prokladači zpožděny o různý čas (toto zpoždění souvisí s hloubkou prokládání H). Proto se při přenosu nevyskytují jednotlivé symboly jednoho kódového slova těsně za

sebou a případný shluk chyb je tudíž rozprostřen mezi více kódových slov. Tyto izolované chyby je pak již možné opravit. [13]

Tím se zlepšují vlastnosti, tak aby byl systém odolný proti rušení. Blok prokládání s délkou l=5 a hloubkou H=10 je ukázáno v tab.5.1 Kódované symboly jsou zapsány do prokládací matice podél sloupců, zatímco přenášené symboly jsou čtené z matice podél řádků. Pokud kódovaná sekvence symbolu je x1,x2,x3…xn, tak výstupem prokládací matice je posloupnost x1,x11,x21,…xn. Příjmač provede inverzní proces tím, že zapíše symboly do řad a čte po sloupcích.

Pokud je velikost bitové posloupnosti $\{w_R\}$ menší než velikost prokládací matice, musíme do volného místa doplnit jedničky. Užitím prokládací matice, získáme sekvenci {wl}:

$\{W_l\} = \{$ 1 1 1 -1 1 1 1 -1 -1 1 1 -1
-1 -1 -1 1 -1 -1 1 -1 1 -1 1 1
-1 1 -1 1 1 -1 -1 -1 1 1 -1 -1
-1 1 1 -1 -1 1 1 1 1 1 1 1
1 1 }

Tab.5.1 Tabulka prokládací matice o rozměrech l=5,H=10

X1	X11	X21	X31	X41
X2	X12	X22	X32	X42
X3	X13	X23	X33	X43
X4	X14	X24	X34	X44
X5	X15	X25	X35	X45
X6	X16	X26	X36	X46
X7	X17	X27	X37	X47
X8	X18	X28	X38	X48
X9	X19	X29	X39	X49
X10	X20	X30	X40	X50

1	1	1	-1	1
1	1	-1	-1	1
1	-1	-1	-1	-1
1	-1	-1	1	-1
1	-1	1	1	-1
1	-1	1	1	-1
-1	-1	1	1	-1
-1	-1	1	1	-1
-1	1	1	1	1
1	1	1	1	1

Pro správnou synchronizaci vodoznačné zprávy v přijímači musíme k posloupnosti $\{w_I\}$ přičíst hlavičku k vodoznaku hlavičku. Tato hlavička je složena pouze z posloupnosti jedniček.

$\{hlavička\} = \{ \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad \}$

Výsledná data vodoznakové sekvence {d} jsou získaná sloučením hlavičky a W_I:

$$\{d\} = \{hlavička\} + \{w_I\} \tag{5.3}$$

$$\{d\} = \{ \begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{matrix} \}$$

BPSK modulace je provedena podle následujícího vztahu:

$$s(t) = d(t)cos(\omega_0 t) \tag{5.1}$$

Rozprostření je uděláno podle:

$$x(t) = c(t)s(t) \tag{5.2}$$

Kde $c(t)$ je PN sekvence, která může být generována podle různých způsobů. Obvykle je užíváno pseudonáhodného číselného generátoru. Důležitý faktor je to, aby vysílač i přijímač měli kopii celé PN sekvence {c}. Takto získaný signál musíme upravit podle minimálního maskovacího prahu, který je vypočten pomocí psychoakustického modelu pro MPEG layer I.

6. Výpočet psychoakustického modelu pro MPEG layer I

Psychoakustický model je aplikován při kompresi audio dat s využitím frekvenčních maskovacích efektů, tímto zajistíme neslyšitelnost kvantizačního šumu podle maskovacího prahu. Ve skutečnosti jsou používané právě existující

modely pro tvarování vodoznakového šumu. Různé psychoakustické modely se liší v komplexnosti a implementaci různých maskovacích efektů. Často užívané modely jsou psychoakustické modely typu 1 pro vrstvy I a II podle ISO-MPEG pro vzorkovací frekvenci $Fs = 41.1kHz$. Psychoakustický model má dva hlavní úkoly. Vybrání typ bloku, který bude při kódování použit a vypočet signal to mask ratio (SMR) v každém dílčím pásmu. To vyžaduje určení maximální úrovně signálu a minimálního maskovacího prahu v každém dílčím pásmu.

Vytvořený vodoznakový signál $x(t)$ užitím rovnice 5.2 musíme upravit podle psychoakustického modelu. Nedbale přidaná vodoznaková sekvence, která je šumového charakteru může ve vodoznačeném zvukovém signálu způsobit nepříjemný slyšitelný zvuk, protože lidské ucho je citlivé i na zvukové vlny, které mají velmi malou energii. Vodoznakový signál, který je vkládán s velmi malou energii vytváří metodu rozprostřeného spektra málo robustní. Řešením k zajištění neslyšitelnosti je tvarování vodoznakové sekvence podle psychoakustického modelu [9] viz.obr.6.1.

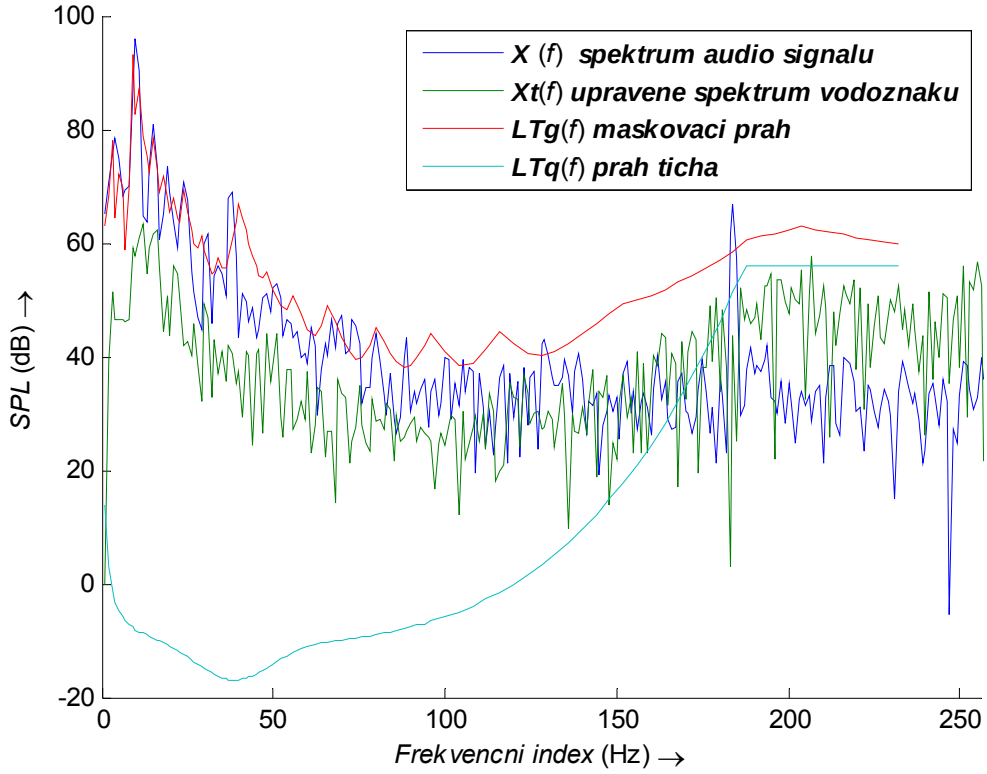
Výstupem psychoakustického modelu je spočítané SMR (Signal to Mask Ratio), určující maximální velikost šumu, který je ještě maskován.

SMR je počítáno pro všechny subpásma takto:

$$SMR(n) = L(n) - LT_{Min}(n) \ [dB] \tag{6.1}$$

Tato informace je užívaná pro ztrátovou kompresi zvukových signálů k opakované alokaci bitů v každém subpásmu. To však není nutné v každém případě u aplikací vodoznačení, protože je důležitý maskovací práh pouze pro každy blok. Proto tedy pro integraci psychoakustického modelu jsou požadovány pouze následující kroky:

- Výpočet výkonového spekra
- Identifikace tónových (sinusoid-like) a netónových (noise-like) složek
- Decimace masek pro eliminaci všech irelevantních masek.
- Výpočet individuálních maskovacích prahů
- Výpočet globálního maskovacího prahu.
- Určení minimálního maskovacího prahu pro každé pásmo.



Obr.6.1 upravené spektrum vodoznakové zprávy pomocí psychoakustického modelu

6.1 Výpočet výkonového spektra

Maskovací práh je odvozen z odhadu výkonu spektra (PSD - power density spectrum), tímto je model převeden z časové oblasti do frekvenční pomocí algoritmu FFT. Spočítaný odhad výkonového spektra vstupního bloku je násoben Hannovým oknem, aby bylo dosaženo minimálního prosakování spektra. Hannovo okno je definováno takto:

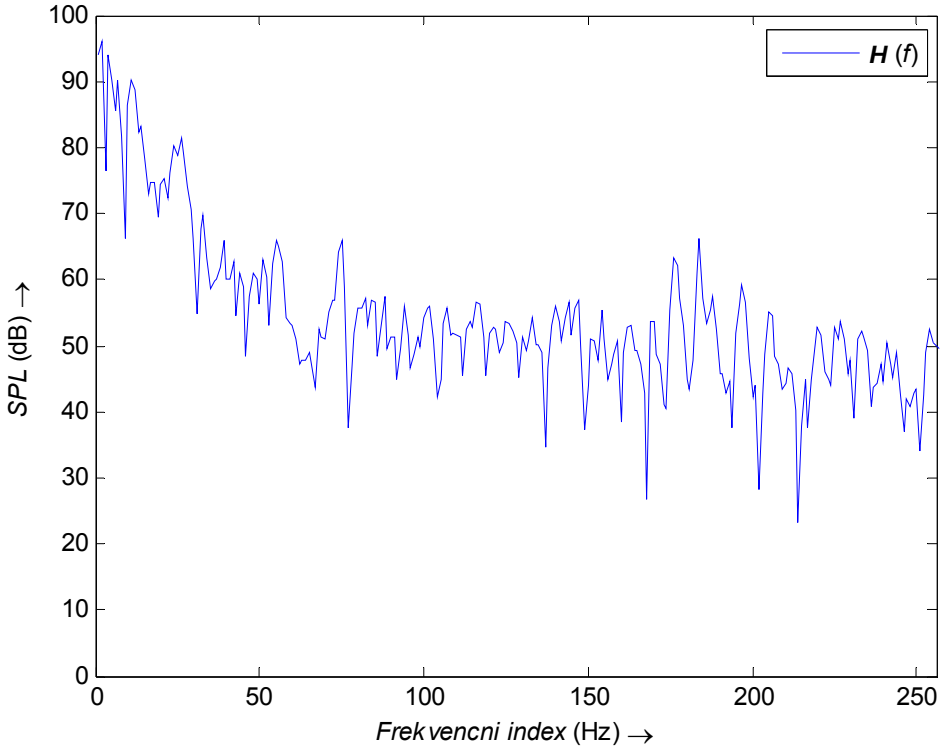
$$h(i) = \sqrt{\frac{8}{3}}\left[1 - cos\left(\frac{2\pi i}{N}\right)\right], \qquad \mathrm{i} = 0,1, \ldots \mathrm{N} - 1 \tag{6.2}$$

MPEG Layer I užívá vstupní blok $s(l), l = 1 \dots N$ o délce $N = 512$, pro layer II se používá vstupní blok o délce $N = 1024$ vzorků.

PSD je spočítáno pomocí FFT:

$$X(k) = 10 \log_{10} \left| \frac{1}{N} \sum_{i=0}^{N-1} h(l) s(l) e^{-i\frac{2\pi kl}{N}} \right|^2 \ [dB] \qquad k = 0 \dots \frac{N}{2} \tag{6.3}$$

Tímto byl získán vstupní blok dat, která jsou potřebná pro výpočet maskovacích prahů. Maximální hodnota výkonového spektra $X(k)$ je normalizováno o hodnotu +96dB SPL (Sound Pressure Level). To je provedeno tak, že maximální hodnota koresponduje k hodnotě 96 dB.



Obr. 6.2 Spektrum signálu

6.2 Výpočet sound pressure level

Sound pressure level Lsb v daných pásmech n je počítáno takto:

$$L_{sb}(n) = max\{X(k), 20 * \log_{10}[scf(n) * 2^{15}] - 10\} \ [dB] \tag{6.4}$$

Pro $X(k)$v dílčím pásmu n, kde $X(k)$ je úroveň akustického tlaku spektrálních čar o indexu k z FFT s maximální amplitudou ve frekvenčním rozsahu korespondující subpásmu n. Výraz $scf(n)$ je využíván v Layer I představující scale factor pro subpásmo n. V každém pásmu se z 12 vzorků vybere jeden vzorek s maximální amplitudou a podle něj se stanoví činitel měřítka (scale factor). Podle tabulky scale faktorů vyhledáme pouze jednu nejvyšší úroveň za periodu času. Faktor násobíme konstantou 2^{15} to je normalizováno k $+96dB$. Hodnota $-10dB$ provádí korekci odlišnosti mezi špičkou a RMS úrovní. Sound pressure level Lsb je počítán pro každé subpásmo n. [4]

6.3 Práh slyšitelnosti

Práh slyšitelnosti (threshold in quiet) LTq je definován, jako závislost akustického tlaku při níž lidský sluch přestává vnímat sinusový akustický signál.

$$LTq = 3{,}64\left(\frac{f}{kHz}\right)^{-0{,}8} - 6{,}5e^{-0{,}6\left(\frac{f}{kHz}-3{,}3\right)^2+10^{-3}}\left(\frac{f}{kHz}\right)^4 \quad [dB] \tag{6.5}$$

Normalizace prahu slyšitelnosti je provedena pro okolní funkce podle následujícího pravidla: Mějme signál o frekvenci 4kHz a určité amlitudě +-1LSB ležící na křivce absolutního prahu. Absolutní práh je k dispozici ve formě tabulky pro odlišné rychlosti vzorkování Fs. Tabulka dále obsahuje hodnoty pro všechny nezbytné frekvence k výpočtu maskovacího prahu. Tabulka je uvedena ve zdrojovém kódu v příloze. K tomu abychom získali práh slyšitelnosti, musíme být v tomto případě ještě vypočítán globální maskovací práh. Dodatečně je provedena opravná kompenzace, která je přidána v závislosti na celkové bitové rychlosti užita pro kanál. [4]

$$Offset = \begin{cases} -12dB, bit\ rate < 96Kbps \\ 0dB, bit\ rate \geq 96Kbps \end{cases} \tag{6.6}$$

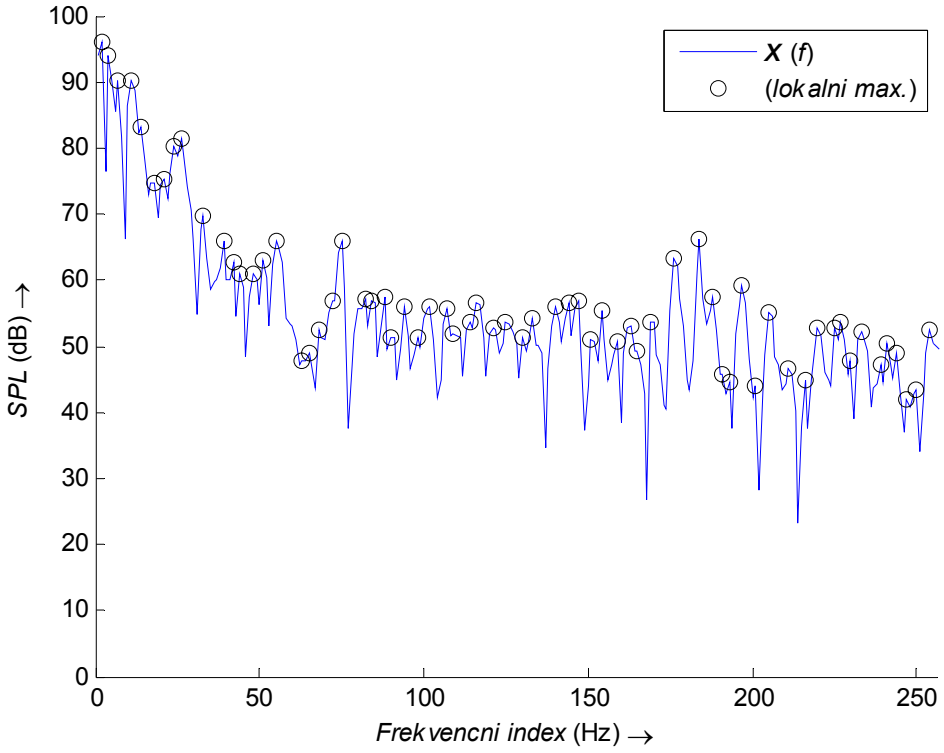
Jestliže je počítán maskovací práh ležící pod prahem slyšení, tak je maskovací práh nastaven na absolutní práh pro každé pásmo.

6.4 Určení tónových a netónových složek

Maskovací křivky jsou stanoveny pomocí tónové individuální masky. Proto rozlišujeme složky s různými charaktery, jako jsou tónové a netónové. Tónové a netónové složky jsou nezbytné k výpočtu globálního maskovacího prahu z frekvenčního spektra. [6]

První krok je určení lokálních maxim z výkonového spektra podle následujícího vzorce.

$$X(k-1) < X(k) \ , \ X(k) \geq X(k+1) \tag{6.7}$$



Obr.6.3 určení lokálních maxim.

Dalším krokem je určení tónových složek. Uvnitř každého kritického pásma je základní prozkoumání výkonového spektra nejbližší spektrální složky. Lokální maximum $X(k)$ je tedy tónová složka podle následujícího kriteria:

$$X(k) - X(k+j) \geq 7 \ [dB] \tag{6.8}$$

kde

$$\begin{array}{lll} j = & -2, +2 & pro \quad 2 < k < 63 \\ j = & -3, -2, -2, +3 & pro \quad 63 \leq k < 127 \\ j = & -6, \dots, -2, +2, \dots, +6 & pro \quad 127 \leq k < 250 \end{array} \tag{6.9}$$

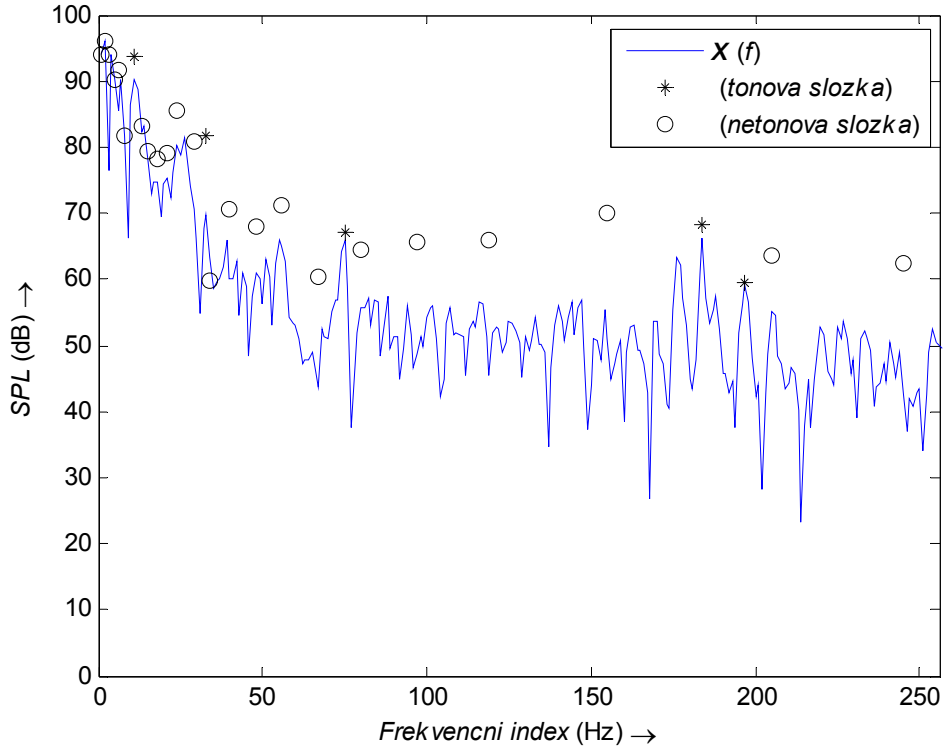
Úroveň akustického tlaku tónové masky lze vypočítat:

$$X_{tm}(k) = 10 \log_{10} \left(10^{\frac{X(k-1)}{10}} + 10^{\frac{X(k)}{10}} 10^{\frac{X(k+1)}{10}} \right) [dB] \tag{6.10}$$

Netónové složky jsou počítány ze zbylých složek kromě tónových uvnitř každého kritického pásma. Výkon těchto spektrálních složek je zahrnut do netónových složek $X_{nm}(k)$ korenspondujících ke kritickému pásmu. Index k netónových složek je definován jako index spektrální čáry nejbližší mezi dvěma okraji kritického pásma. Obr.2.3 ukazuje výkonové spektrum s tónovými a netónovými složkami nalezené v bloku o délce $N = 512$ vzorků s vzorkovací frekvenci $F_s = 44100\ kHz$.

Šířka frekvenčního pásma z kritického pásma se mění kolem střední frekvence s šířkou pásma pouze o $0{,}1kHz$ pro nízké frekvence a s šířkou pásma přibližně o $4kHz$ pro vyšší frekvence. K určení jestli lokální maximum může být tónového charakteru ve frekvenčním rozsahu df okolo lokálního maxima bylo už prozkoumáno a frekvenční rozsah df je definován:

$$\begin{array}{ll} df = 172.266Hz & 0kHz < f \leq 5{,}512kHz \\ df = 281{,}25Hz & 5{,}512kHz < f \leq 11{,}024kHz \\ df = 562{,}50Hz & 11{,}024kHz < f \leq 19.982kHz \end{array} \tag{6.11}$$



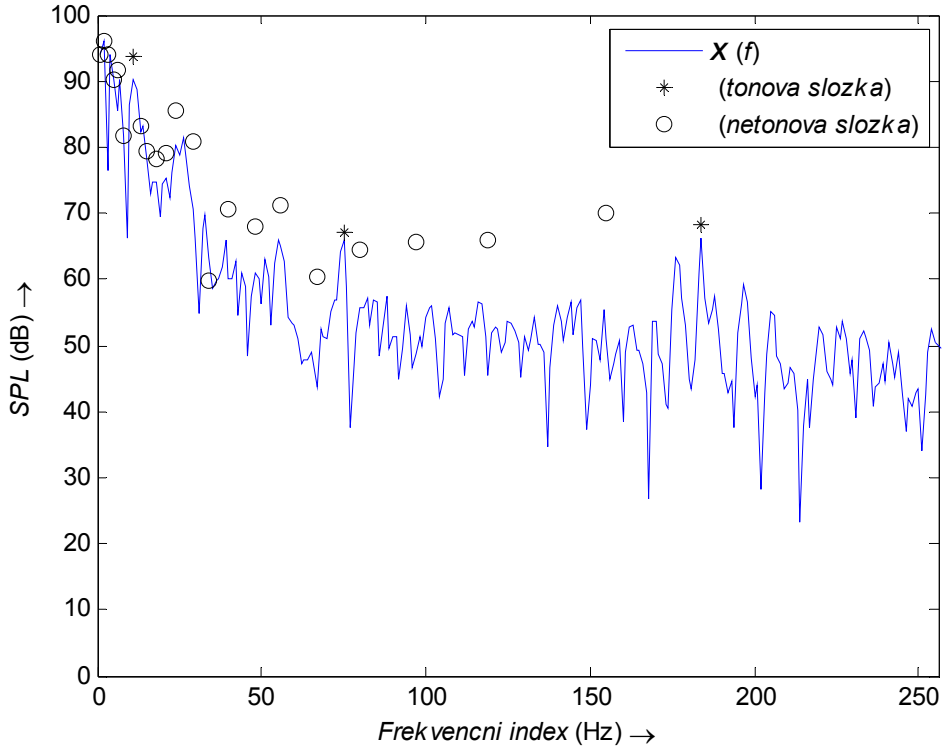
Obr.6.4 Určení tónových a netónových složek.

6.5 Podvzorkování tónových a netónových složek

Decimaci tónových netónových složek provádíme kvůli vyšší rychlosti algoritmu. Počet masek je požadován pro výpočet globálního maskovacího prahu. Tento počet je snížen v tomto kroku. Složky jsou odstraněny ze seznamu významných složek, jestliže jejich energie je menší než absolutní maskovací práh.

$$X_{tm}(k) \geq LTq(k) \quad nebo \quad X_{nm}(k) \geq LTq(k) \tag{6.12}$$

Pro tónové složky se užívá ještě dodatečná decimace. Ta je provedena jestliže dvě nebo více složek jsou odděleny méně jak $0{,}5\ Bark$. Tónová složka s vyšším výkonem je ponechána, zatímco všechny ostatní složky jsou odstraněny ze seznamu tónových složek. Tato operace je výkonnější použitím posuvného okna o šířce 0,5 Bark v každém kritickém pásmu. Zbývající tónové a netónové složky jsou užity k výpočtu individuálního maskovacího prahu. [6]



Obr.6.5 určení decimovaných tónových a netónových složek

6.6 Výpočet individuálního maskovacího prahu $LTtm$ a $LTnm$

V MPEG modelu je užívána pouze podmnožina o $N/2$ spektrálních čar k výpočtu globálního maskovacího prahu. To vede k redukci podvzorkování ve frekvenční oblasti jako nelineární mapování o $N/2$ frekvenčních složek. Číslo vzorků je užíváno k podvzorkování frekvenční oblasti v závislosti na vzorkovací rychlosti a vrstvě. Pro Layer I, jsou počet vzorků [2]:

$$\begin{aligned} f_s &= 32kHz \quad n = 108 \\ f_s &= 44{,}1kHz \quad n = 106 \\ f_s &= 48kHz \quad n = 102 \end{aligned} \tag{6.13}$$

Maskovací práh pro tónové a netónové masky je počítán takto:

$$\begin{aligned} LT_{tm}[z(j), z(i)] &= X_{tm}[z(j)] + av_{tm}[z(j)] + vf[z(j), z(i)]\ [dB] \\ LT_{nm}[z(j), z(i)] &= X_{nm}[z(j)] + av_{nm}[z(j)] + vf[z(j), z(i)]\ [dB] \end{aligned} \tag{6.14}$$

Tabulka 6.1 Barkova stupnice pro MPEG

n	index	frekvence [Hz]	Bark [z]
0	1	86.133	.850
1	2	172.266	1.694
2	3	258.398	2.525
3	5	430.664	4.124
4	6	516.797	4.882
5	8	689.063	6.301
6	9	775.195	6.959
7	11	947.461	8.169
8	13	1119.727	9.244
9	15	1291.992	10.195
10	17	1464.258	11.037
11	20	1722.656	12.125
12	23	1981.055	13.042
13	27	2325.586	14.062
14	32	2756.250	15.100
15	37	3186.914	15.955
16	45	3875.977	17.079
17	50	4478.906	17.904
18	55	5340.234	18.922
19	61	6373.828	19.963
20	68	7579.688	20.971
21	75	9302.344	22.074
22	81	11369.531	22.984
23	93	15503.906	24.013
24	106	19982.813	24.573

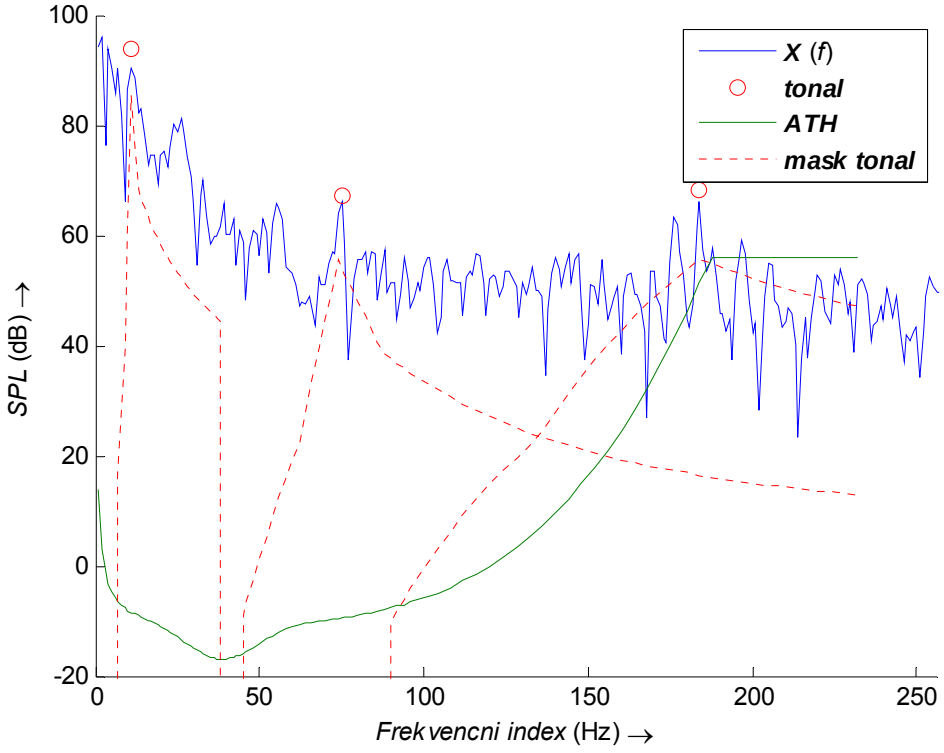
Maskovací práh je počítán z frekvenčního indexu i, do té doby dokud j je frekvenční index masky a $X_{tm}[z(j)]$ je výkonové spektrum masky s indexem j. Člen av je tzv. critical band rate a označen jednotkách $Bark$. Hodnota frekvence je v Bark a korespondující frekvenční index jsou uloženy v Tab. 6.1. Obr.2.5 a obr 2.6 ukazuje decimované tónové masky a příslušné individuální maskovací prahy.

Maskovací index pro tónové a netónové masky je počítán takto:

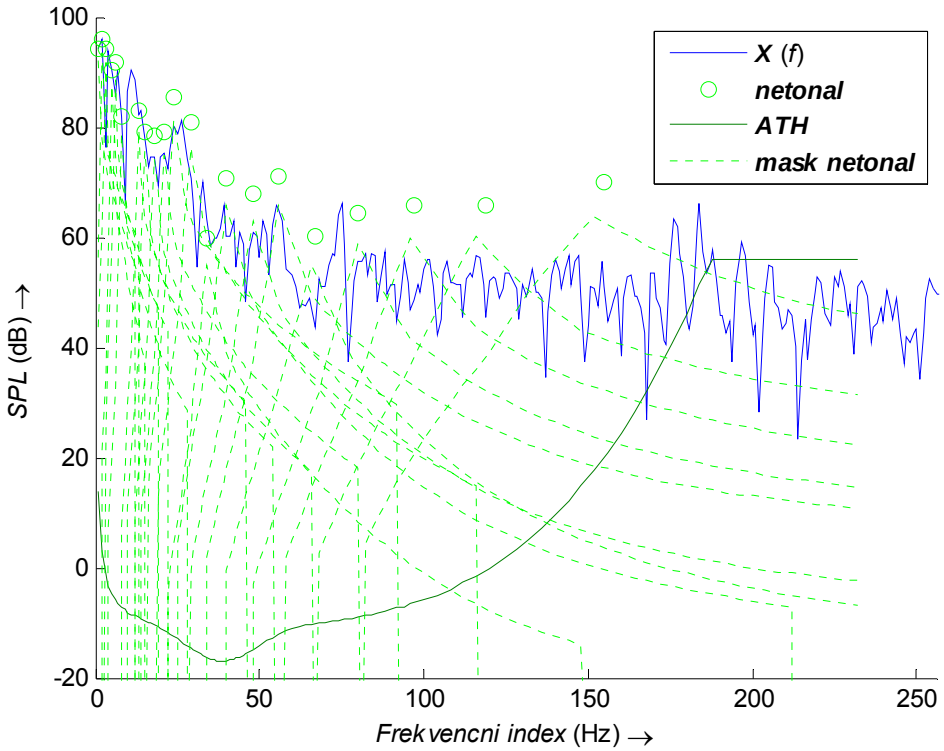
$$av_{tm}[z(j)] = -1{,}525 - 0{,}275 * z(j) - 4{,}5 \ [dB]$$
$$av_{nm}[z(j)] = -1{,}525 - 0{,}175 * z(j) - 0{,}5 \ [dB] \quad (6.15)$$

Maskující funkce $vf[z(j), z(i)]$ s rozestupem v Barcích $\Delta z = z(i) - z(j)$ je definována takto:

$$vf = \begin{cases} 17(\Delta z + 1) - (0{,}4X[z(j)] + 6) & -3 \le \Delta z < -1 \\ (0{,}4X[z(j)] + 6) * \Delta z & -1 \le \Delta z < 0 \\ -17\Delta z & 0 \le \Delta z < 1 \\ -(\Delta z - 1) * (17 - 0{,}15X[z(j)]) - 17 & 1 \le \Delta z < 8 \\ v\ decibelech & v\ barcích \end{cases} \tag{6.16}$$



Obr.6.6 Zobrazení maskovacích křivek pro tónové složky



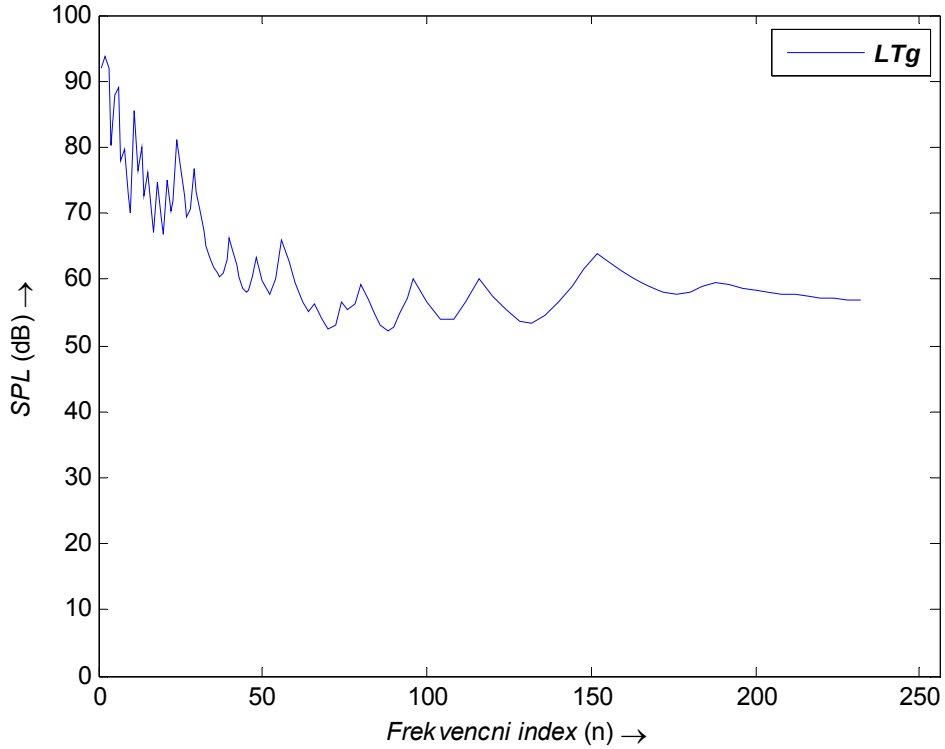
Obr. 6.7 Zobrazení maskovacích křivek pro netónové složky.

6.7 Výpočet globálního maskovacího prahu LTg

K tomu abychom vypočítali globální maskovací práh LTg, tak rozdílné složky musí být navýšeny. Globální maskovací práh pro frekvenční index i jsou vypočítány navýšené výkony pro práh v tichu pro tónové a netónové masky podle obr.6.8

$$LT_g(i)10\log_{10}\left(10^{\frac{LT_q(i)}{10}}+\sum_{j=1}^{N_t}10^{\frac{LT_{tm}[z(i),z(j)]}{10}}\sum_{j=1}^{N_n}10^{\frac{LT_{nm}[z(i),z(j)]}{10}}\right) \tag{6.17}$$

N_t a N_n udávají číslo tónových a netónových složek.



Obr.6.8 určení globálního maskovacího prahu.

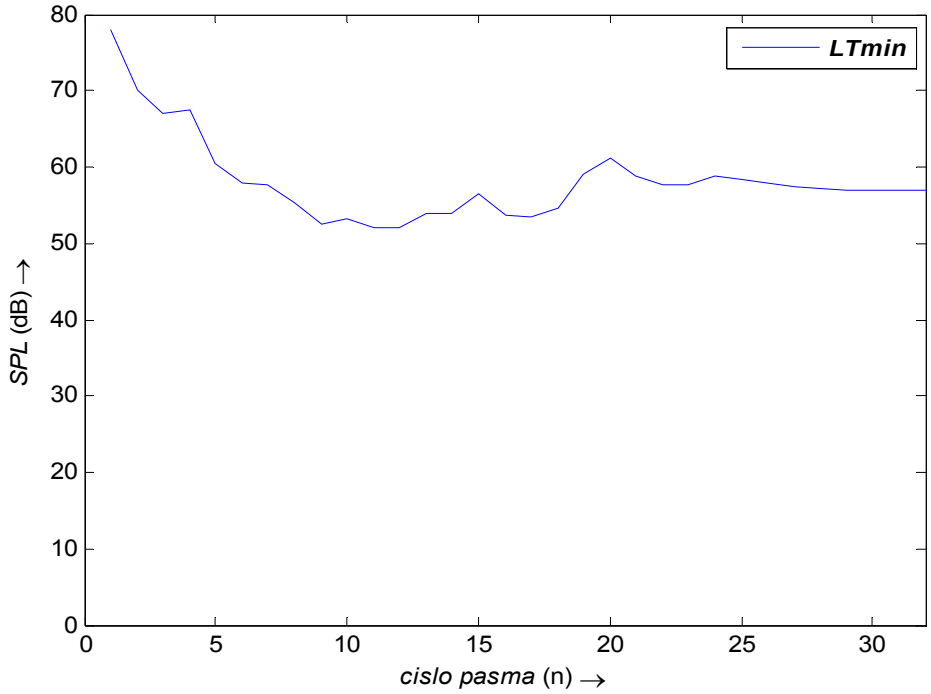
6.8 Výpočet minimálního maskovacího prahu LTmin

Globální maskovací práh LTg je počítán z podvzorkované frekvenční oblasti s číslem spektrální čáry podle rovnice (6.13). Tyto frekvenční indexy jsou mapovány do 32 subpásem. Minimální maskovací práh (viz. obr.6.9) vypočteme podle:

$$LT_{Min}(n) = \min_{f(i)\ \epsilon\ subpásmo\ n} LT_q(i)\ [dB] \tag{6.18}$$

Užitím psychoakustického modelu je zvýšena robustnost vodoznakové sekvence a zároveň můžeme dostatečně zvýšit intenzitu vkládané vodoznakové zprávy až na takovou hodnotu, kdy vodoznaková zpráva nepřesáhne hranici minimálního maskovacího prahu. Takto upravené spektrum vodoznakové zprávy je přičteno ke spektru originálního zvukového signálu podle vztahu:

$$OUT(jw) = X(jw) + Wl(jw) \tag{6.19}$$

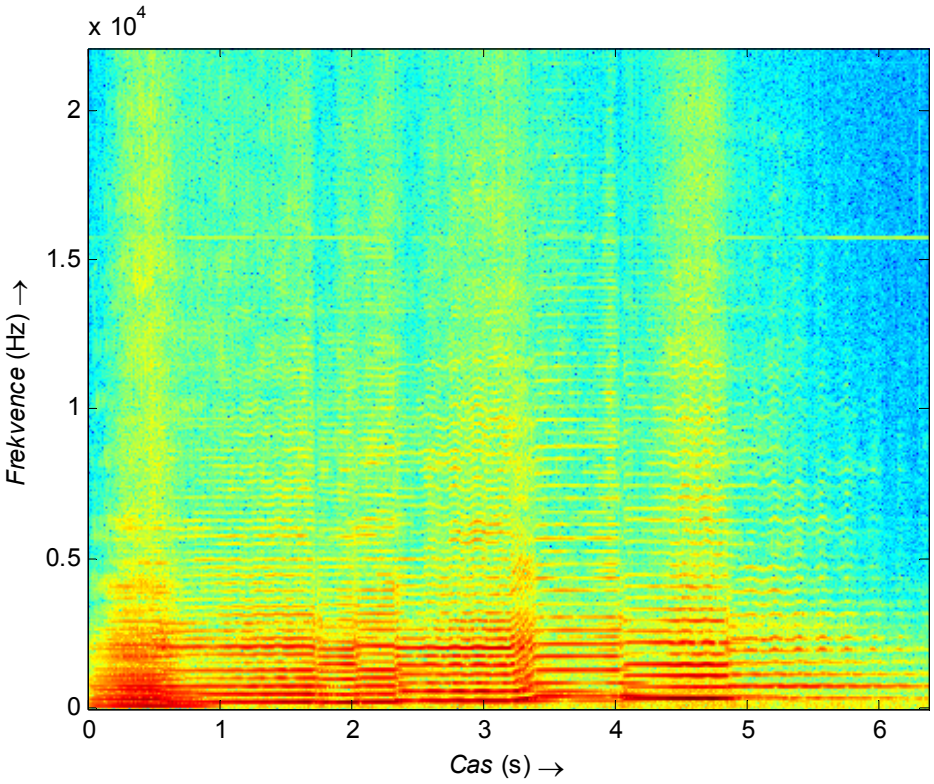


Obr.6.9 určení minimálního maskovacího prahu.

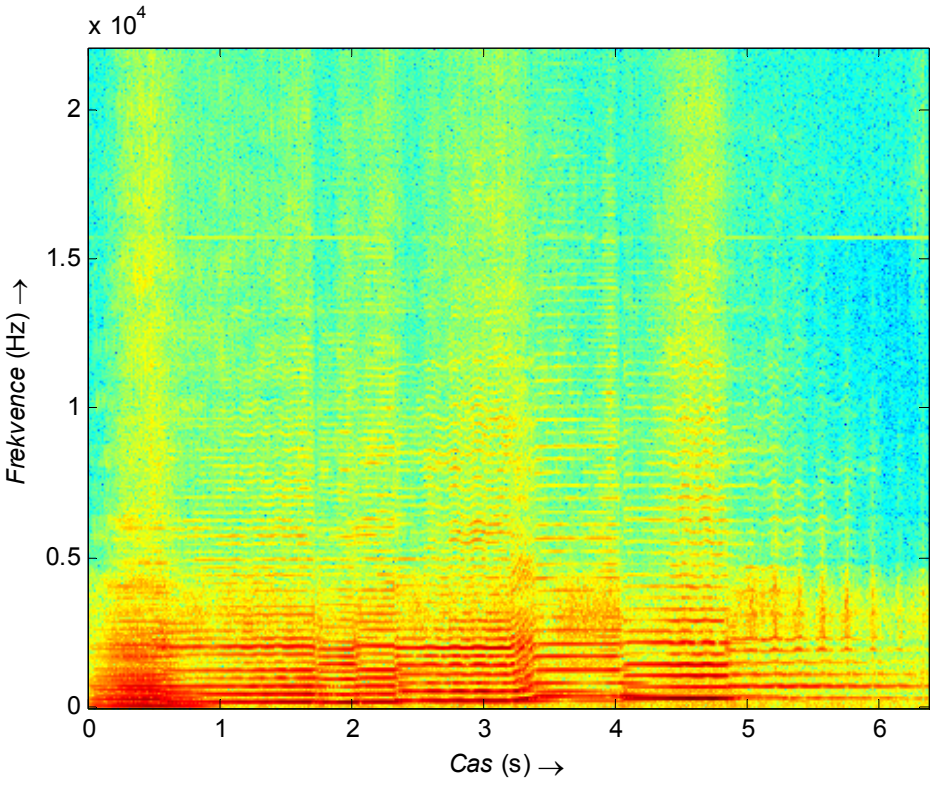
Transformaci z frekvenční oblasti do časové oblasti užitím inverzní Fourierovi transformace (IFFT) je získán segment vodoznačeného zvukového signálu.

$$out(t) = IFFT\{OUT(jw)\} \tag{6.20}$$

Z těchto segmentů je poskládán vodoznačený zvukový signál, který by měl být téměř totožný s originálním zvukovým signálem. Na obrázcích 6.10 a 6.11 k jsou k porovnání spektrogramy originální zvukové nahrávky s vodoznačenou.



Obr.6.10 spektrogram audio signálu bez vloženého vodoznaku



Obr.6.10 spektrogram audio signálu s vloženým vodoznakem

7 Extrakce vodoznaku

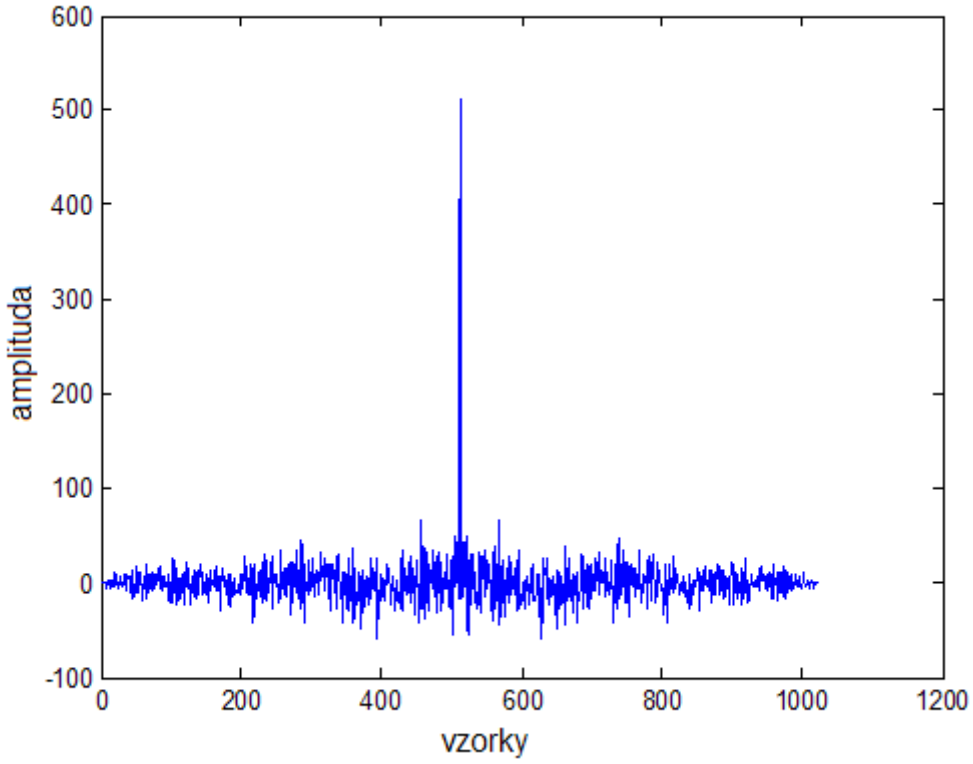
Pro extrakci vodoznaku potřebujeme vodoznačený a originální audio signál. Tyto signály převedeme do frekvenční oblasti pomocí rychlé Fourierovy transformace (FFT) a odečteme spektrum originálního zvukového signálu od vodoznačeného zvukového signálu. Rozdílem těchto spekter jsme získali spektrum, které bylo užito pro vkládání vodoznaku. Takto získaná data jsou zpětně transformována do časové oblasti užitím IFFT:

$$r(r) = IFFT\{R(j\omega)\} \tag{7.1}$$

7.1 Synchronizace systému rozprostřeného spektra

Pseudonáhodná sekvence je užívána ve vysílači k tomu, aby přizpůsobila signál. Požadavkem je to aby i přijímač měl k dispozici kopii pn sekvence. Kopie PN sekvence je potřebná k dekódování přijatého signálu. To je uděláno násobením vstupního signálu a lokální kopií pn sekvence. Pro správné dekódování přijatého signálu, musí být kopie PN sekvence synchronizovaná se vstupním signálem.. Proces synchronizace je obvykle vykonán ve dvou krocích: první, je přesné zarovnání PN sekvence. K porovnání vstupního signálu s PN sekvencí využíváme korelační funkci. [9] Aby byla provedena správná detekce a dekódování vodoznakového signálu, je nezbytné znát parametry užité ke generaci vodoznakového signálu, jako je modulační signál, rozměry prokládací matice, pn-sekvence, velikost vodoznaku, velikost hlavičky atd.

Typický výstup korelace nám dává špičku, která je použita k synchronizaci, jak je ukázáno na obr.7.1 kdy je použito dvou shodných pn sekvencí.



Obr.7.1 užití korelační funkce dvou pn sekvencí

Pokud však přijímaný signál je zkreslen aditivním šumem nebo nějakým jiným rušením, může se stát, že výsledek korelace nebude schopen správně detekovat špičku a to vede ke špatnému dekódování vodoznakové informace. Ke zvýšení detekční schopnosti se může v takovém případě využívat přizpůsobivého filtru

Korelace nebo přizpůsobivý filtr (matched filter) může být použit pro obnovení vodoznakové informace využívané u metod s rozšířeným spektrem. Použitím korelace nebo přizpůsobivého filtru může být použito pro znovuzískání modulovaných bitových hodnot. Pokud binární klíčování s fázovým posuvem (BPSK) modulovaného signálu je korelováno se známou pseudonáhodnou sekvencí, může nastat stav, kdy výsledkem není dostatečně velká špička, to může způsobit aditivní šum nebo nějaké jiné rušení způsobené přenosem. Výpočet korelace může být proveden v časové nebo ve frekvenční oblasti. V našem případě je vhodné použití ve frekvenční oblasti. Výsledek korelace je přesnější, protože frekvenční špičky, které nepřispívají k určení správného

výsledku detekce, jsou přizpůsobivým filtrem potlačeny a tím jsme schopni přesněji určit správnou špičku, která určí správný výsledek detekce. [10]

Vodoznačené signály vytvořené pomocí metody rozšířeného spektra jsou detekovány pomocí korelačních vztahů nebo přizpůsobivého filtru. Použitím takové metody se zvyšuje spolehlivost dekódování vodoznaku a zároveň vytváří vodoznak více robustní. Korelace může být provedena použitím jednorozměrné realné symetrické FFT

$$R = real\big(FFT(r)\big) \tag{7.2}$$

$$C = real\big(FFT(c)\big) \tag{7.3}$$

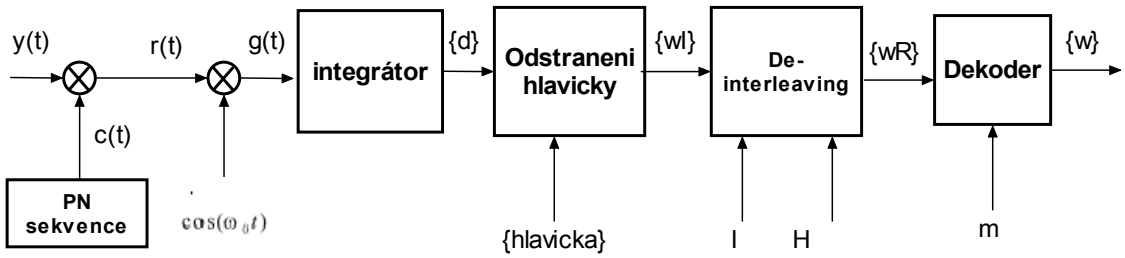
Ve kterém segment R představuje první transformovaný vektor a pseudonáhodná sekvence C představuje druhý transformovaný vektor mající N hodnot. Jeden z transformovaných vektorů je násoben konjugovanou maticí transformovaného vektoru k tomu, aby vytvořil korelaci v kmitočtové oblasti, a užitím inverzní IFFT získáme přesnější detekci.

$$DET = R * conj(PN) \tag{7.4}$$

$$\det = real\big(IFFT(DET)\big) \tag{7.5}$$

7.2 Dekódování vodoznaku

Po detekci špiček najdeme jejich příslušnou pozici ve vodoznakové sekvenci. Pro každý rámec y(t) se stejnou délkou jako je vodoznakový signál je proveden proces, který je zobrazen na obr.7.2.



Obr.7.2 Blokové schéma dekódovacího procesu

Přijatý signál je násoben PN sekvencí užitím rovnice:

$$r(t) = c(t)y(t) \tag{7.6}$$

Demodulace BPSK signálu je provedena:

$$g(t) = r(t)cos(2\pi f_0) \tag{7.7}$$

K odhadu bitové posloupnosti použijeme následující vztah:

$$r_i = \int_{(i-1)T_s}^{iTs} g(t)dt \tag{7.8}$$

$$i = 1,2 \dots k$$

Na konec se získá bitová sekvence $\{d\}$:

$$d_i = \begin{cases} 1, & pro\ r_i > 0 \\ -1, & pro\ r_i \leq 0 \end{cases} \tag{7.9}$$

Po získání bitové sekvence $\{d_i\}$ odstraníme hlavičku. Tímto způsobem získáme bitovou posloupnost $\{w_I\}$. Ze získané bitové posloupnosti provedeme de-interleaving to je provedeno užitím stejné matice, kterou jsme použili pro generaci vodoznaku viz.tab.5.1. Takto vypočtenou bitovou posloupnost w_R dekódujeme pomocí opakovací funkce m ke získání vodoznakové posloupnosti užitím následujícího vztahu:

$$w_k = \begin{cases} 1 & \sum_{i-1}^{m} w_{Ri} > 0 \\ -1 & \sum_{i-1}^{m} w_{Ri} \leq 0 \end{cases} \tag{7.10}$$

8 Test robustnosti a transparentnosti vodoznaku

Pro test robustnosti a transparentnosti jsem vybral tři zvukové nahrávky o délce 10s. Tyto nahrávky jsem ze tří hudebních stylů a to je pop, rock a klasika. Tyto nahrávky jsou formátu *.wav, vzorkovány frekvencí Fs=44100Hz a bitové rychlosti 128kbps. U těchto nahrávek jsem provedl změnu jejich obsahu užitím

metod pro běžné zpracování signálů, jako je filtrace, převzorkování, a převod do formátu MP3 a zpět.

Pro test transparentnosti jsem použil výpočtu SNR mezi originálním signálem a vodoznačeným:.

Pro:

- Rock $SNR = 22{,}3\ (dB)$
- Pop $SNR = \ 22{,}75\ (dB)$
- Klasika $SNR = 15.5\ (dB)$

Test robustnosti: Při testu robustnosti jsem počítal BER (Bit Error Ratio) extrahovaných bitů.

- **Filtrace:** Dolní propust používající Butterworthův filtr 4-řádu s mezním kmitočtem 8kHz.
- **Převzorkování:** Bylo provedeno změnou vzorkovací frekvence z původních 44100Hz na 22050Hz a poté zpět na 44100Hz.
- **Komprese:** Pro kompresi bylo použito formátu MP3 pro bitový tok 64kbps, 96kbps a 128kbps. Tyto komprimované soubory jsem poté převedl zpět do formátu *.wav pro vzorkovací frekvenci Fs=44100Hz a bit tok 128kbps.

METODA ZPRACOVÁNÍ	ROCK BER (%)	POP BER (%)	KLASIKA BER (%)
Vodoznačený signál (bez úprav)	0	0	0
Dolní propust	32,21	32,95	4,58
převzorkování	4,6	12	0
Komprese WAV-> MP3(64kbps)->WAV	40	43,3	22,61
Komprese WAV-> MP3(96kbps)->WAV	35,1	34,5	15,8
Komprese WAV-> MP3(128kbps)->WAV	34,5	33,9	16,11

Výpočetní náročnost: Je vypočtena jako průměrná doba, která byla potřebná pro vložení a extrakci vodoznaku.

Pro vkládání a dekódování vodoznaku byly použity audio nahrávky o délce $t = 10s$. Pro vkládání vodoznakové zprávy byla změřena doba $T_{vlož} = 54s$ a pro dekódování byla změřena doba $T_{dek} = 3{,}7\ s$

Proces vkládání a dekódování byl použit na PC jehož konfigurace je:

CPU	Intel core duo(1,86GHz)
Operační pamět	1GB
Operační systém	Microsoft windows XP Profesional

9 Závěr

Při zpracování akustických signálů jsem využíval vývojové prostředí Matlab 7.014. a jeho knihovny Signal processing toolbox. Tato knihovna obsahuje řadu funkcí pro zpracování a analýzu signálu. Ke zpracování akustických signálů jsem využíval moderní prostředky signálové analýzy, jako jsou FFT. Jako vstupní data jsem používal formát *.wav o vzorkovací frekvenci 44100Hz a bitové rychlosti 128kbps.

Pro digitální vodoznačení zvukových signálů jsem si vybral metodu rozprostřeného spektra. Jedná se o metodu, která je robustní a má dobré vlastnosti při potlačování rušení během přenosu. Je velice podobná symetrickému šifrování v kryptografii. Na zakódování a dekódování jsem použil vygenerovanou PN sekvenci, která představuje tajný symetrický klíč. Výstupem rozprostřeného spektra byl zakódovaný vodoznakový signál šumového charakteru, který jsem upravil podle psychoakustického modelu, který je používán v MPEG I layer I. Takto upravené spektrum vodoznakové zprávy jsem vkládal do spektra originálního zvukového signálu. Tímto jsem získal vodoznačený zvukový signál.

Pro detekci jsem použil tzv. přizpůsobivého filtru (matched filter), který snižuje vliv šumu na vstupní signál. kterým jsem prováděl korelaci ve frekvenční oblasti. ,Detekční výsledky byli mnohem lepší než použití korelace v časové oblasti.

Z vodoznačeného zvukového signálu jsem extrahoval vloženou vodoznakovou zprávu. Na závěr jsem provedl test robustnosti, tak že jsem vodoznačený signál modifikoval filtrací, převzorkováním a převodem do formátu MP3 a zpět.

Ve výsledném vodoznačeném signálu nebyla vložená informace téměř slyšet. Avšak pokud zvukový signál měl velmi malou energii nebo pokud v něm byly pauzy mohla být vodoznaková zpráva slyšitelná jako šum, Tento problém by mohl být odstraněn užitím časového maskování.

Použitá literatura

[1] Nedeljko Cvejic. Algorithms for audio watermarking and stenography; Oulu 2004; ISBN 951-42-7384-2 (PDF). <http://herkules.oulu.fi/isbn9514273842/isbn9514273842.pdf>

[2] Arnold M; Schmucker M; Wolthusen S; Techniques and applicatios of digital watermarking and content protection artech. House, inc, 2003. ISBN 1-58053-111-3

[3] Fastl, H; Zwicker, E. Psychoacoustic: Facts and models. 3rd edition; Springer 2006. ISBN 3540231595

[4] Juergen Seitz; by Idea Group Inc. Digital Watermarking for Digital Media. . ISBN 1-59140-519

[5] Prof. Ing. Vladimír Šebesta.CSc. Teorie sdělování, Druhé vydání, Vydavatel-VUT v Brne 2001. ISBN 80-214-1843-5.

[6] <http://neuron2.net/library/mpeg1>

[7]Rangding Wang; Peiqi Chai; A new adaptive audio watermarking algorithm for copyright protection.

[8] Yuval Cassuto; Michael Lustig; Shay Mizrachy. Real-Time DigitalWatermarking System for Audio Signals Using Perceptual Masking <http://www.stanford.edu/~mlustig/ti_challange.pdf>

[9] Yuksel Tokur; Ergun Ercelebi. Spread Spectrum audio watermarking scheme based on psychoacoustic model. <http://www.emo.org.tr/resimler/ekler/ecc353586042b6d_ek.pdf>

[10Method and apparatus for decoding watermark information items of a watermarked audio or video signal using correlation. European patent application.<https://publications.european-patent-office.org/PublicationServer/getpdf.jsp?cc=EP&pn=1798686&ki=A1>

[11] J. Mee. Spread Spectrum (SS)
<http://sss-mag.com/pdf/Ss_jme_denayer_intro_print.pdf>

[12]Karel Zaplatílek; Bohuslav Doňarů MATLAB – Začínáme se signály; BEN-Technická literatura;ISB 800-7300-200-0

Zkratky a symboly:

Av_{nm}	maskovací index pro netónové složky
Av_{tm}	maskovací index pro tónové složky
BPSK	dvoustavové fázové klíčování (Binary phase shift keying)
dB	decibel
df	frekvenční rozsah
DFT	diskrétní Fourierova transformace
FEC	dopředné potlačení chyb
FFT	rychlá Fourierova transformace
FHSS	frequency hoping spread-spectrum
Fs	vzorkovací kmitočet
h	hannovo okno
K	klíč
L	intenzita zvuku
LSB	least significant bit
L_{sb}	sound pressure level (dB)
LTg	globální maskovací práh
LTmin	minimální maskovací práh
LT_{nm}	maskovací křivky pro netónové složky
LTq	Threshold in quiet- práh ticha
LT_{tm}	maskovací křivky pro tónové složky
MPEG	Motion Picture Expert Group

N	délka FFT
N_n	číslo netónových složek
Nt	číslo tónových složek
Pa	Pascal
Pn	pn-sekvence
p_0	tlak (Pa)
PCM	pulse code modulation
PSD	power density spectrum
RMS	root mean square
Scf	scale factor
SMR	signal to mask ratio
SPL	sound pressure level (dB)
X	spektrum signálu
X_{nm}	spektrum netónových složek
X_{tm}	spektrum tónových složek
vf	maskovací funkce
z(j)	critical band rate (Bark)
π	Ludolfovo číslo 3,14
W	vodoznak

Příloha:

```
%=================================================
%========= VKLADANI VODOZNAKU=====================
%=================================================



clear all
close all
clc
warning off;
disp('=============================================');
disp('Program te vyzve k nacteni audio souboru    =');
disp('do ktereho chces vlozit vodoznak            =');
disp('=============================================');
disp('v dalsim kroce zadej nazev souboru          =');
disp('do ktereho chces vlozit vodoznak            =');
disp('=============================================');
disp('pote zadej svoji autorskou signaturu        =');
disp('kterou chces vlozit do audio signalu        =');
disp('=============================================');
disp('na zaver zadej uroven vodoznaku v (dB)      =');
disp('=============================================');
disp('pro pokracovani stiskni libovolnou klavesu  =');
disp('=============================================');
pause;
%zobrazeni dialogu pro nacteni souboru
[filename,pathname] = uigetfile( ...
{'*.wav', 'Zvukov0 soubory';...
'*.*', 'Vsechny soubory'},...
'Vyber soubor')
%nacteni souboru
[data,FS,BITS]=wavread(fullfile(pathname, filename));
[Nx,channels] = size(data);
%uprava vstupnich data
data=data(1:length(data)/1)';
data=data./max(data);
wat_audio = input('zadej nazev pro ulozeni vodoznaceneho audio signalu
:','s');
help_file = input('zadej jmeno pro vytovreni souboru potrebneho pro
detekci :','s');
wstring = input('zadej 6 znaku pro vodoznaceni :','s');
level = input('zadej uroven znaku v dB  : -','s');
level = str2num(level);
level = abs(level);
clc;
disp('');
disp('CEKEJ...');
tic;    %stopky
%*****************prevod vodoznaku na posloupnost 1,-1
*****************
temp=strcat(wstring,'########');
watermark(1:6)=temp(1:6);
q=dec2bin(watermark,8);
for K = 1:size(q,1),
    for L = 1:8,
        w_bin(L+(K-1)*8) = str2num(q(K,L))*2-1; %
```

```
    end
end
% deklarace
BLOCK=512;          %delka bloku
LENGTH = length(data);      %delka signalu
OVERLAP = 75 ;            % prekryti v procentech
OVERLAP=round(OVERLAP * BLOCK/100); %)prekryti
N_FRAMES = round ( (LENGTH - OVERLAP) / (BLOCK - OVERLAP) );
N_SUBBAND=32;

% metoda rozprostreneho spektra
m = 5;      %zopakovani vodoznaku pro vyssi robustnost
I = 9; H = 27;  % velikost prokladaci matice
matrix=zeros(H,I);
header = ones(1,120); %generace hlavicky

Rs=500; % bitrate pro BPSK
Td=1/Rs;
Td_samples=FS/Rs; %cas ve vzorcich pro kazdy data bit
Rc=1500; % chip rate pseudonadne sekvence pnt (nasobek Rs)
Tc=1/Rc;
Tc_samples=FS/Rc; %data bit PN sekvence ve vzorcich
w_length=length(w_bin); %delka bitove posloupnosti vodoznaku

%pouziti opakovaciho kodu
wR=[zeros(1,w_length*m) ones(1,I*H-w_length*m)];
for ii=1:length(w_bin),
    wR(m*(ii-1)+1:m*ii)=w_bin(ii); %kazdy bit vodoznakove posloupnosti
je m-krat zopakovan
end

%zapis do prokladaci matice (interleaving)
for ii=1:I,
    matrix(1:H,ii)=wR(H*(ii-1)+1:H*ii)';
end

%zapis vodoznakove zpravy  do radkoveho vektoru
wI=zeros(1,H*I);
for ii=1:H,
    wI(I*(ii-1)+1:I*ii)=matrix(ii,1:I);
end
d = [ header wI]; %pridani hlavicky k vodoznakove sekvence
d_length=length(d);
F0 = 3500; %frekvence BPSK modulace
load -ascii pn_sekvence; %nacteni pn sekvence
pn=pn_sekvence';
pn_length = length(pn);
dt_length=floor(Td_samples*d_length); %delka casove oblasti bitove
posloupnosti
dt = d( ceil([1:dt_length]/Td_samples) ); %bitova poslounost v case
st = dt.*cos(2*pi*[1:dt_length]*F0/FS); %modulace BPSK
pn_t = pn( ceil(
(mod([1:dt_length],pn_length*Tc_samples)+0.1)/Tc_samples) );%pn
sekvence v case
xt=st.*pn_t; %rozprostreny signal
xt_length=length(xt);
 %pocet ramcu rozprostreneho signalu
xt_FRAMES=ceil( (xt_length-OVERLAP) / (BLOCK-OVERLAP));
for kk=1:xt_FRAMES, %segmentace ramcu
```

```
    if kk==xt_FRAMES    % testovani na posledni blok
        pos1=round((kk-1)*(BLOCK-OVERLAP))+1; %zacatek ramce
        pos2=xt_length;                       %konec ramce
        frame=[xt([pos1:pos2]) zeros(1,BLOCK-(pos2-pos1)-1)]; %zbytek
doplnen nulami
     else
        pos1=round((kk-1)*(BLOCK-OVERLAP))+1;
        pos2=round(kk*(BLOCK-OVERLAP)+OVERLAP);
        frame=xt([pos1:pos2]);
    end

    %prevod do frekvencni oblasti
    w=sqrt(8/3)*hamming(BLOCK)';
    w_pn=frame.*w;
    XT(kk,:)=fft(w_pn);
    XT_angle(kk,:)=angle(XT(kk,:))';
    XT(kk,:) = 20 * log10((fft(w_pn,BLOCK)));% / BLOCK);
    XT(kk,:) = XT(kk,:) +  96;      %uprava podle sound pressure level
end;


output = zeros(1,LENGTH);       %delka vodoznaceneho signalu
%nacteni tabulky absolutniho prahu pro uzite v psychoakustickem modelu
[TH, MAP,LTq] = Table_absolute_threshold(FS,BLOCK, 128);

%Segmentace ramcu, a uprava vodoznaku podle psyhoakustickeho modelu
for kk=1:N_FRAMES,
    if kk==N_FRAMES   %testovani posledniho ramce
        pos1=round( (kk-1) * (BLOCK - OVERLAP) )+1;
        pos2=LENGTH;
        s=[data([pos1:pos2]) zeros(1,BLOCK-(pos2-pos1)-1)]; %zbytek
ramce doplnen nulami
    else
        pos1=round((kk-1)*(BLOCK-OVERLAP))+1;
        pos2=round(kk*(BLOCK-OVERLAP)+OVERLAP);
     s=data([pos1:pos2]);
    end

    %************vypocet minimalniho maskovaciho prahu
     %vypocet spektra signalu
    [S,X,X_angle,Delta]=spektrum(s);
    %vypocet tonovych a netonovych slozek
    [tonal,nontonal,Xnm,Xtm] = Tonal_nontonal(X,TH,MAP);
    %Decimace tonovych a netonovych slozek

[D_tonal,D_nontonal,D_Xtm,D_Xnm]=decimace(Xtm,Xnm,tonal,nontonal,TH,LT
q,MAP);
    %Vypocet individualniho maskovaciho prahu
    [LTtm, LTnm] = vypocet_individual_mask_prah(X, D_tonal,
D_nontonal,D_Xtm,D_Xnm, TH, MAP);
     %vypocet LSB
    [Lsb,N_sb]=soundpresure(X,BLOCK);
     %vypocet globalniho maskovaciho prahu
    [LTg] = global_mask_prah(LTq,D_tonal,D_nontonal,LTtm,LTnm);
     %vypocet minimalniho maskovaciho prahu
    [LTmin]=Min_mask_prah(X,N_SUBBAND,LTg,MAP);

   %uprava vodoznaku podle minimalniho maskovaciho prahu
    xframe=mod(kk,xt_FRAMES);
    if xframe==0
```

```
       xframe=xt_FRAMES;
    end

    X_water=XT(xframe,1:BLOCK/2);                %polovina spektra
    X_water_angle = XT_angle(xframe,1:BLOCK);    %faze
    %pouziti psychoakustickeho modelu
     for ii=1:N_SUBBAND
       X_water(8*ii-7:8*ii)=(X_water(8*ii-7:8*ii))+mean(LTmin(ii)');
     end

     X_water=X_water-100-level;
     % zrcadleni spektra
     X_water=[X_water(1:BLOCK/2) conj(fliplr(X_water(1:BLOCK/2)))];
     X_water=X_water-96;
     X_water=(10.^(X_water/20));
     WATER = abs(X_water).*exp(j*X_water_angle);

    X=X-Delta;
    X=(10.^(X/20));
    X=X*BLOCK;
    SIGNAL=abs(X).*exp(j*X_angle);

    %pridani vodoznakove sekvence k nosnemu signalu ve spektralni
oblasti
    OUT=WATER+SIGNAL;
    out=real(ifft(OUT));    %prevedeni do casove oblasti
    output([pos1:pos2])=output([pos1:pos2])+out(1:(pos2-pos1)+1);
 end
toc;        %stopky
output=output./max(output);     %Korekce signalu
sound(output,FS);
wavwrite(output,FS,strcat(wat_audio));

clc;
disp('-------------------------------------------------------------');
disp(('Vodoznak byl vlozen do audio signalu'));
disp(sprintf('doba vkladani byla: = %f sec',toc));
disp('-------------------------------------------------------------');
%ulozeni potrebnych informaci, ktere mohou byt vyuzite pro detekci
save(strcat(help_file),'pn','output','header','w_length','d_length','w
atermark','w');
warning on
return;


%=====================================================
%========= EXTRAKCE VODOZNAKU=========================
%=====================================================


clear all
close all
clc
warning off;
disp('=============================================');
disp('Program te vyzve k nacteni originalniho     =');
disp('audio souboru                               =');
disp('=============================================');
disp('v dalsim kroce vyber audio soubor           =');
disp('ze ktereho chces extrahovat vodoznak        =');
```

```
disp('=============================================');
disp('pote       vyber soubor ktery obsahuje      =');
disp('doplnujici informace pro extrakci           =');
disp('=============================================');
disp('pro pokracovani stiskni libovolnou klavesu  =');
disp('=============================================');
pause;
%zobrazeni dialogu pro nacteni souboru
[filename,pathname] = uigetfile( ...
{'*.wav', 'Zvukov0 soubory ';...
 '*.*', 'Vsechny soubory'},...
 'Vyber originalni nahravku')
%nacteni originalniho souboru
[origdata,FS,BITS]=wavread(fullfile(pathname, filename));
[Nx_o,channels_o] = size(origdata);
origdata=origdata(1:length(origdata)/1)';
origdata=origdata./max(origdata);

[filename,pathname] = uigetfile( ...
{'*.wav', 'Zvukov0 soubory ';...
 '*.*', 'Vsechny soubory'},...
 'Vyber vodoznacenou nahravku')
%nacteni vodoznakoveho souboru
[waterdata,FS,BITS]=wavread(fullfile(pathname, filename));
[Nx_w,channels_w] = size(waterdata);
waterdata=waterdata(1:length(waterdata)/1)';
waterdata=waterdata./max(waterdata);

help_wat=input('zadej nazev pro nacteni pomocneho souboru =','s');
load(help_wat);    %nacteni pomocnych dat
clc;
disp('CEKEJ...');
tic;                % stopky
BLOCK=512;         %delka bloku
LENGTH = length(origdata);      %delka signalu
OVERLAP = 75 ;            %v procentech
OVERLAP=round(OVERLAP * BLOCK/100); %)prekryti
N_FRAMES = round ( (LENGTH - OVERLAP) / (BLOCK - OVERLAP) );  %pocet
ramcu
N_SUBBAND=32;       %pocet sub-pasem pro MPEG-I uzito pro vypocet
LTmin

d_length_original=d_length; % delka vodoznaku uzita pri vkladani
out_water = zeros(1,LENGTH); % urceni delky extrahovane vodoznakove
zpravy
%Segmentace ramcu
for kk=1:N_FRAMES
    if kk==N_FRAMES         %testovani na posl. blok
        pos1=round( (kk-1) * (BLOCK - OVERLAP) )+1;
        pos2=LENGTH;
         s_o=[origdata([pos1:pos2]) zeros(1,BLOCK-(pos2-pos1)-1)];
         s_w=[waterdata([pos1:pos2]) zeros(1,BLOCK-(pos2-pos1)-1)];
    else
         pos1=round((kk-1)*(BLOCK-OVERLAP))+1;
         pos2=round(kk*(BLOCK-OVERLAP)+OVERLAP);
         s_o=origdata([pos1:pos2]);
        s_w=waterdata([pos1:pos2]);
    end
    %vypocet spektra vodoznakoveho signalu odectenim orig. od
vodoznac.
    X_w = (fft(s_w));
```

```
    X_o = (fft(s_o));
     X_w_angle = angle(X_w);
     X_o_angle = angle(X_o);
     X_w = abs(X_w).*exp(j*X_w_angle);
     X_o = abs(X_o).*exp(j*X_o_angle);
    X_delta = X_w - X_o;
    delta=real(ifft(X_delta)); % prevod do casove oblasti
    out_water([pos1:pos2])=out_water([pos1:pos2])+delta(1:(pos2-
pos1)+1);  %poskladani ramcu
end


% Spread Spectrum vytvorene v prijmaci (detektoru)
%uzitim stejne pn-sekvence jako ve vysilaci to je porovnane s
prijmutym signalem
m = 5; %hodnota opakovaciho kodu pro dekodovani
I = 9;H = 27; %rozmery prokladaci matice
Rs=500;%bitrate pro BPSK
Td=1/Rs;
Td_samples=FS/Rs; %as ve vzorcich pro kazdy data bit
Rc=1500;          %chip rate pseudonadne sekvence pnt (nasobek Rs)
Tc=1/Rc;
Tc_samples=FS/Rc; % data bity PN sekvence ve vzorcich
load -ascii pn_sekvence; %nacteni pn sekvence
pn=pn_sekvence';
pn_length = length(pn);
d = header; %hlavicka kterou jsme pouzili pri vkladani
d_length=length(d);

%rozprostreni a modulace bpsk and modulation
F0 = 3500; %BPSK modulator center frequency
dt_length=floor(Td_samples*d_length); %delka casove oblasti bitove
posloupnosti
dt = d( ceil([1:dt_length]/Td_samples) ); %bitova poslounost v case
st = dt.*cos(2*pi*[1:dt_length]*F0/FS); % modulace BPSK
pn_t = pn( ceil(
(mod([1:dt_length],pn_length*Tc_samples)+0.1)/Tc_samples) ); %pn
sekvence v case
spread_temp=st.*pn_t; %rozprostreny signal hlavicky
length_spread=length(spread_temp);
spread_temp=[spread_temp zeros(1,length_spread)];
BLOCK=length(spread_temp); %
OVERLAP=BLOCK*0.25; %50 % overlap
FRAMES=ceil( (LENGTH-OVERLAP) / (BLOCK-OVERLAP));
POSITION=[];
counter=1;
R_WEST=[];
for kk=1:FRAMES,


    if kk==FRAMES
        pos1=round((kk-1)*(BLOCK-OVERLAP))+1;
        pos2=LENGTH;
        spread=[out_water([pos1:pos2]) zeros(1,BLOCK-(pos2-pos1)-1)];
    else
        pos1=round((kk-1)*(BLOCK-OVERLAP))+1;
        pos2=round(kk*(BLOCK-OVERLAP)+OVERLAP);
        spread=out_water([pos1:pos2]);
    end
```

```
    %pouziti matched filtru pro detekci spicky
    SPREAD=fft(spread);         %vypocet spektra z prijate vodoznakove
sekvence
    SPREAD_angle = angle(SPREAD);
    SPREAD = abs(SPREAD).*exp(j*SPREAD_angle);
    S_test=fft(spread_temp);     %spektrum pro testovani (detekci)
    test_angle = angle(S_test);
    S_test = abs(S_test).*exp(j*test_angle);
    DET= SPREAD.*conj(S_test);
    det=real(ifft(DET));
    [peak_value,peak_pos]=max(det);             %ulozime hodnotu a
pozici maximalni detekovane spicky
     if peak_pos<BLOCK*0.75
        POSITION(counter)=pos1+peak_pos-1;      %ulozime pozici spicky
        counter=counter+1;
     end
end

clc;
% watermark de-spreading
 dt_length=floor(Td_samples*d_length_original); %velikost vodoznaku ve
vzorcich
 pn_t = pn( ceil(
(mod([1:dt_length],pn_length*Tc_samples)+0.1)/Tc_samples) );%pn
sekvence v case

for kk=1:length(POSITION),
    if POSITION(kk)+dt_length-1 <= LENGTH
        yt=out_water([POSITION(kk):POSITION(kk)+dt_length-1]);


        rt= yt.*pn_t;%
        gt=rt.*cos(2*pi*[1:dt_length]*F0/FS);%##BPSK de-modulaCE

        %urceni delky vodoznaku
        ri=zeros(1,d_length_original);
        for ii=1:d_length_original,
            ri(ii)=sum(gt(floor(Td_samples*(ii-
1))+1:floor(Td_samples*ii)));
        end

        %
        d_est=sign(ri);
        d_est(find(d_est==0))=-1;
        %odstraneni hlavicky
        wI_est=d_est(length(header)+1:d_length_original);

        matrix=zeros(H,I); %vytvoreni prokladaci matice
        for ii=1:H,
            matrix(ii,1:I)=wI_est(I*(ii-1)+1:I*ii);
        end

        wR_est=zeros(1,H*I);
        for ii=1:I,
            wR_est(H*(ii-1)+1:H*ii)=matrix(1:H,ii); %zapis do
radkoveho vektoru z prokladaci matice
        end

        %dekodovani opakovacim kodem
        w_est=zeros(1,w_length);
```

```
        for ii=1:w_length,
            temp=sum(wR_est(m*(ii-1)+1:m*ii));
            if temp>0
                w_est(ii)=1;
            else
                w_est(ii)=-1;
            end
        end

        R_WEST(kk,:)=w_est;
        bit=8;
        for i=1:length(w_est),
            q(floor((i-1)/bit)+1,i-floor((i-
1)/bit)*bit)=num2str((w_est(i)+1)/2);
        end
        str=char(bin2dec(q))';

        disp(sprintf(' detekovany vodoznak cislo %d je:  %s
',kk,str));
    end
end

[total,bit]=size(R_WEST);
counter=0;

%prevod bit. sekvence na znaky
q=dec2bin(watermark,8);
for K=1:size(q,1),
    for L=1:8,
        w_bin(L+(K-1)*8)=str2num(q(K,L))*2-1; %
    end
end

%vypocet bit. error
for ii=1:total;
    for jj=1:bit;
        if  (R_WEST(ii,jj) == w_bin(jj))
        else
            counter=counter+1;
         end;
    end;
end;
bit_error=(counter/(total*bit))*100;

disp(' ');
disp('==========================================================================
====================');
disp(sprintf(' Bylo extrahovano: = %d vodoznaku',kk-1));
disp(sprintf(' doba extrakce byla: = %f sec',toc));
disp(sprintf(' bit error je : procent %f procent',bit_error));
disp('==========================================================================
====================');
warning on;
return;
```

Přiložené podprogramy:

```
function [S,X,X_angle,Delta] = spektrum(s);



FFT_SHIFT   = 384;
FFT_SIZE    = 512;
FFT_OVERLAP = (FFT_SIZE - FFT_SHIFT) / 2;
MIN_POWER = -200;




% Prepare the Hanning window
h = sqrt(8/3) * hanning(512, 'periodic');
w=sqrt(8/3)*hamming(FFT_SIZE)';
sw=s.*w;
% Power density spectrum

S=(fft(sw));
X_angle=angle(S);

X = 20 * log10(abs(fft(sw,512)) / FFT_SIZE);
%       20 * log10
% Normalization to the reference sound pressure level of 96 dB
Delta = 96 - max(X);
X = X +  Delta;
%X=X+96;


 function [Lsb,N_sb] = soundpresure (X,fftdelka);

N_sb=32;
scale = mpeg_scale;
Xmin = min(X);
n = fftdelka / 2 / 32; % Size of each subband

for i = 1:N_sb,
   local_max1 = Xmin;
   for j = 1:n,
      local_max1 = max(X((i - 1) * n + j), local_max1);
   end
   Lsb(i) = max(local_max1, 20 * log10(scale(i) * 32768) - 10);
end

function [tonal,nontonal, Xnm,Xtm] = Tonal_nontonal(X,TH,MAP);
%urceni lokalnich maxim, tonovych a netonovych slozek
Fs=44100;
BLOCK=512;
CB = kriticka_pasma(Fs);
N1=round(BLOCK/2);

poc=1;

loc_max = [];
  for k=2:BLOCK/2,
    if (X(k) > X(k-1)) & (X(k) >= X(k+1)),
```

```
      loc_max = [loc_max, k];
      loc_max1(poc, 2) = k;
      loc_max1(poc, 1) = X(k);
      poc = poc + 1;

    end;
  end;
  tonal = [];
  Xtm = [];
  X_nontonal = X;
  for i=2:size(loc_max,1),
    if (loc_max(i) > 2) & (loc_max(i) < 63),
      j = [-2,2];
    elseif (loc_max(i) >= 63) & (loc_max(i) < 127),
      j = [-3,-2,2,3];
    elseif (loc_max(i) >= 127) & (loc_max(i) <= 250),
      j = [-6:-2,2:6];
    else
      j = NaN;

    end;
    if isfinite(j),
      if X(loc_max(i,2)) >= max(X(loc_max(i,2)+j)) + 7,
        tonal = [tonal,loc_max(i)];
        Xtm = [Xtm, 10*log10(10.^( X(loc_max(i,2))/10 )+(10.^(
X(loc_max(i-1,2))/10 )+(10.^( X(loc_max(i+1,2))/10 )))];
        X_nontonal(loc_max(i,2)+[j,-1,0,1]) = -200;
      end;
    end;
  end;

  %TH=TH';
  N_crit = length(CB);
  Xnm = zeros(1,N_crit);
  nontonal = zeros(1,N_crit);

  for i=1:N_crit -1,
    j=[TH(CB(i),1):TH(CB(i+1),1)-1];

    Xnm(i) = 10*log10(sum(10.^( X_nontonal(j)/10 )));
    nontonal(i) = round(exp(mean(log(j))));
  end;
  j=[TH(CB(N_crit)):BLOCK/2+2];
   Xnm(N_crit) = 10*log10(sum(10.^( X_nontonal(j)/10 )));
  nontonal(N_crit) = round(exp(mean(log(j))));
  i=0;
  return



function
[D_tonal,D_nontonal,D_Xtm,D_Xnm]=decimace(Xtm,Xnm,tonal,nontonal,TH,LT
q,MAP);
% Odstraneni tonovych slozek pokud jejich hodnota je nizsi nez prah
ticha
  T = [];
  for i=1:length(tonal),
    if Xtm(i) >= TH(MAP(tonal(i)),3)
       T = [T,i]; end;
  end;
```

```
  Xtm = Xtm(T);
  tonal = tonal(T);

%Odstraneni netonovych slozek pokud jejich hodnota je nizsi nez prah
ticha
  N = [];
  for i=1:length(nontonal),
    if Xnm(i) >= TH(MAP(nontonal(i)),3)
       N = [N,i]; end;
  end;
  Xnm = Xnm(N);
  nontonal = nontonal(N);

  %dodatecna decimace, pokud tonove slozky jsou od sebe vzdaleny mene
  %jak 0.5Bark
  i=1;
  while i<length(tonal)-1,
    if abs( hzbark(tonal(i+1)) - hzbark(tonal(i)) ) < 0.5,
      if Xtm(i) > Xtm(i+1),
        Xtm = Xtm([1:i,i+2:length(tonal)]);
        tonal = tonal([1:i,i+2:length(tonal)]);
      else
        Xtm = Xtm([1:i-1,i+1:length(tonal)]);
     tonal = tonal([1:i-1,i+1:length(tonal)]);
      end;
    else
      i = i+1;
    end;
  end;
  D_Xtm=Xtm;
  D_Xnm=Xnm;
  D_tonal=tonal;
  D_nontonal=nontonal;


function  [LTtm, LTnm] = vypocet_individual_mask_prah(X, Tonal,
Nontonal,Xtm,Xnm, TH, Map)
% LTtm - individualni maskovaci prah pro tonove slozky
% LTnm - individualni maskovaci prah pro netonove slozky



LTtm = zeros(length(Tonal), length(TH))-200;
LTnm = zeros(length(Nontonal), length(TH))-200;
%**********************vypocet pro tononve
slozky************************
for i = 1:length(TH(:, 1))
   zi = TH(i, 2);  % prislusna frekvence kritickeho pasma

        for k = 1:length(Tonal),
          j  = Tonal(k);
          zj = TH(Map(j), 2);
          dz = zi - zj;          % rozdil frekvenci v Bark

          if (dz >= -3 & dz < 8)

             % Maskovaci index pro tonovou slozku
             avtm = -1.525 - 0.275 * zj - 4.5;
```

```
        %Maskovaci funkce
             if (-3 <= dz & dz < -1)
                vf = 17 * (dz + 1) - (0.4 * X(j) + 6);
             elseif (-1 <= dz & dz < 0)
                vf = (0.4 * X(j) + 6) * dz;
             elseif (0 <= dz & dz < 1)
                vf = -17 * dz;
             elseif (1 <= dz & dz < 8)
                vf = - (dz - 1) * (17 - 0.15 * X(j)) - 17;
             end

             LTtm(k, i) = Xtm(k) + avtm + vf;
          end
      end

%********************Vypocet pro netonove
slozky*************************
   for k = 1:length(Nontonal),
      j  = Nontonal(k);
      zj = TH(Map(j), 2);
      dz = zi - zj;          % Rozdil frekvenci v barcich

      if (dz >= -3 & dz < 8)


         avnm = -1.525 - 0.175 * zj - 0.5;      % Maskovaci index

         % Masking funkce
         if (-3 <= dz & dz < -1)
            vf = 17 * (dz + 1) - (0.4 * X(j) + 6);
         elseif (-1 <= dz & dz < 0)
            vf = (0.4 * X(j) + 6) * dz;
         elseif (0 <= dz & dz < 1)
            vf = -17 * dz;
         elseif (1 <= dz & dz < 8)
            vf = - (dz - 1) * (17 - 0.15 * X(j)) - 17;
         end

         LTnm(k, i) = Xnm(k) + avnm + vf;
      end
   end
end

function [LTg]=global_mask_prah(LTq,tonal,nontonal,LTtm,LTnm)
% vypocet globalniho maskovaciho prahu

  LTtm=LTtm';

  pom = 10.^(LTq/10);           % prah ticha
  for i=1:length(tonal),
    pom = pom + 10.^(LTtm(:,i)/10); % pro tonove masky
  end;

  LTnm=LTnm';
  for i=1:length(nontonal),
    pom = pom + 10.^(LTnm(:,i)/10); % pro netonove masky
  end;
  LTg = 10*log10(pom);
```

```
function [LTmin]=Min_mask_prah(X,N_sb,LTg,MAP)

% vypocet minimalniho maskovaciho prahu

  for m=1:32,
    jj = round( [(m-1)*length(X)/2/32+1:m*length(X)/2/32] );
    LTmin(m) = min( LTg(MAP(jj)) );
  end;


function bark = hzbark(hz)
%prevod hz na bark

bark=13*atan(.00076*hz)+3.5*atan((hz/7500).^2);

function scale = mpeg_scale

scale = [
   2.00000000000000; 1.58740105196820; 1.25992104989487;
1.00000000000000;
   0.79370052598410; 0.62996052494744; 0.50000000000000;
0.39685026299205;
   0.31498026247372; 0.25000000000000; 0.19842513149602;
0.15749013123686;
   0.12500000000000; 0.09921256574801; 0.07874506561843;
0.06250000000000;
   0.04960628287401; 0.03937253280921; 0.03125000000000;
0.02480314143700;
   0.01968626640461; 0.01562500000000; 0.01240157071850;
0.00984313320230;
   0.00781250000000; 0.00620078535925; 0.00492156660115;
0.00390625000000;
   0.00310039267963; 0.00246078330058; 0.00195312500000;
0.00155019633981;
   0.00123039165029; 0.00097656250000; 0.00077509816991;
0.00061519582514;
   0.00048828125000; 0.00038754908495; 0.00030759791257;
0.00024414062500;
   0.00019377454248; 0.00015379895629; 0.00012207031250;
0.00009688727124;
   0.00007689947814; 0.00006103515625; 0.00004844363562;
0.00003844973907;
   0.00003051757813; 0.00002422181781; 0.00001922486954;
0.00001525878906;
   0.00001211090890; 0.00000961243477; 0.00000762939453;
0.00000605545445;
   0.00000480621738; 0.00000381469727; 0.00000302772723;
0.00000240310869;
   0.00000190734863; 0.00000151386361; 0.00000120155435  ];

function [TH, MAPPING,LTq] =
Table_absolute_threshold(Fs,fftdelka,bitrate);

%Table 3-D.1b.:  Frequencies, Critical Band Rates and Absolute
Threshold
%Tabulka  je platna pro Layer I pro vzorkovaci frekvenci 44.1 kHz.
%Tuto tabulku jsem pouzil z HTTP://NEURON2.NET/LIBRARY/MPEG1/
%annex_d.doc
```

```
% Frequency | Crit Band rate | Absolute threshold
      TH = [
            86.13   0.850  25.87  ;   172.27   1.694  14.85  ;
           258.40   2.525  10.72  ;   344.53   3.337   8.50  ;
           430.66   4.124   7.10  ;   516.80   4.882   6.11  ;
           602.93   5.608   5.37  ;   689.06   6.301   4.79  ;
           775.20   6.959   4.32  ;   861.33   7.581   3.92  ;
           947.46   8.169   3.57  ;  1033.59   8.723   3.25  ;
          1119.73   9.244   2.95  ;  1205.86   9.734   2.67  ;
          1291.99  10.195   2.39  ;  1378.13  10.629   2.11  ;
          1464.26  11.037   1.83  ;  1550.39  11.421   1.53  ;
          1636.52  11.783   1.23  ;  1722.66  12.125   0.90  ;
          1808.79  12.448   0.56  ;  1894.92  12.753   0.21  ;
          1981.05  13.042  -0.17  ;  2067.19  13.317  -0.56  ;
          2153.32  13.577  -0.96  ;  2239.45  13.825  -1.38  ;
          2325.59  14.062  -1.79  ;  2411.72  14.288  -2.21  ;
          2497.85  14.504  -2.63  ;  2583.98  14.711  -3.03  ;
          2670.12  14.909  -3.41  ;  2756.25  15.100  -3.77  ;
          2842.38  15.283  -4.09  ;  2928.52  15.460  -4.37  ;
          3014.65  15.631  -4.60  ;  3100.78  15.795  -4.78  ;
          3186.91  15.955  -4.91  ;  3273.05  16.110  -4.97  ;
          3359.18  16.260  -4.98  ;  3445.31  16.405  -4.92  ;
          3531.45  16.547  -4.81  ;  3617.58  16.685  -4.65  ;
          3703.71  16.820  -4.43  ;  3789.84  16.951  -4.17  ;
          3875.98  17.079  -3.87  ;  3962.11  17.204  -3.54  ;
          4048.24  17.327  -3.19  ;  4134.38  17.447  -2.82  ;
          4306.64  17.680  -2.06  ;  4478.91  17.904  -1.32  ;
          4651.17  18.121  -0.64  ;  4823.44  18.331  -0.04  ;
          4995.70  18.534   0.47  ;  5167.97  18.730   0.89  ;
          5340.23  18.922   1.23  ;  5512.50  19.108   1.51  ;
          5684.77  19.288   1.74  ;  5857.03  19.464   1.93  ;
          6029.30  19.635   2.11  ;  6201.56  19.801   2.28  ;
          6373.83  19.963   2.46  ;  6546.09  20.120   2.63  ;
          6718.36  20.273   2.82  ;  6890.63  20.421   3.03  ;
          7062.89  20.565   3.25  ;  7235.16  20.705   3.49  ;
          7407.42  20.840   3.74  ;  7579.69  20.971   4.02  ;
          7751.95  21.099   4.32  ;  7924.22  21.222   4.64  ;
          8096.48  21.341   4.98  ;  8268.75  21.457   5.35  ;
          8613.28  21.676   6.15  ;  8957.81  21.882   7.07  ;
          9302.34  22.074   8.10  ;  9646.88  22.253   9.25  ;
          9991.41  22.420  10.54  ; 10335.94  22.575  11.97  ;
         10680.47  22.721  13.56  ; 11025.00  22.857  15.31  ;
         11369.53  22.984  17.23  ; 11714.06  23.102  19.34  ;
         12058.59  23.213  21.64  ; 12403.13  23.317  24.15  ;
         12747.66  23.414  26.88  ; 13092.19  23.506  29.84  ;
         13436.72  23.592  33.05  ; 13781.25  23.673  36.52  ;
         14125.78  23.749  40.25  ; 14470.31  23.821  44.27  ;
         14814.84  23.888  48.59  ; 15159.38  23.952  53.22  ;
         15503.91  24.013  58.18  ; 15848.44  24.070  63.49  ;
         16192.97  24.124  68.00  ; 16537.50  24.176  68.00  ;
         16882.03  24.225  68.00  ; 17226.56  24.271  68.00  ;
         17571.09  24.316  68.00  ; 17915.63  24.358  68.00  ;
         18260.16  24.398  68.00  ; 18604.69  24.436  68.00  ;
         18949.22  24.473  68.00  ; 19293.75  24.508  68.00  ;
         19638.28  24.541  68.00  ; 19982.81  24.573  68.00
      ];

N = length(TH(:, 1));

      % Prepocet frekvence do FFT indexu
       for i = 1:N,
```

```
         TH(i, 1) = round(TH(i, 1) / 44100 * 512);
      end
      % Generovani mapy mezi FFT daty a tabulkou
      for j = 1:TH(1,1),
         MAPPING(j) = 1;
      end
      for i = 2:N-1,
         for j = TH(i,1):TH(i+1,1)-1,
            MAPPING(j) = i;
         end
      end
      for j = TH(N):fftdelka/2,
         MAPPING(j) = N;
      end

     %uprava offset, pokud je bit. rychlost>=96kbps odecteme 12dB
     if (bitrate >= 96)
         LTq = TH(:,3) - 12;
      else
         LTq = TH(:,3);
      end

function CB = kriticka_pasma(Fs);

%vrati index v tabulce absolute threshold pro hranice kritickeho pasma
%
%
%
if (Fs == 44100),
CB = [1, 2, 3, 5, 6, 8, 9, 11, 13, 15, 17, 20, 23, 27, 32, 37, 45,
50,...
      55, 61, 68, 75, 81, 93, 106];
 else
       error('Frequency neni 44100');
    end
clear all
close all

%n=input('Zadej počet bitu:');
n=512;
%fs=input('Zadej vzorkovaci frekvenci:');
a=zeros(1,n);
c=rand(1,n);
  a=(c<=0.5);
%f=[A];
for i=1:n;
if (c(i)>=0.5);
    c(i)=1;
else c(i)=-1;
end;
end;

fid = fopen('pn2.mat', 'w');
for i=1:length(c)
    fprintf(fid, '%d\n', c(i));
end;

fclose(fid);

function IIR_DP;
```

```
close all;
clc
[filename,pathname] = uigetfile( ...
{'*.wav', 'Zvukov0 soubory nevodoznacene';...
'*.*', 'Vsechny soubory'},...
'Vyber soubor')
    %nacteni souboru
    [x,fvz,BITS]=wavread(fullfile(pathname, filename));
    wat_audio=input('zadej nazev pro ulozeni souboru vodoznaceneho
audio signalu ktery bude filtrovan dolni propusti:','s');
     x=x./max(x);
fm = 8000 / (fvz/2);
fs = 20000 / (fvz/2);
As = 3;
Am = 20;

[n, Wn] = buttord(fm, fs, As, Am);

[b, a] = butter(4, Wn);
freqz(b, a, 100, fvz)

y = filter(b,a,x);
y=y./max(y);
wavwrite(y,fvz,strcat(wat_audio));


% tato funkce decimuje zvukovy signal cinitelem M=2;
% a pote jej opet inkrementuje cinitelem L=2;


M=2;
L=2;

   [filename,pathname] = uigetfile( ...
{'*.wav', 'Zvukov0 soubory';...
'*.*', 'Vsechny soubory'},...
'Vyber soubor')
    %nacteni souboru
    [x1,fs1,BITS1]=wavread(fullfile(pathname, filename));
[Nx1,channels1] = size(x1);
wat_audio=input('zadej nazev pro ulozeni souboru vodoznaceneho audio
signalu ktery bude decimovan a nasledne inkrementovan:','s');

% delka vstupniho signalu
% index n
nx1 = 0:Nx1-1;
% casova osa t
tx1 = nx1(:)/fs1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
%% PODVZORKOVANI S CINITELEM M
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
y1=decimate(x1,M);
y1=y1./max(y1);
wavwrite(y1,fs1/2,strcat(wat_audio));

%nacteni souboru
```

```
[x2,fs2,BITS2]=wavread(strcat(wat_audio));
[Nx2,channels2] = size(x2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
%% NADVZORKOVANI S CINITELEM L
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% pocet vzorku nadvzorkovaneho signalu
%y=interp(x,L);
Ny = Nx2*L;
% index n nadvzorkovaneho signalu
ny = 0:Ny-1;
% casova osa nadvzorkovaneho signalu
ty = ny/(fs2*L);
% provedeni nadvzorkovani
y2=interp(x2,L);
y2=y2./max(y2);
wavwrite(y2,fs2*2,strcat(wat_audio));
```