

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

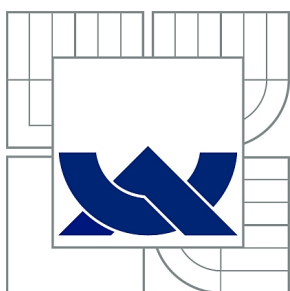
EXTRAKCE A MODIFIKACE VLASTNOSTÍ ČÍSLICOVÝCH ZVUKOVÝCH
SIGNÁLŮ
V DYNAMICKÉ ROVINĚ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

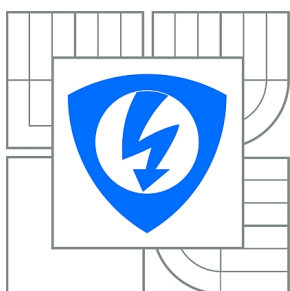
Bc. ONDŘEJ KRAMOLIŠ

BRNO 2010



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

EXTRAKCE A MODIFIKACE VLASTNOSTÍ ČÍSLICOVÝCH ZVUKOVÝCH SIGNÁLŮ V DYNAMICKÉ ROVINĚ

DIGITAL AUDIO SIGNAL FEATURE EXTRACTION AND MODIFICATION IN DYNAMIC PLANE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

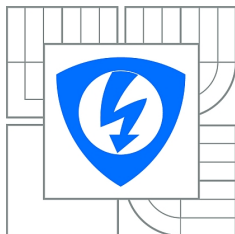
Bc. ONDŘEJ KRAMOLIŠ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JIŘÍ SCHIMMEL, Ph.D.

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Ondřej Kramoliš
Ročník: 2

ID: 83591
Akademický rok: 2009/2010

NÁZEV TÉMATU:

**Extrakce a modifikace vlastností číslicových zvukových signálů
v dynamické rovině**

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte metody extrakce vlastností číslicových zvukových signálů v dynamické rovině pracující v reálném čase. Zaměřte se na algoritmy měření efektivní, špičkové a stejnosměrné hodnoty signálu, sledování amplitudové obálky signálu a měření úrovně signálu typu Volume Unit a Peak Program podle standardů DIN a EBU. Algoritmy otestujte v prostředí Matlab. Navrhněte algoritmus změny dynamiky zvukového signálu pomocí obecné po částech lineární neklesající převodní funkce s možností nastavení doby náběhu a poklesu prahových úrovní. Algoritmy implementujte v jazyce C++ jako plug-in moduly technologie VST pro zpracování zvukových signálů v reálném čase.

DOPORUČENÁ LITERATURA:

- [1] HARTMANN, W., M. Signals, Sound, and Sensation. New York: Springer-Verlag, 1998. 647 p. ISBN 1-56396-283-7.
- [2] ZÖLZER, U. DAFX - Digital Audio Effects, 1st ed. New York: John Wiley & Sons, Ltd, 2002, 533 p. ISBN 0-471-49078-4.
- [3] ZÖLZER, U. Digital Audio Signal Processing, 1st ed. New York: McGraw-Hill, Inc., 1997, 290 p. ISBN 0-47-197226-6.

Termín zadání: 29.1.2010

Termín odevzdání: 26.5.2010

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

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Tato práce se zabývá základními metodami měření efektivní a špičkové hodnoty číslicového zvukového signálu, metodami měření úrovně hlasitosti a skutečné špičkové hodnoty dle doporučení ITU-R BS.1770-1 a číslicovými systémy pro změnu dynamiky signálu. Dále ukazuje výsledky metod měření efektivní a špičkové hodnoty a výsledky implementace dynamického procesoru s obecnou po částech lineární neklesající převodní funkcí a metod měření dle doporučení ITU-R BS.1770-1.

KLÍČOVÁ SLOVA

efektivní hodnota, špičková hodnota, RMS, ITU-R BS.1770-1, úroveň hlasitosti, skutečná špičková hodnota, dynamický procesor, kompresor, expander, limiter, šumová brána, převodní funkce, Bézierova křivka, explicitní Bézierova křivka, algoritmus de Casteljaou

ABSTRACT

This thesis deals with basic methods of root mean square and peak value measurement of a digital acoustic signal, algorithms to measure audio programme loudness and true-peak audio level according to recommendation ITU-R BS.1770-1 and digital systems for control of signal dynamic range. It shows achieved results of root mean square and peak value measurement and results of implementation of dynamic processor with general piecewise linear non-decreasing static curve and algorithms according to recommendation ITU-R BS.1770-1.

KEYWORDS

root mean square value, peak value, RMS, ITU-R BS.1770-1, loudness level, true-peak level, dynamic processor, compressor, expander, limiter, noise gate, static curve, Bézier curve, explicit Bézier curve, de Casteljaou's algorithm

KRAMOLIŠ O. *Extrakce a modifikace vlastností číslicových zvukových signálů v dynamické rovině*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2010. 66 s. Diplomová práce. Vedoucí práce Ing. Jiří Schimmel, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Extrakce a modifikace vlastností číslíkových zvukových signálů v dynamické rovině“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené 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

Děkuji vedoucímu diplomové práce Ing. Jiřímu Schimmelovi, Ph.D. za podporu v situaci, kdy bylo nutné práci na rok přerušit a za umožnění navázání na rozpracované téma.

V Brně dne

.....

(podpis autora)

OBSAH

Úvod	11
1 Metody měření parametrů zvukového signálu	12
1.1 Základní metody měření odhadu efektivní a špičkové hodnoty	12
1.1.1 Měření efektivní hodnoty pomocí plovoucího odhadu	12
1.1.2 Měření špičkové hodnoty pomocí plovoucího odhadu	16
1.2 Metoda měření efektivní hodnoty pomocí výpočtu z matematické definice	19
1.2.1 Zpracování signálu po úsecích	19
1.2.2 Charakter výstupního signálu v závislosti na překrytí a délce oken	21
1.3 Doporučení ITU-R BS.1770-1	24
1.3.1 Metoda měření úrovně hlasitosti	25
1.3.2 Metoda měření skutečné špičkové hodnoty	27
1.4 Srovnání uvedených metod měření	28
2 Dynamické procesory	30
2.1 Převodní charakteristika	31
2.1.1 Dělení dle tvaru převodní charakteristiky	31
2.2 Dynamický filtr	34
3 Popis implementace	36
3.1 Implementace metod měření	36
3.1.1 Implementace metody měření efektivní hodnoty pomocí výpočtu z matematické definice	36
3.1.2 Implementace základních metod měření odhadu efektivní a špičkové hodnoty	37
3.1.3 Implementace metody měření úrovně hlasitosti dle doporučení ITU-R BS.1770-1	37
3.1.4 Implementace metody měření skutečné špičkové hodnoty dle doporučení ITU-R BS.1770-1	38
3.2 Implementace převodní charakteristiky	41
3.2.1 Uživatelské rozhraní a jeho vnitřní implementace	41
3.2.2 Funkční rozhraní a jeho vnitřní implementace	46
3.3 Implementace dynamického filtru	50
3.4 Implementace dynamického procesoru	50
3.4.1 Parametry dynamického procesoru	52

3.4.2	Parametry měření dle doporučení ITU-R BS.1770-1	52
3.4.3	Kombinovaná stupnice	53
3.5	Výsledky implementace	53
3.6	Hodnocení výpočetní náročnosti	55
3.7	Slabá místa a chyby implementace	56
4	Závěr	58
	Literatura	59
	Seznam symbolů, veličin a zkratk	60
	Seznam příloh	63
A	Tabulky naměřených hodnot	64
B	Obsah přiloženého CD	66

SEZNAM OBRÁZKŮ

1.1	Blokové schéma měření efektivní hodnoty signálu pomocí plovoucího odhadu [8, 9].	13
1.2	Zobrazení amplitudově modulovaného harmonického signálu a plovoucího odhadu efektivní hodnoty při správném nastavení parametru t_{TAV} pro $t_{TAV} = 3$ ms a při kmitočtu vstupního harmonického signálu $f = 500$ Hz.	14
1.3	Zobrazení amplitudově modulovaného harmonického signálu a plovoucího odhadu efektivní hodnoty při špatném nastavení parametru t_{TAV} pro $t_{TAV} = 0,2$ ms a $t_{TAV} = 50$ ms při kmitočtu vstupního harmonického signálu $f = 500$ Hz.	15
1.4	Zobrazení amplitudově modulovaného harmonického signálu a plovoucího odhadu efektivní hodnoty při správném nastavení parametru t_{TAV} pro $t_{TAV} = 3$ ms a při kmitočtu vstupního harmonického signálu $f = 1000$ Hz.	16
1.5	Blokové schéma měření špičkové hodnoty signálu pomocí plovoucího odhadu [8, 9].	17
1.6	Zobrazení amplitudově modulovaného harmonického signálu a plovoucího odhadu špičkové hodnoty při správném i špatném nastavení a pro hodnoty parametrů t_{AT} a t_{RT} uvedených nad patřičným průběhem a při kmitočtu vstupního harmonického signálu $f = 500$ Hz.	18
1.7	Schematické znázornění výpočtu efektivní hodnoty u signálu děleného do bloků bez překrytí úseků signálu vybíraných okny.	20
1.8	Schematické znázornění výpočtu efektivní hodnoty u signálu děleného do bloků pro překrytí úseků signálu vybíraných okny 1/3.	21
1.9	Zobrazení amplitudově modulovaného harmonického signálu o kmitočtu $f = 500$ Hz a vypočítaného průběhu efektivní hodnoty pro délku okna odpovídající polovině periody vstupního signálu a pro nejmenší a největší velikost překrytí oken.	22
1.10	Zobrazení amplitudově modulovaného harmonického signálu o kmitočtu $f = 500$ Hz a vypočítaného průběhu efektivní hodnoty pro velmi krátké okno a pro nejmenší a největší velikost překrytí oken.	23
1.11	Zobrazení amplitudově modulovaného harmonického signálu o kmitočtu $f = 500$ Hz a vypočítaného průběhu efektivní hodnoty pro velmi dlouhé okno a pro nejmenší a největší velikost překrytí oken.	23
1.12	Blokové schéma metody měření úrovně hlasitosti dle BS.1770-1 [3].	26
1.13	Amplitudová kmitočtová charakteristika filtru zohledňujícího akustický stín hlavy (nahore) a filtru dle váhovací křivky RLB (dole) [3].	27

1.14	Blokové schéma metody měření skutečné špičkové hodnoty dle doporučení ITU-R BS.1770-1 [3].	28
2.1	Obecné schéma dynamického procesoru [7, 8].	31
2.2	Převodní charakteristiky kompresoru a expanderu při poměru komprese a expanze 2 : 1 [7, 8].	32
2.3	Převodní charakteristiky limiteru a šumové brány [7, 8].	33
2.4	Převodní charakteristika kombinovaného systému [8, 7].	34
2.5	Blokové schéma dynamického filtru [8, 9]	35
3.1	Blokové schéma implementace metody měření skutečné špičkové hodnoty dle doporučení ITU-R BS.1770-1.	38
3.2	Amplitudové kmitočtové charakteristiky interpolačního filtru (a)) a filtru preemfáze (b)) a chyba kompenzace útlumu interpolačního filtru pro kmitočet od 0 Hz do 24 kHz (c)).	39
3.3	Srovnání původního signálu o vzorkovacím kmitočtu $f_{vz0} = 48$ kHz (červený) se signálem nadvzorkovaným zvoleným způsobem (modrý) a se signály bez preemfáze – pouze interpolované ideálním filtrem (zelený) a filtrem s nízkým řádem (tyrkysový).	40
3.4	Průběhu ukazatele skutečné špičkové hodnoty signálu (červeně) a průběh vstupního zvukového signálu s původním vzorkovacím kmitočtem v oblasti špičkových hodnot (modře).	41
3.5	Převodní funkce rozdělená na tři křivky v různých módech.	43
3.6	Převodní funkce po rozdělení jednotlivých křivek na dvě nové v relativním čase $t = 0, 5$	44
3.7	Obecná Bézierova křivky (plnou čarou) s kontrolními body P a odvozená explicitní Bézierova křivka (čárkovaně) s kontrolními body R	47
3.8	Závislost relativního času t na poloze bodu na Bézierově křivce v ose X pro obecnou Bézierovu křivku (plnou čarou) a explicitní Bézierovu křivku (čárkovaně).	47
3.9	Obecná Bézierova křivka (plnou čarou) a dvě odvozené explicitní Bézierovy křivky (čárkovaně).	48
3.10	Uživatelské rozhraní pro modelaci převodní charakteristiky.	49
3.11	Uživatelské rozhraní dynamického procesoru a měření dle doporučení ITU-R BS.1770-1.	51
3.12	Převodní funkce kombinovaného systému použitá pro úpravu signálu na obrázku 3.13 pro dobu náběhu 5 ms, dobu poklesu 15 ms a pro reakci systému na efektivní hodnotu.	53
3.13	Časový průběh signálu s upravenou dynamikou kombinovaným systémem z obrázku 3.12 (nahore) a časový průběh původního signálu (dole) při shodné úrovni hlasitosti obou signálů $L_K = -25, 1$ LUFS.	54

ÚVOD

Tato práce se zabývá problematikou zpracování číslicových zvukových signálů v dynamické rovině metodami pracujícími v reálném čase. Základními typy těchto systémů, či konkrétněji zvukových efektů, k tomuto účelu v praxi používaných, jsou kompresory, expandery, limity a šumové brány. Ty tvoří stavební kameny v systémech složitějších, mezi něž je možné zařadit například kompander. Výše zmíněné efekty se zkráceně nazývají *dynamické procesory*. Jejich popisu se věnuje kapitola 2.

Funkce dynamických procesorů je založena na ovlivňování vstupního signálu určitým způsobem v závislosti na signálu řídicím. Není podmínkou, aby systém nemohl být řízen *libovolným* signálem, nicméně pro běžné účely se sleduje určitý parametr signálu *vstupního*, například efektivní nebo špičková hodnota, ze kterého je následně řídicí signál odvozen. V kapitole 1 jsou shrnuty metody měření různých parametrů zvukových signálů. Text kapitoly postupuje od metod základních až k metodám dle doporučení ITU-R BS.1770-1.

V jednotlivých částech kapitoly 3 jsou ukázány a popsány výsledky práce. Část 3.1 shrnuje implementace metod měření efektivní a špičkové hodnoty signálu a metod měření dle doporučení ITU-R BS.1770-1. Část 3.2 se podrobně věnuje způsobu implementace převodní funkce dynamického procesoru. Část 3.3 popisuje implementaci filtru řídicího signálu. Část 3.4 ukazuje implementaci dynamického procesoru jako celku s využitím předešlých implementací. Závěrem kapitoly jsou hodnoceny výsledky.

1 METODY MĚŘENÍ PARAMETRŮ ZVUKOVÉHO SIGNÁLU

Aby dynamické procesory pracovali správně, či výstižněji řečeno dle očekávání uživatele, je mimo bezchybné funkčnosti procesoru samotného nutné, aby signál řídicí jejich činnost odpovídal co možná nejvíce skutečné velikosti měřeného parametru vstupního signálu. Pokud tato podmínka splněna nebude, chování celého systému ovládaného přímo či nepřímo tímto signálem bude nepředvídatelné a s největší pravděpodobností i nežádoucí.

Sledovanými parametry zvukového signálu jsou nejčastěji efektivní a špičková hodnota. Základní metody jejich měření (přesněji řečeno *získávání odhadu*) publikované v [8] tvoří první část této kapitoly. Následuje popis metody založené na *výpočtu* efektivní hodnoty signálu vycházející z matematické definice a metod měření dle doporučení ITU-R BS.1770-1. Závěrem jsou srovnány výhody a nevýhody jednotlivých přístupů.

1.1 Základní metody měření odhadu efektivní a špičkové hodnoty

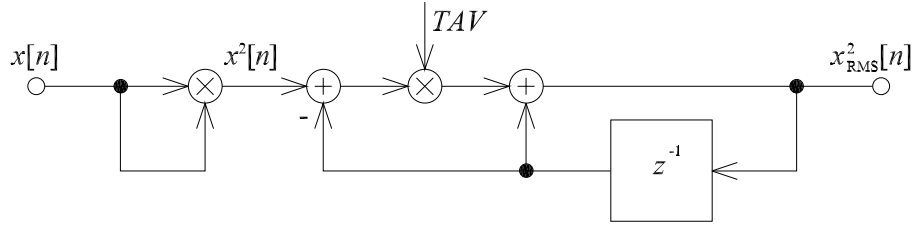
Jak bylo v úvodu této kapitoly naznačeno, nejedná se u těchto metod o měření v pravém slova smyslu, nýbrž o získávání přibližné velikosti daného parametru. Výslednou hodnotu je tedy možné označit jako plovoucí odhad skutečné velikosti hledaného parametru signálu. Při nevhodném nastavení vnitřních parametrů těchto systémů může být výsledek značně odlišný od správné hodnoty.

1.1.1 Měření efektivní hodnoty pomocí plovoucího odhadu

Na obrázku 1.1 je uvedeno blokové schéma měření efektivní hodnoty zvukového signálu dle [8]. Jak je vidět, celý tento systém je možné realizovat pouze pomocí operací součtu a násobení; nutné je také uchovávat výstupní vzorek pro mezivýpočty v následující vzorkovací periodě. Výstupem ovšem není přímo efektivní hodnota, jak je uváděno v [8], nýbrž její druhá mocnina. To může působit drobné komplikace zejména u implementací na signálových procesorech; s výhodou je však možné provést odmocnění výstupního signálu až po následném převodu do logaritmického měřítka (ve kterém dynamické procesory pracují) pouhým dělením dvěma.

Jediným parametrem, označeným jako TAV , je ovládáno chování celého systému a je definován následujícím vztahem [8]

$$TAV = 1 - e^{-2,2T/t_{TAV}}, \quad (1.1)$$



Obr. 1.1: Blokové schéma měření efektivní hodnoty signálu pomocí plovoucího odhadu [8, 9].

kde t_{TAV} je proměnná s významem času v sekundách a T je vzorkovací perioda. Parametr t_{TAV} je významově podobný době náběhu (attack time) a době poklesu (release time), ovšem jsou jím popsány vlastnosti systému jak při nárůstu, tak i poklesu okamžité hodnoty sledované veličiny. Diferenční rovnice systému je následující [9]

$$x_{\text{RMS}}^2[n] = (1 - TAV)x_{\text{RMS}}^2[n - 1] + TAVx^2[n], \quad (1.2)$$

kde $x_{\text{RMS}}^2[n]$ je výstupní signál systému a $x[n]$ je vstupní signál systému.

Na obrázku 1.2 je ukázán výstup popisovaného systému na amplitudově modulovaném harmonickém signálu o kmitočtu $f = 500$ Hz při velikosti parametru $t_{TAV} = 3$ ms. Tenkou plnou čarou je vykreslen vstupní signál, tlustá plná čára popisuje plovoucí odhad efektivní hodnoty vstupního signálu, a tedy i výstup celého systému a správná okamžitá velikost efektivní hodnoty je zobrazena tenkou čerchovanou čarou. V levém horním rohu je detail vyznačené oblasti. Vstupní harmonický signál byl modulován obdelníkovým neperiodickým signálem s proměnlivou střídou i velikostí a signálem s velikostí lineárně narůstající.

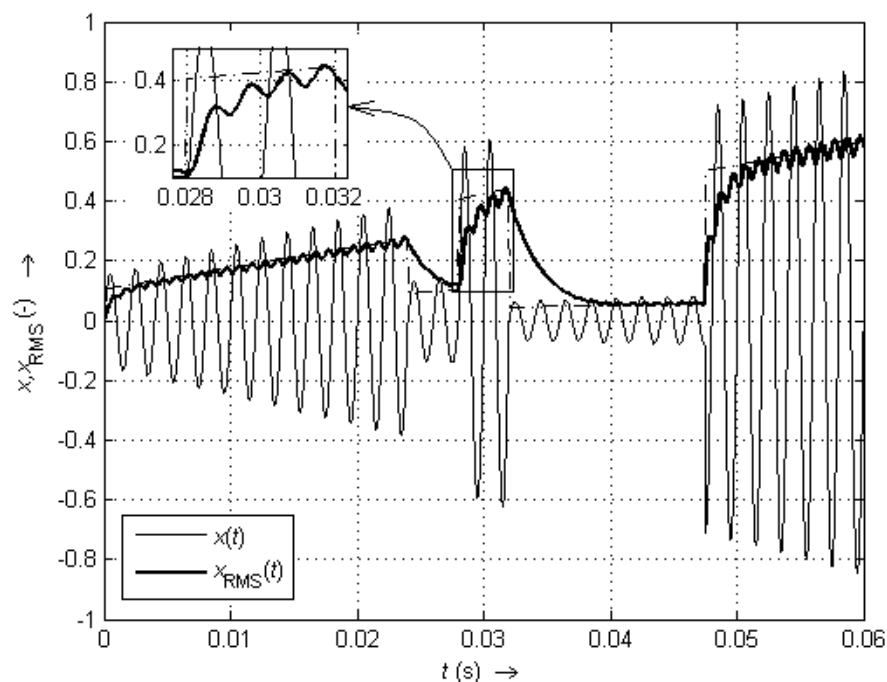
Charakteristické vlastnosti systému

Je možné najít tři vlastnosti, které určitým způsobem charakterizují popisovaný systém. Jsou to tyto:

- setrvačnost,
- kolísavost výstupního signálu,
- citlivost na správné nastavení.

Ještě než budou podrobně popsány, je vhodné uvést, že obdobně lze charakterizovat i systém měřící špičkovou hodnotu vstupního signálu popsany v části 1.1.2.

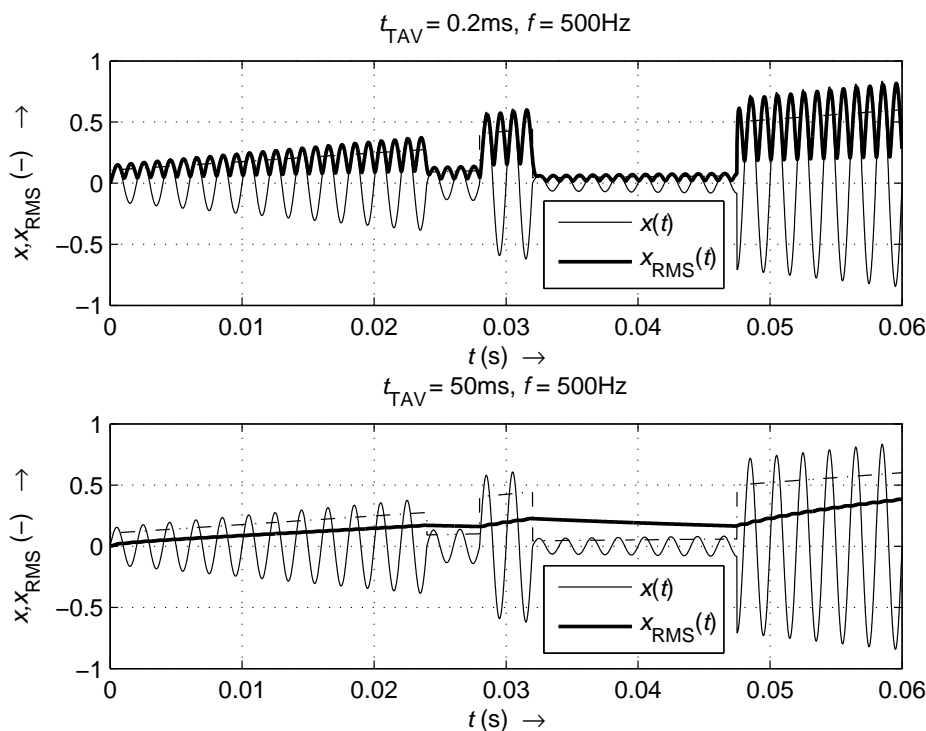
Setrvačnost systému je dána velikostí parametru t_{TAV} . Rychlost reakce na skokové změny v efektivní hodnotě vstupního signálu je tedy nepřímo úměrná velikosti zmíněného parametru t_{TAV} . Na obrázku 1.2 je vidět, že při rychlých změnách obálky



Obr. 1.2: Zobrazení amplitudově modulovaného harmonického signálu a plovoucího odhadu efektivní hodnoty při správném nastavení parametru t_{TAV} pro $t_{TAV} = 3$ ms a při kmitočtu vstupního harmonického signálu $f = 500$ Hz.

ve střední části se velikost odhadu nestačí na hodnotě správné ani ustálit. Naopak při plynulém nárůstu amplitudy vstupního signálu, pozorovatelné v okrajových oblastech, nevykazuje činnost systému žádné podobné nedostatky. Projev setrvačnosti je na obrázku 1.2 velmi patrný, nicméně, jak bude dále ukázáno, při odlišném nastavení parametru t_{TAV} tomu tak být nemusí; nebude tomu ovšem bez zhoršení vlastnosti uvedené jako druhé v pořadí a popsané v následujícím odstavci.

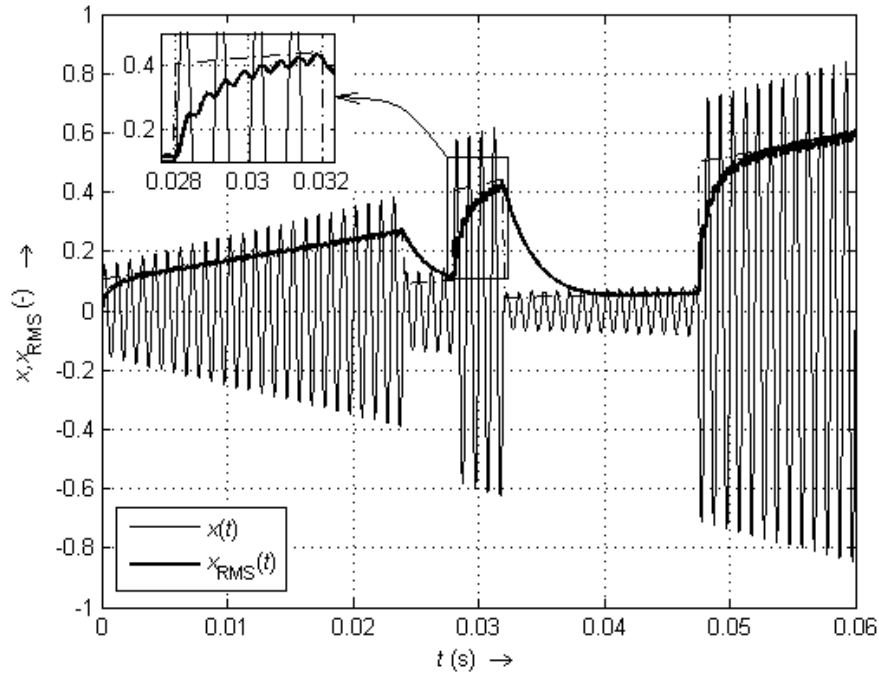
Kolísavost výstupního signálu, jak bylo naznačeno, je tedy značně závislá na nastavení parametru t_{TAV} a je určitým protikladem setrvačnosti. Na obrázku 1.3 jsou zobrazeny dva průběhy, které se od obrázku 1.2 liší pouze nastavením parametru t_{TAV} , který byl v prvním uvedeném případě roven 3 ms. Při velmi malém t_{TAV} průběh odhadu efektivní hodnoty sice nevykazuje takřka žádnou setrvačnost, navzdory tomu však téměř kopíruje absolutní hodnotu vstupního signálu a velikost jeho rozkmitu se tak blíží až k hodnotě amplitudy vstupního signálu. Takovýto výsledek je nepřijatelný a nežádoucí. Obdobně je tomu i v dalším uvedeném případě, kdy parametr t_{TAV} byl volen naopak příliš velký a ač je průběh výstupního signálu téměř hladký, tak nejenže nedokáže úspěšně sledovat rychlé změny obálky, ale dokonce nemá ani přílišnou tendenci blížit se ke správné velikosti efektivní hodnoty. Vlivem působícím na míru kolísavosti není ovšem jen parametr t_{TAV} , ale i kmitočet vstup-



Obr. 1.3: Zobrazení amplitudově modulovaného harmonického signálu a plovoucího odhadu efektivní hodnoty při špatném nastavení parametru t_{TAV} pro $t_{\text{TAV}} = 0,2 \text{ ms}$ a $t_{\text{TAV}} = 50 \text{ ms}$ při kmitočtu vstupního harmonického signálu $f = 500 \text{ Hz}$.

ního signálu. Srovnáním obrázků 1.2 a 1.4 je zřejmé, že celkové průběhy plovoucího odhadu se vzájemně nikterak neliší. Ovšem při pohledu na zvětšenou část je patrné, že kolísavost pro signál s vyšším kmitočtem je nižší. Analogie s dvoucestným usměrňovačem střídavého napětí je zjevná.

Citlivost na správné nastavení je u tohoto systému značná a jak bylo popsáno výše, jednotlivé výsledky se mohou velmi lišit. Je nutné tedy dbát zvýšené pozornosti při volbě parametru t_{TAV} a dopředu stanovit požadavek, zda má být upřednostněna nižší setrvačnost systému či menší kolísavost výstupního signálu. Je také vhodné uplatnit případnou znalost charakteru vstupního signálu a jeho kmitočtového obsahu, který je pro každou barvu zvuku, a tedy i jednotlivý hudební nástroj či lidský hlas, specifický. Jeví se tedy jako vhodné upravovat dynamiku již na úrovni samostatných stop nahrávky a odstranit nedostatky jak z hlediska použitého zdroje signálu (magnetické snímače, mikrofon, ap.), tak z pohledu muzikantského (přílišný důraz na hru nebo „fouknutí“ do mikrofonu v určité pasáži) a při závěrečné mixáži nastavit výslednou dynamiku a hlasitost nahrávky požadovanou dle typu záznamového média a reprodukčního zařízení.



Obr. 1.4: Zobrazení amplitudově modulovaného harmonického signálu a plovoucího odhadu efektivní hodnoty při správném nastavení parametru t_{TAV} pro $t_{TAV} = 3$ ms a při kmitočtu vstupního harmonického signálu $f = 1000$ Hz.

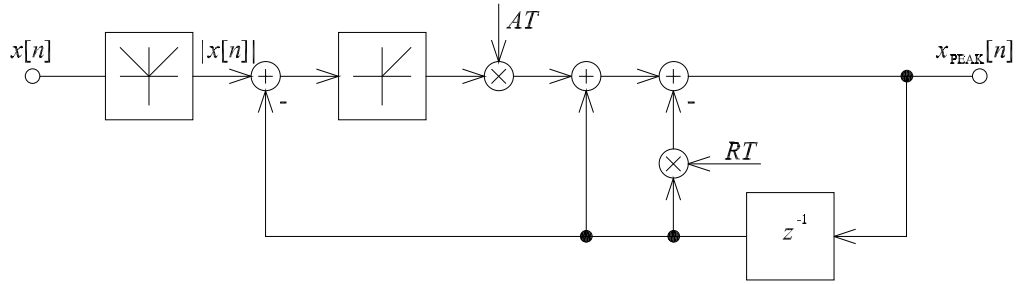
1.1.2 Měření špičkové hodnoty pomocí plovoucího odhadu

Jak bylo uvedeno v předchozí části 1.1.1, charakteristické vlastnosti popsaného systému, získávajícího efektivní hodnotu pomocí plovoucího odhadu, jsou shodné s vlastnostmi systému, jehož výstupem je odhad hodnoty špičkové. Jsou jimi tedy *setrvačnost* systému, *kolísavost* výstupního signálu a *citlivost* na nastavení.

Jak je vidět na blokovém schématu 1.5, popisujícím systém pro získání odhadu špičkové hodnoty, rozdílem oproti systému uvedenému v předcházející části není jen vlastní zapojení jednotlivých prvků, ale i počet parametrů určujících chování systému. Jsou jimi AT a RT vycházející z doby náběhu (attack time), respektive doby poklesu (release time), a jsou definovány obdobně jako parametr TAV ve vztahu (1.1) takto

$$\begin{aligned} AT &= 1 - e^{-2,2T/t_{AT}}, \\ RT &= 1 - e^{-2,2T/t_{RT}}, \end{aligned} \tag{1.3}$$

kde t_{AT} je doba náběhu, t_{RT} je doba poklesu a T je vzorkovací perioda. Je tedy zřejmé, že systém poskytuje možnost ovlivňovat jeho vlastnosti jak při skokovém nárůstu velikosti obálky vstupního signálu (t_{AT}), tak i při jejím náhlém poklesu



Obr. 1.5: Blokové schéma měření špičkové hodnoty signálu pomocí plovoucího odhadu [8, 9].

(t_{RT}), odděleně. Obvykle je doba náběhu volena přibližně jako polovina doby poklesu. Pokud je absolutní hodnota vstupního signálu větší než předchozí velikost výstupního signálu, je diferenční rovnice systému následující [9]

$$x_{PEAK}[n] = (1 - AT - RT)x_{PEAK}[n - 1] + AT|x[n]|, \quad (1.4)$$

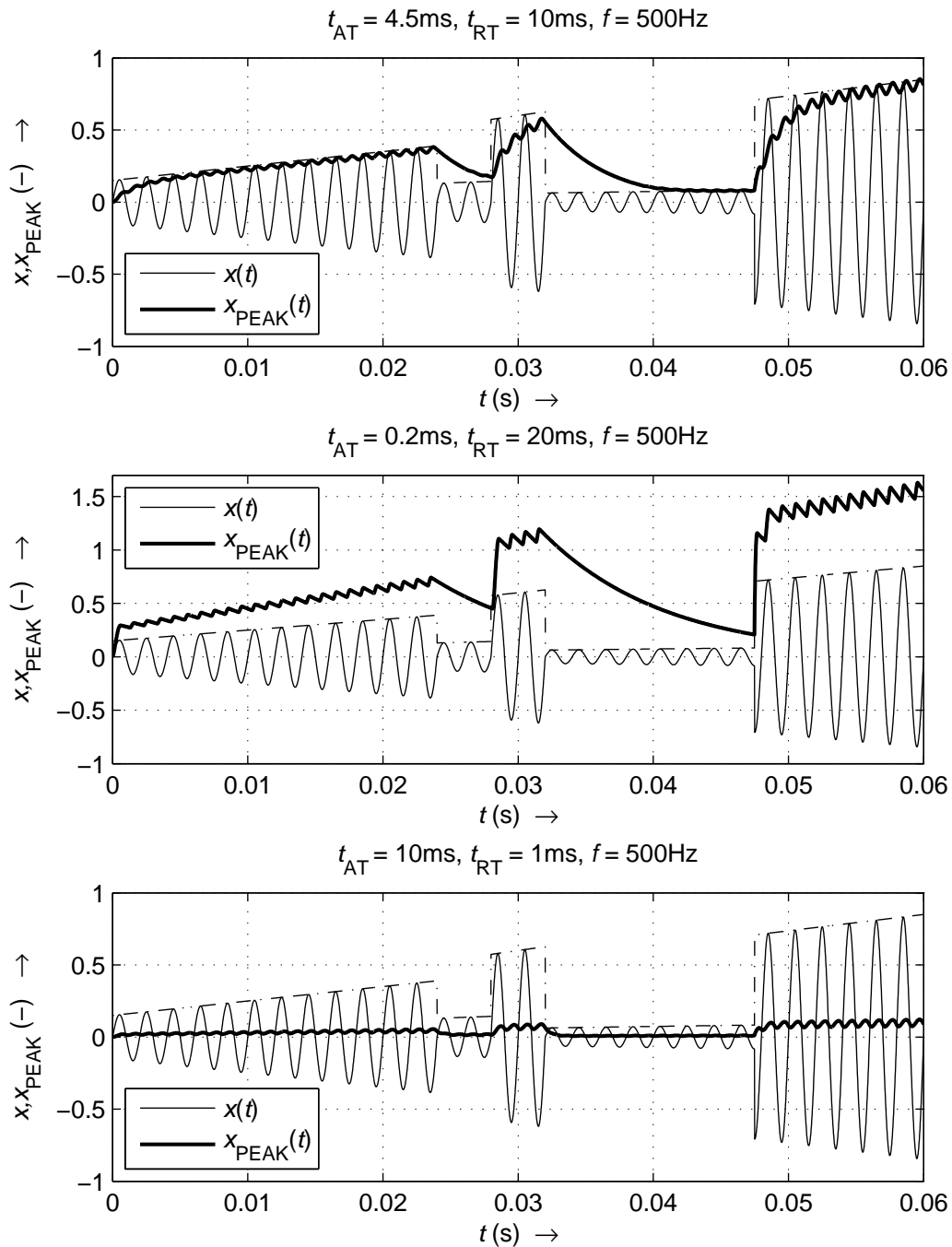
kde $x_{PEAK}[n]$ je výstupní signál systému a $x[n]$ je vstupní signál systému. Pokud je absolutní hodnota vstupního signálu menší než předchozí velikost výstupního signálu, je diferenční rovnice systému takováto [9]

$$x_{PEAK}[n] = (1 - RT)x_{PEAK}[n - 1]. \quad (1.5)$$

Princip funkce zapojení lze jednoduše popsat následovně. Nejprve je ze vstupního signálu získána absolutní hodnota a poté po odečtení velikosti předcházejícího výstupního vzorku je signál přiveden na blok s nelineární převodní charakteristikou, který záporným vstupním hodnotám přiřadí nulovou funkční hodnotu a kladné ponechá beze změny. Následně je už pouze prováděno sčítání patřičných vzorků signálu a jejich násobení uvedenými parametry AT a RT . Výstupem je okamžitá hodnota odhadu špičkové hodnoty vstupního signálu.

Na obrázku 1.6 je zobrazen výstup systému při vstupním signálu totožném jako v části 1.1.1. Do horní části je umístěn případ pro správné nastavení doby náběhu a doby poklesu ($t_{AT} = 4,5 \text{ ms}$, $t_{RT} = 10 \text{ ms}$). Je opět vidět charakteristické chování systému jak při rychlých skokových změnách velikosti obálky vstupního signálu, tak i při jejím postupném nárůstu.

Jak bylo na začátku této části zmíněno, rozhraní systému umožňuje nastavení jak doby náběhu t_{AT} , tak i doby poklesu t_{RT} samostatně. Toto poskytuje větší stupeň volnosti, nicméně může dojít i k nežádoucímu nastavení, které u systému odhadujícího efektivní hodnotu signálu nastat nemohlo. Dva případy nevhodné kombinace velikostí doby náběhu a doby poklesu, kdy se vzájemně *výrazně liší*, jsou ukázány na prostředním a dolním průběhu obrázku 1.6.



Obr. 1.6: Zobrazení amplitudově modulovaného harmonického signálu a plovoucího odhadu špičkové hodnoty při správném i špatném nastavení a pro hodnoty parametrů t_{AT} a t_{RT} uvedené nad patričným průběhem a při kmitočtu vstupního harmonického signálu $f = 500\text{ Hz}$.

V obou zmíněných situacích je výstupní signál naprosto nepoužitelný a není již možné hovořit o sledování špičkové hodnoty. V prvním případě, kde $t_{AT} \ll t_{RT}$ je doba náběhu t_{AT} natolik krátká, že dojde k značnému překmitu okamžité hodnoty odhadu. Vzhledem k výrazně delší době poklesu oproti době náběhu nestihne výstupní signál klesat a drží se na úrovni překmitu. Naopak v druhém případě, kdy $t_{AT} \gg t_{RT}$, výstupní signál narůstá natolik pomalu oproti rychlosti poklesu, že se pohybuje v oblasti těsně nad nulovou hodnotou.

Závěrem této části je nutné uvést, že obrázky 1.2, 1.3, 1.4 a 1.6 jsou výstupem modelů (upravených dle požadavků této práce) vytvářených v Simulinku ve cvičeních z předmětu Číslicové zpracování akustických signálů vyučovaném Ing. Miroslavem Balíkem, Ph.D. Model pro měření odhadu špičkové hodnoty se od zapojení na obrázku 1.5 liší blokem zesílení zařazeným na výstup systému. Tento způsob byl přijat i pro implementaci dané metody popsané v části 3.1.2.

1.2 Metoda měření efektivní hodnoty pomocí výpočtu z matematické definice

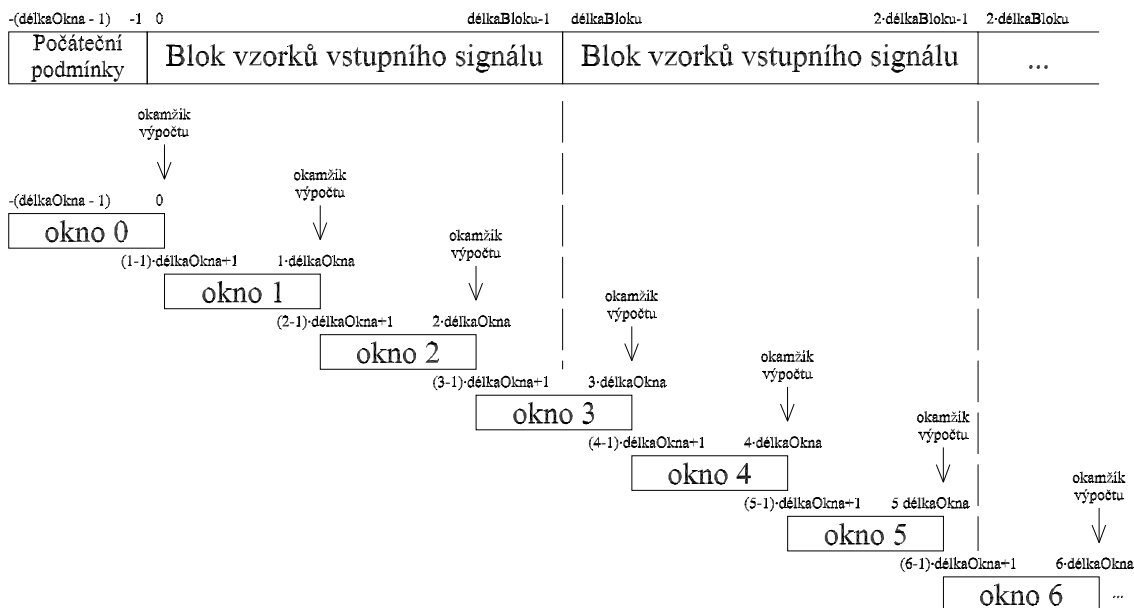
Pro výpočet vzorku výstupního signálu u metod využívajících plovoucího odhadu (uvedených v částech 1.1.1 a 1.1.2) byla potřeba znalost pouze současné hodnoty vstupního signálu a předchozí hodnoty signálu výstupního. Tento počet vzorků signálu nutných pro výpočet zůstává z principu metody po celou dobu *neměnný*. Odlišně je tomu však u metody vycházející přímo z matematické definice

$$RMS = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} x^2[n]}, \quad (1.6)$$

kde $x[n]$ je vzorek vstupního signálu v diskrétním čase n , N je počet vzorků, nad nimiž je prováděn výpočet, a RMS je velikost efektivní hodnoty signálu $x[n]$ v čase od $n = 0$ do $n = N - 1$. Je tedy zřejmé, že počet vzorků signálu nutných pro výpočet je obecně *libovolný*.

1.2.1 Zpracování signálu po úsecích

Vztah (1.6) však předpokládá znalost aktuálního a $N - 1$ následujících vzorků signálu. To ovšem při zpracování v reálném čase není obecně možné zaručit. Z hlediska kauzality lze tedy vztah (1.6) označit jako *nekauzální*. Tuto variantu bez dalších úprav by bylo možné použít pouze a jen tehdy, pokud by signál byl zpracováván po blocích délky násobků N . V praxi se blokové zpracování samozřejmě používá,



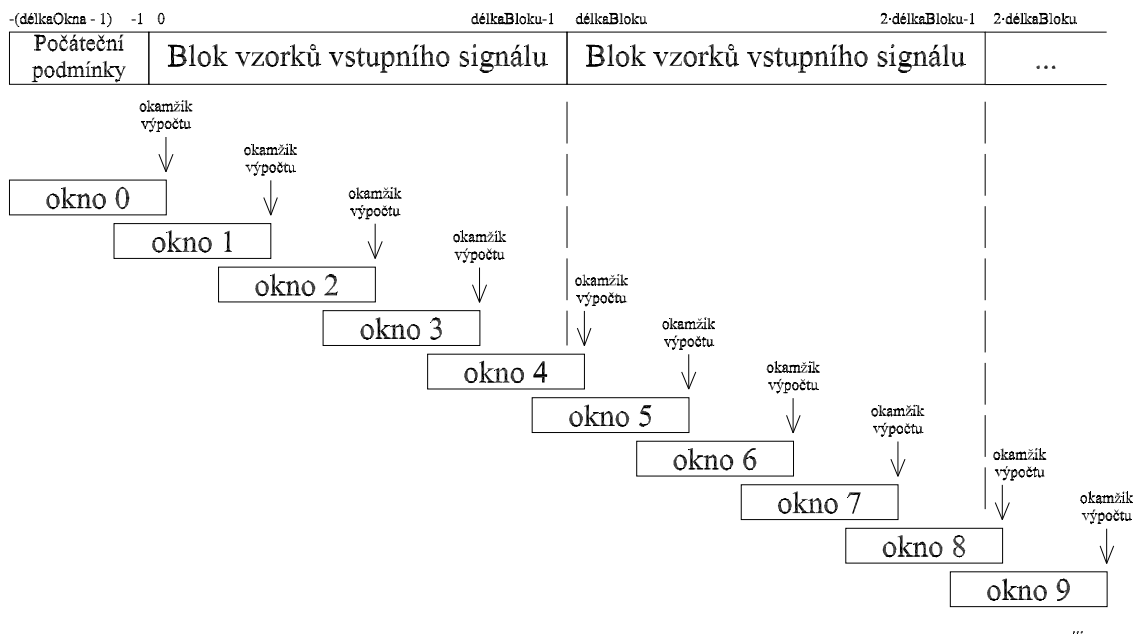
Obr. 1.7: Schematické znázornění výpočtu efektivní hodnoty u signálu děleného do bloků bez překrytí úseků signálu vybíraných okny.

nicméně nastavení délky bloku je věcí konkrétního zvukového zařízení a jeho ovladačů a za běhu jej měnit nelze. Proto je vhodné vztah (1.6) upravit do *kauzální* podoby

$$x_{\text{RMS}}[m] = \sqrt{\frac{1}{N} \sum_{n=m}^{m-N+1} x^2[n]}, \quad (1.7)$$

kde $x_{\text{RMS}}[m]$ je velikost efektivní hodnoty signálu $x[n]$ od $n = m$ do $n = m - N + 1$ v čase m . V tomto případě je k výpočtu potřeba aktuální vzorek signálu a $N - 1$ vzorků předchozích. Výběr konkrétních vzorků signálu od $n = m$ do $n = m - N + 1$ je možné popsat jako násobení signálu $x[n]$ obdelníkovým oknem délky N a počátkem v čase $m - N + 1$. Je samozřejmě možné použít i libovolné jiné okno, které omezí nepřesnost výpočtu vlivem necelého počtu period signálu v okně délky N .

Na obrázku 1.7 je schematicky znázorněn výpočet efektivní hodnoty signálu děleného do bloků. Nad okraji každého bloku signálu a každého okna jsou umístěny výrazy, které svou hodnotou odpovídají diskrétnímu času m pro krajní vzorky daného objektu. Je vidět, že jednotlivá okna vybírající patřičný úsek signálu na sebe plynule navazují (nebo-li bez překrytí oken). Okamžiky výpočtu efektivní hodnoty, označené šipkou, jsou tedy od sebe vzdáleny právě o N vzorků. Každý vzorek signálu je pro takovýto případ do výpočtu zahrnut právě *jednou*. To ovšem neplatí u situace na obrázku 1.8, kde jsou sousední okna překryta vždy z jedné třetiny a polovina vzorků signálu tak bude do výpočtu efektivní hodnoty zahrnuta *dvakrát*. Úměrně tomu naroste samozřejmě výpočetní náročnost. Nicméně doba mezi dvěma okamžiky



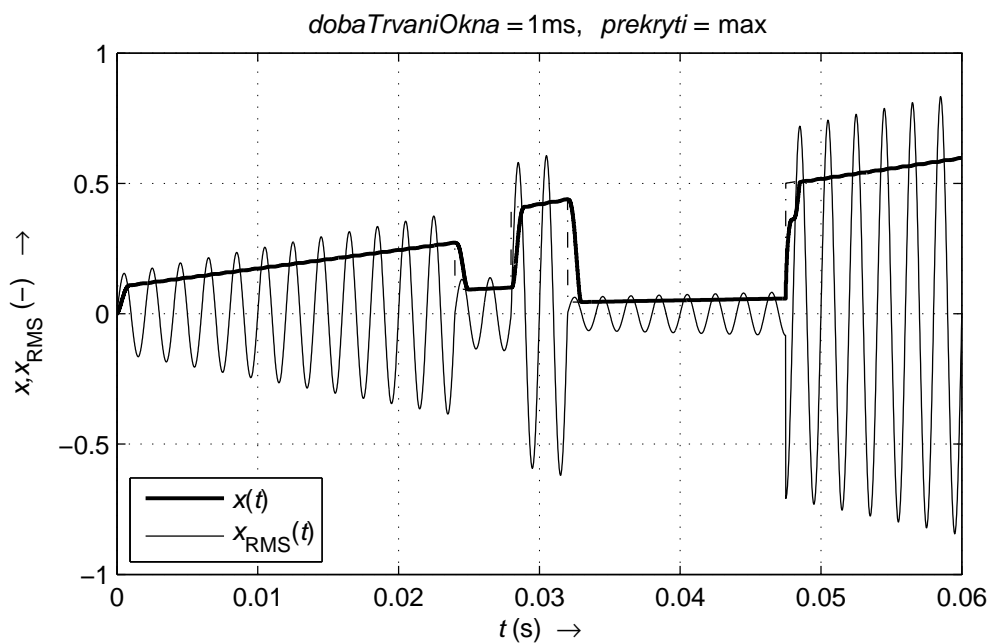
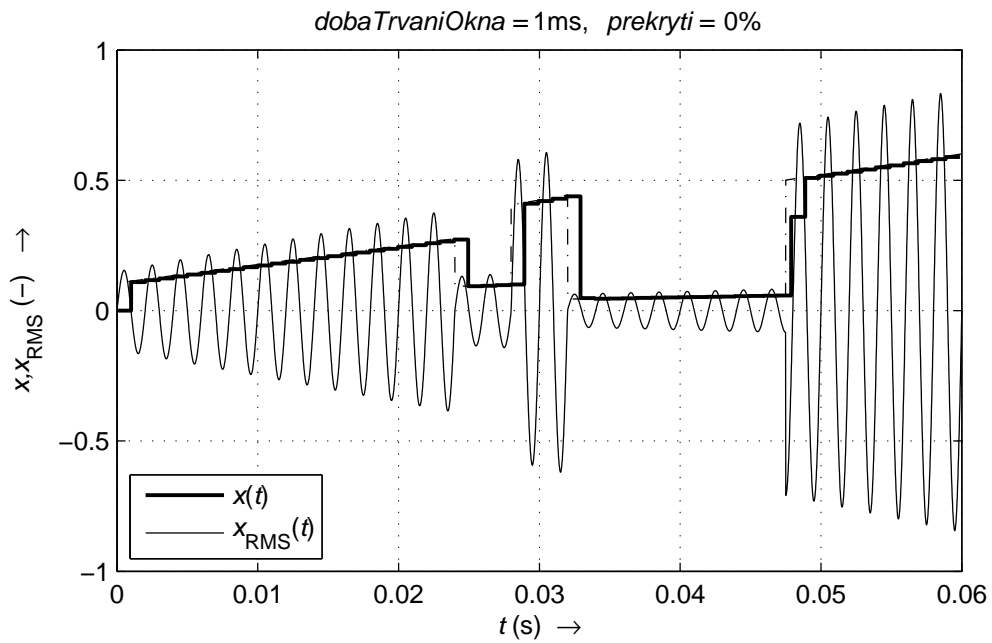
Obr. 1.8: Schematické znázornění výpočtu efektivní hodnoty u signálu děleného do bloků pro překrytí úseků signálu vybíraných okny $1/3$.

výpočtu se zkrátí z N na $2/3N$, tedy právě o jednu třetinu. Tím vzroste rozlišení v čase dané zmiňovanou dobou mezi dvěma okamžiky výpočtu.

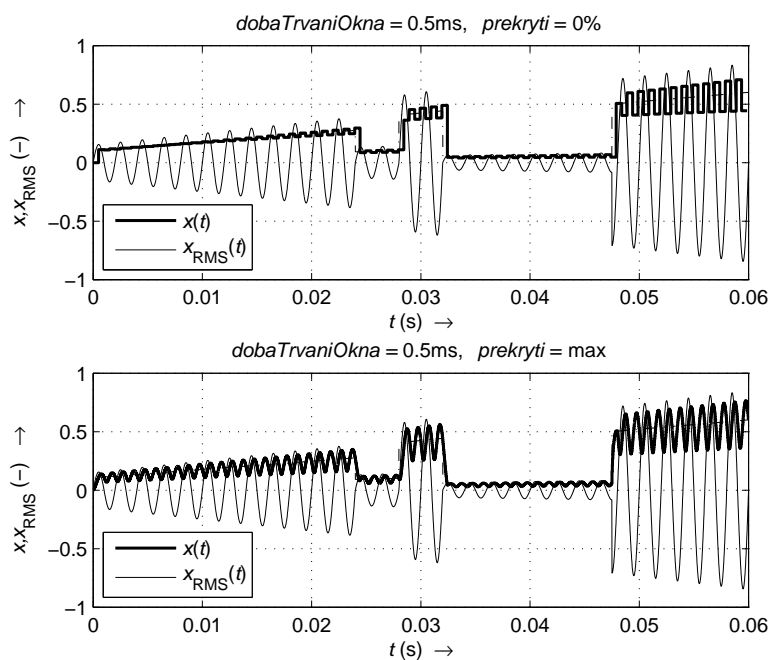
Jak je z obou obrázků patrné, na konci každého bloku signálu je nutné uložit poslední vzorky do paměti a připravit je (včetně výchozí pozice) pro první výpočty v následujícím bloku. Právě v tomto místě by při použití nekauzálního způsobu výpočtu vznikl problém s nedostupností prvních vzorků z následujícího bloku signálu. Při prvním spuštění je tento proces taktéž nutné zajistit. Jak je také z obrázků vidět, jedná se o $N-1$ blíže neurčených vzorků nesoucích informaci o počátečních podmínkách systému, nejběžněji ovšem nastavovaných na nulu.

1.2.2 Charakter výstupního signálu v závislosti na překrytí a délce oken

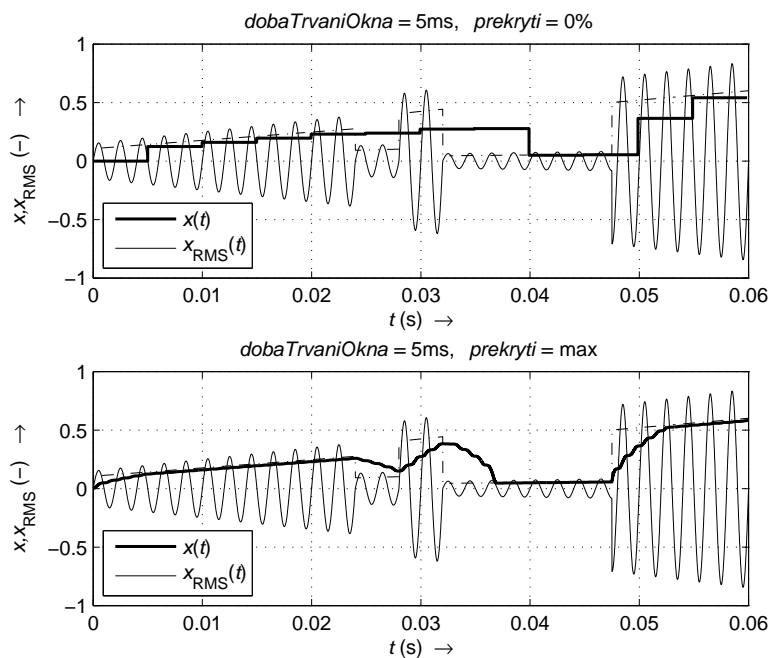
Podobně jako bylo u systémů popsaných v části 1.1 obtížné najít správné hodnoty parametrů t_{TAV} , respektive t_{AT} a t_{RT} , tak u výpočtu efektivní hodnoty signálu z matematické definice je problematické stanovit nejvhodnější délku okna N . Na následujících obrázcích 1.9, 1.10 a 1.11 je ukázán vliv délky okna a velikosti překrytí oken na signálu použitým i v části 1.1 obsahujícím pouze jednu harmonickou složku, pro kterou je volba délky okna N snadná. U reálného zvukového signálu je situace už zcela odlišná. Obecně by mohlo platit, že délka okna N by měla odpovídat periodě nejnižší kmitočtové složky svojí velikostí ještě významné.



Obr. 1.9: Zobrazení amplitudově modulovaného harmonického signálu o kmitočtu $f = 500\text{ Hz}$ a vypočítaného průběhu efektivní hodnoty pro délku okna odpovídající polovině periody vstupního signálu a pro nejmenší a největší velikost překrytí oken.



Obr. 1.10: Zobrazení amplitudově modulovaného harmonického signálu o kmitočtu $f = 500\text{ Hz}$ a vypočítaného průběhu efektivní hodnoty pro velmi krátké okno a pro nejmenší a největší velikost překrytí oken.



Obr. 1.11: Zobrazení amplitudově modulovaného harmonického signálu o kmitočtu $f = 500\text{ Hz}$ a vypočítaného průběhu efektivní hodnoty pro velmi dlouhé okno a pro nejmenší a největší velikost překrytí oken.

Aby byl zachován stejný způsob zobrazení obrázků jako v části 1.1 a přitom bylo patrné, že k výpočtu efektivní hodnoty signálu pro aktuální okamžik dochází jen v okamžicích daných koncem okna, byla pro vykreslení průběhu výstupního signálu zvolena metoda simulující techniku vzorkování *sample and hold*; výsledný průběh je tak po částech konstantní funkcí. Tato skutečnost je vidět na horních průbězích zmíněných tří obrázků, pro které byla zvolena nulová velikost překrytí oken, a jednotlivé okamžiky výpočtu jsou tak od sebe dostatečně vzdáleny a je možné je rozeznat. Ve spodních částech je vždy umístěn průběh pro nejvyšší možnou velikost překrytí, kdy k výpočtu výstupního vzorku dochází pro každý vzorek vstupní. Tato varianta je však při zpracování v reálném čase prakticky nepoužitelná.

Velká výhoda metody používající matematickou definici efektivní hodnoty je ta, že při optimální volbě délky okna může dojít k získání naprosto přesných hodnot. Tento jev je ukázán na obrázku 1.9. Podmínkou je délka okna odpovídající polovině periody vstupního harmonického signálu. Přesného výsledku by bylo samozřejmě dosaženo i při celočíselných násobcích této hodnoty, nicméně volit délku okna delší nežli je třeba, je zbytečné a navíc by tak byla snížena rychlost reakce systému na změny obálky vstupního signálu. Při porovnání horního a dolního průběhu je možné také pozorovat důsledek narůstajícího překrytí oken, kdy se ze skokových změn vypočítané efektivní hodnoty stávají změny plynulé. Tento důsledek je patrný i na obrázcích 1.10 a 1.11. Ovšem ne vždy je prospěšný.

Na obrázku 1.10 je vidět, že v případě příliš krátkého okna, kde již dochází k značnému rozkmitu hodnot výstupního signálu, má zvýšení velikosti překrytí oken naopak negativní vliv na výsledný průběh efektivní hodnoty vstupního signálu. Druhý extrémní případ je ukázán na obrázku 1.11, kde je naopak volena značná délka okna. Doba mezi jednotlivými okamžiky výpočtu výstupního vzorku je ovšem při takovémto nastavení bez překrývání oken natolik velká, že systém nedokáže reagovat na rychlé změny obálky vstupního signálu. Jak je vidět na spodní části obrázku, zvýšením velikosti překrytí oken se tato reakční doba sníží. Toto řešení však není vhodné, jelikož kombinace velmi dlouhého okna s maximální velikostí překrytí oken dosahuje největší možné výpočetní náročnosti použitého algoritmu. Na obou popísaných obrázcích dochází k obdobným jevům jako v části 1.1.1 na obrázku 1.3.

1.3 Doporučení ITU-R BS.1770-1

Dva nežádoucí jevy současné hudební i televizní či rozhlasové produkce vedly organizaci ITU-R k vytvoření volně širitelného doporučení BS.1770-1. Prvním z nich je nejednotnost úrovní hlasitosti zvukových děl, a to jak z pohledu jednotlivých produkčních společností, tak i žánru hudební nahrávky nebo druhu pořadu.

Druhým nežádoucím jevem je zkreslení zvukového signálu při přebuzení DA převodníku, k němuž může docházet důsledkem chybně vyhodnocené špičkové hodnoty signálu. U obecného analogového signálu není možné zaručit, že jeho špičková hodnota bude při vzorkování zachycena. Špičková hodnota navzorkovaného signálu (vzorek s nejvyšší absolutní hodnotou) je tedy obecně *menší nebo rovna* špičkové hodnotě původního analogového signálu. Nebezpečí „podečtení“ (under-read) hrozí zejména u přechodných dějů ve zvukovém signálu, kde vzrůstá podíl vyšších kmitočtových složek signálu. Velikost chyby, ke které v takových případech může dojít se pohybuje v řádu jednotek dB [3]. Při normalizaci signálu ke špičkové hodnotě je tak nutné dodržovat rezervu (headroom) alespoň 3 dB pod maximální úroveň 0 dBFS. Pokud není zajištěna dostatečná rezerva, může velikost skutečné špičky překročit hodnotu 0 dBFS, a při převodu do analogové oblasti tak dojít ke zmíněnému zkreslení signálu.

Výsledkem doporučení jsou dvě metody měření. Metoda měření úrovně hlasitosti a metoda měření skutečné špičkové hodnoty.

1.3.1 Metoda měření úrovně hlasitosti

Úvodem je vhodné zmínit, že doporučení BS.1770-1 je kritizováno za používání termínu „hlasitost“ (loudness) v případech, kde by měl být použit termín „úroveň hlasitosti“ (loudness level) [5]. Dále jsou podle [5] v BS.1770-1 použity symboly pro označení veličin a jednotek v rozporu se standardem ISO 80000-8:2007. Tato práce se přiklání k [5].

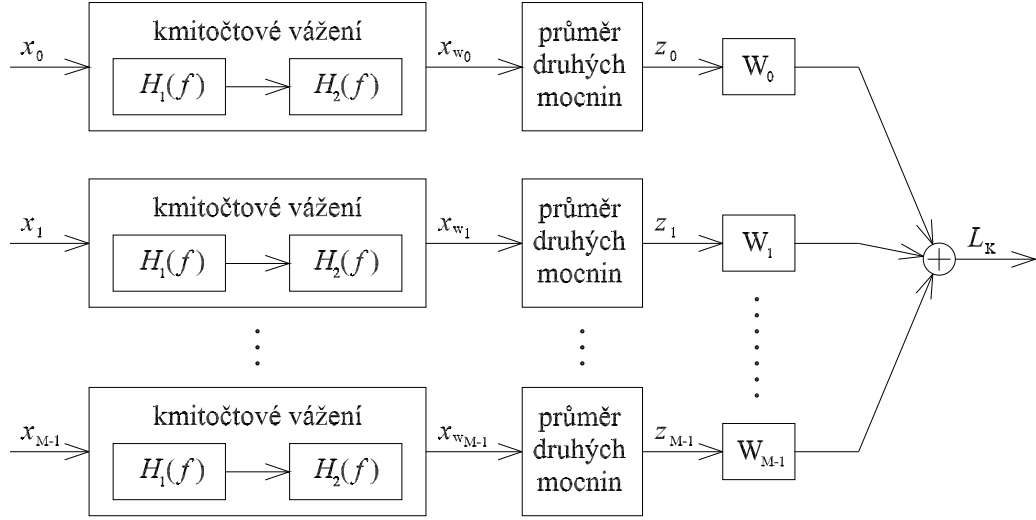
Cílem doporučení BS.1770-1 bylo navrhnout metodu, jejíž používání sjednotí postup při produkční práci a umožní objektivně nastavit výslednou hlasitost. Navržená metoda je použitelná pro obecný vícekanálový systém a má nízkou výpočetní náročnost. Zkráceně se označuje jako $Leq(R2LB)$ nebo $Leq(K)$.

Metoda měření $Leq(K)$ byla vybrána na základě výsledků subjektivních testů. Jejich princip spočíval v dorovnání hlasitosti posuzované nahrávky k hlasitosti referenční nahrávky. Získaná hodnota např. 4 LU (loudness unit) znamená, že posuzovaná nahrávka byla o 4 LU hlasitější než referenční. Referenční hodnotou bývá rozsah převodníku a jednotka úrovně hlasitosti vztažená k této hodnotě se označuje LUFS. Je důležité říci, že změna o 1 LU odpovídá změně o 1 dB.

V měření úrovně zvukového signálu doporučení BS.1770-1 zavádí zcela nový přístup. Rozdílem oproti měření efektivní hodnoty, popsáném v části 1.2, je zavedení kmitočtového vážení vstupního signálu zohledňující způsob vnímání hlasitosti zvuku lidským sluchem. Výsledek metody je tak označován jako objektivní odhad úrovně subjektivní hlasitosti [3].

Na obrázku 1.12 je znázorněno blokové schéma metody pro M -kanálový systém. Nejprve je prováděno kmitočtové vážení vstupního signálu a následně výpočet dle

vztahů (1.9) a (1.10). To vše pro každý kanál zvlášť.



Obr. 1.12: Blokové schéma metody měření úrovně hlasitosti dle BS.1770-1 [3].

Kmitočtové vážení vstupního signálu

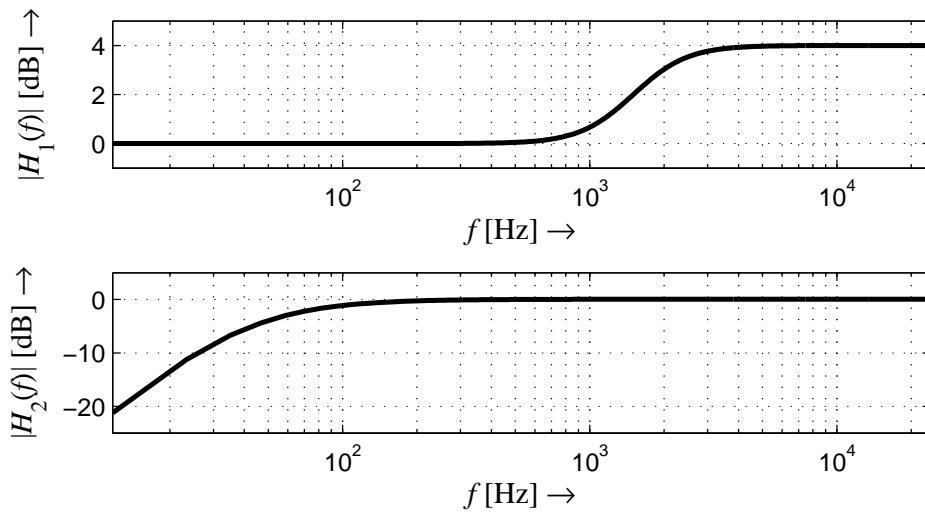
Kmitočtové vážení vstupního signálu je prováděno dvěma kmitočtovými filtry druhého řádu s nekonečnou impulsní odezvou. V prvním kroku je zohledněn akustický stín hlavy. Amplitudová kmitočtová charakteristika tohoto filtru je uvedena na obrázku 1.13 nahoře. Následně je použita dolní propust s amplitudovou kmitočtovou charakteristikou označovanou jako RLB (revised low-frequency B-curve) uvedenou na obrázku 1.13 dole. Doporučení BS.1770-1 udává přesné koeficienty filtrů pro vzorkovací kmitočet 48 kHz. Pro jiné vzorkovací kmitočty je nutné koeficienty přepočítat, aby výsledná kmitočtová charakteristika kopírovala původní v *absolutních* hodnotách kmitočtu, nikoli relativních jak by tomu bylo bez přepočítání. V tomto ohledu nebyla v práci vyvinuta žádná snaha.

Přenosové funkce filtrů jsou následující [3]

$$H_1(z) \doteq \frac{1,5351248596 - 2,69169618941z^{-1} + 1,1983928109z^{-2}}{1 - 1,6906592932z^{-1} + 0,73248077422z^{-2}}, \quad (1.8)$$

$$H_2(z) = \frac{1 - 2z^{-1} + 1z^{-2}}{1 - 1,99004745483398z^{-1} + 0,99007225036621z^{-2}},$$

kde $H_1(z)$ je přenosová funkce filtru zohledňující akustický stín hlavy a $H_2(z)$ je přenosová funkce filtru s amplitudovou kmitočtovou charakteristikou RLB. Koeficienty přenosové funkce $H_1(z)$ nejsou uvedeny s plnou přesností.



Obr. 1.13: Amplitudová kmitočtová charakteristika filtru zohledňujícího akustický stín hlavy (nahore) a filtru dle váhovací křivky RLB (dole) [3].

Výpočet úrovně hlasitosti

Po provedení kmitočtového vážení je nejprve proveden výpočet aritmetického průměru druhých mocnin signálu [3]

$$z_i = \frac{1}{N} \sum_{n=0}^{N-1} x_w^2[n], \quad (1.9)$$

kde $x_w[n]$ je vzorek kmitočtově váženého vstupního signálu v diskretním čase n , N je počet vzorků, nad nimiž je prováděn výpočet, a z_i je velikost aritmetického průměru druhých mocnin signálu $x_w[n]$ v čase od $n = 0$ do $n = N - 1$ pro kanál i .

Jakmile jsou známy výsledky z_i pro všechny kanály, je možné vypočítat výslednou úroveň hlasitosti L_K [LUFS] dle [3]

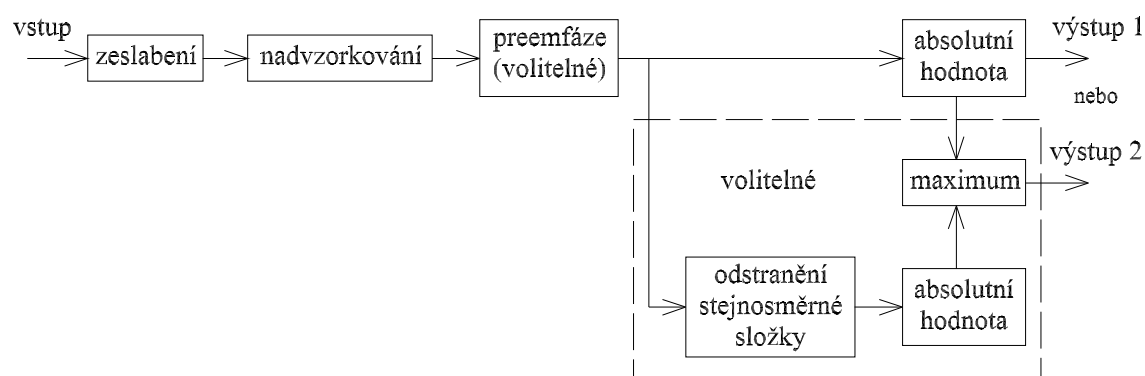
$$L_K = -0,691 + 10 \log_{10} \sum_{i=0}^{M-1} W_i \cdot z_i, \quad (1.10)$$

kde W_i je váhovací koeficient pro kanál i a M je počet kanálů. Pro přední levý, pravý a prostřední kanál stanovuje doporučení BS.1770-1 váhovací koeficienty $W_i = 1$ a pro levý a pravý zadní kanál je $W_i = 1,41$.

1.3.2 Metoda měření skutečné špičkové hodnoty

Na začátku části 1.3 byl popsán důvod proč je nutné znát *skutečnou* špičkovou hodnotu diskretního signálu (true-peak). Se vzrůstajícím kmitočtem zvukového signálu se zkracuje jeho perioda a přibližuje se tak periodě vzorkovací. Zároveň s tím

narůstá i možná chyba podečtením. Řešením je nadvzorkování vstupního zvukového signálu. Zkrátí se tím perioda vzorkování, ale nejvyšší kmitočet zvukového signálu zůstane zachován. Nejvyšší možná velikost chyby podečtením pak klesá s rostoucím poměrem nadvzorkování. Po zvýšení vzorkovacího kmitočtu na čtyřnásobek (např. z $f_{vz} = 48$ kHz na $f_{vz} = 192$ kHz) se nejvyšší možná chyba podečtením na kmitočtu rovném čtvrtině původního vzorkovacího kmitočtu sníží ze 3 dB na 0,169 dB [3]. Na kmitočtu rovném polovině původního vzorkovacího kmitočtu by tomu bylo z ∞ dB na 0,688 dB [3].



Obr. 1.14: Blokové schéma metody měření skutečné špičkové hodnoty dle doporučení ITU-R BS.1770-1 [3].

Blokové schéma je znázorněno na obrázku 1.14. Blok **zeslabení** je nutné zařadit pouze pokud jsou následující výpočty prováděny v pevné řádové čárce. Útlum tohoto bloku je 12 dB [3]. Následuje blok **nadvzorkování**. Hodnota vzorkovacího kmitočtu, při které se měření skutečné špičkové hodnoty provádí musí být nejméně 192 kHz. Vstupní zvukový signál se vzorkovacím kmitočtem 48 kHz bude nadvzorkován čtyřikrát. Pouze dvakrát bude nadvzorkován signál se vzorkovacím kmitočtem 96 kHz. Následující volitelný blok **preemfáze** zesiluje nejvyšší kmitočty signálu, které jsou náchylnější k chybám v důsledku podečtení. Na závěr je zjištěna absolutní hodnota aktuálního vzorku signálu, a to buď se simulací odstranění stejnosměrné složky nebo bez ní. Výstupem je vždy jen jedna hodnota, a to ta největší.

Je-li vnitřní zpracování prováděno při vzorkovacím kmitočtu nejméně 192 kHz, může být pro označení úrovně změřené špičkové hodnoty použita jednotka dBTP.

1.4 Srovnání uvedených metod měření

Metody měření odhadu špičkové a efektivní hodnoty signálu uvedené v části 1.1 jsou v principu totožné. Poskytují podobné rozhraní pro nastavení svých vlastností

ve formě jedné či dvou časových konstant určujících dobu náběhu a dobu poklesu systému a i jejich výsledky jsou kvalitativně srovnatelné. Jejich nejdůležitějším rysem pro porovnání s metodou popsanou v části 1.2 je konstantní a nízká výpočetní náročnost – vlastnost vhodná pro použití systému ke zpracování signálu v reálném čase. Ta je u metody získávající efektivní hodnotu signálu výpočtem z matematické definice závislá na překrytí oken a v případě nenulové velikosti překrytí i na délce okna. Nicméně při nulovém nebo velmi malém překrytí oken by mohla být i menší než u metod používajících plovoucí odhad. Všechny tři zmíněné metody jsou citlivé na správné nastavení, z nichž v největší míře se tato nevýhoda projevuje u metody měření odhadu špičkové hodnoty, kde se výsledek vlivem nevyváženého nastavení doby náběhu a poklesu značně a dlouhodobě liší od správné hodnoty (viz obr. 1.6).

Výpočetní náročnost metody měření úrovně hlasitosti je oproti výpočtu efektivní hodnoty z matematické definice vyšší, jelikož je nutné provádět kmitočtovou filtraci vstupního signálu. Metodu měření úrovně hlasitosti lze také použít pro zpracování po úsecích. Pro výpočetní náročnost vzhledem k délce oken a velikosti jejich překrytí pak platí to stejné jako u metody výpočtu efektivní hodnoty z matematické definice. Podstatným rozdílem metody měření úrovně hlasitosti od ostatních popsaných metod je význam výstupní hodnoty. Nejedná se o parametr elektrického signálu jakým je efektivní nebo špičková hodnota, ale akustický parametr objektivně popisující lidský vjem daného elektrického signálu.

Metoda měření skutečné špičkové hodnoty jednoznačně převyšuje metodu měření odhadu špičkové hodnoty ve spolehlivosti. Její výsledek navíc není pouze odhadem, ale hodnotou s přesně definovanou nejvyšší možnou chybou v porovnání s původním analogovým signálem. Výpočetní náročnost této metody je však oproti metodě měření odhadu vyšší a roste úměrně poměru nadvzorkování L . Značný vliv na výpočetní náročnost má řád použitých filtrů.

2 DYNAMICKÉ PROCESORY

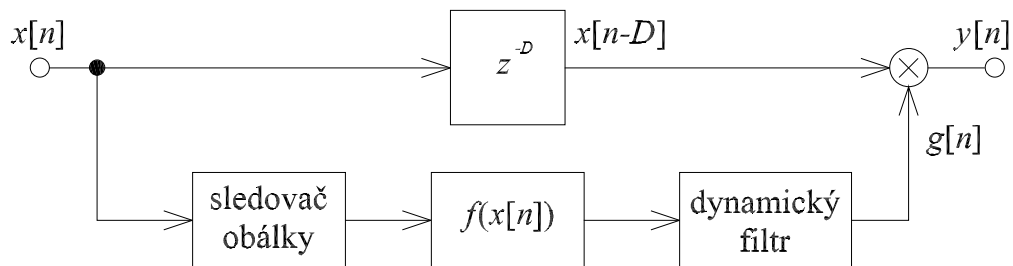
Úvodem této kapitoly by bylo vhodné od sebe oddělit dvě různé oblasti práce se zvukovými efekty obecně. Jsou jimi vytvoření vlastního efektu (ať ve formě fyzického zařízení či počítačového programu) a jeho používání a nastavení pro konkrétní vstupní signál. Tyto oblasti je možné nazvat inženýrskou a hudební; požadavky na orientaci v nich jsou od sebe ovšem značně odlišné. U běžně používaných kytarových efektů jimiž jsou například distortion, overdrive, chorus, delay a další není zmiňované nastavení nijak obtížné, přičemž je důsledek provedené změny ve výstupním signálu jasně patrný. Opačně však tomu je například právě u dynamických procesorů a jejich používání v nahrávacích studiích při mixáži výsledné zvukové stopy. Studioví technici, provádějící takovou činnost, musejí mít značné zkušenosti, aby získali požadovaný výsledek.

Již v úvodu této práce bylo zmíněno, že dynamické procesory ovlivňují vstupní signál určitým způsobem v závislosti na signálu řídicím, jenž je většinou odvozen právě od vstupního signálu. Onen způsob je pouhé násobení vzorku vstupního signálu vzorkem signálu řídicího, což si je možné představit jako zařazení zesilovače do signálové cesty, jehož přenos je nastavován řídicím signálem. Ne všechny dynamické procesory jsou řízeny stejnými parametry signálu. Limitery potřebují ke své činnosti sledovat špičkovou hodnotu signálu, kompresory, expandery a šumové brány jsou řízeny dle efektivní hodnoty [8, 9]. Z hlediska vlivu na vstupní signál jako na celek je řídicím signálem ovládán tvar obálky vstupního signálu. Výstupním signálem dynamických procesorů je tak signál vstupní s pozměněnou dynamikou.

Konkrétní typ dynamického procesoru (například kompresor, expander, limiter či šumová brána) je určen tvarem převodní charakteristiky, která zpracovává měřený parametr vstupního signálu (například špičkovou či efektivní hodnotu), a jejíž výstup po jisté úpravě slouží jako řídicí signál daného systému. Tvar převodní charakteristiky je obecně nelineární.

Blokové schéma platící pro každý dynamický procesor je uvedeno na obrázku 2.1. Systém obsahuje dvě větve, *přímou* (direct path) a *řídicí* (side chain path). Přímou větví prochází vstupní signál $x[n]$ beze změny až na zpoždění dané nezápornou celočíselnou proměnnou D . Obecně se tedy předpokládá, že doba výpočtu vzorku řídicího signálu $g[n]$ je větší než vzorkovací perioda. Aby systém reagoval vždy na vzorek $x[n - D]$ způsobem zjištěným výpočtem právě na základě tohoto konkrétního vzorku a aby tak byl zachován pár vzorků vstupního a řídicího signálu $x[n]$ a $g[n]$ (vzorek $g[n]$ je počítán právě na základě vzorku $x[n]$), je nutné, aby velikost zpoždění D byla rovna počtu dokončených vzorkovacích period za dobu výpočtu vzorku $g[n]$.

V řídicí větvi je jako první obsažen systém pro měření požadovaného parametru vstupního signálu (viz. kapitola 1), který je zastoupen blokem *sledovač obálky* a dále



Obr. 2.1: Obecné schéma dynamického procesoru [7, 8].

převodní charakteristika naznačená vztahem $f(x[n])$ a *dynamický filtr* produkující výsledný průběh řídicího signálu $g[n]$. Poslední dva zmíněné bloky jsou popsány v následujících částech 2.1 a 2.2.

2.1 Převodní charakteristika

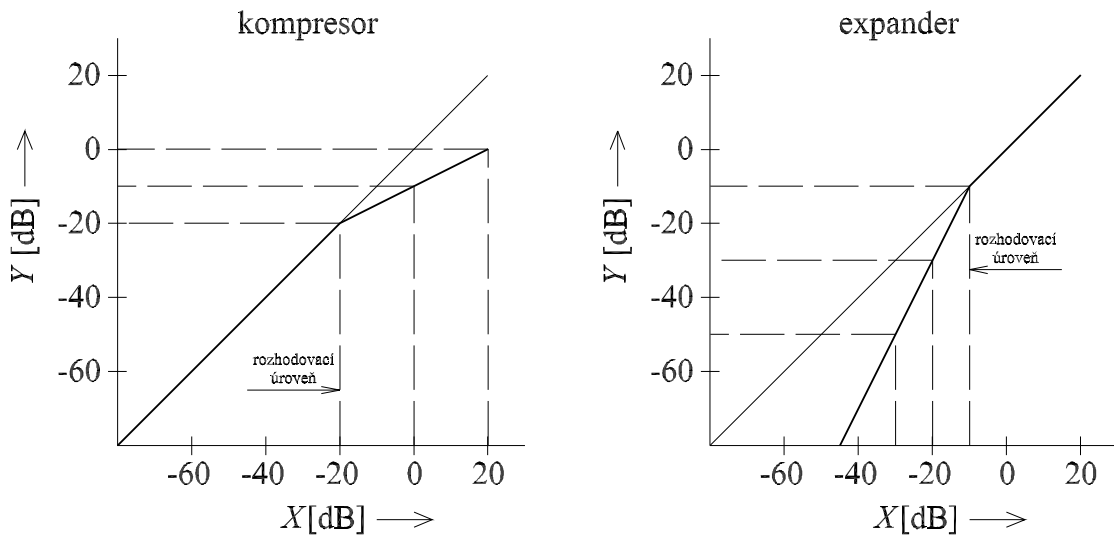
Vstupním signálem bloku převodní charakteristiky je tedy změřená špičková či efektivní hodnota. Převodní charakteristika bývá zpravidla definována v logaritmickém měřítku, a proto je nutné převést vstupní signál do této oblasti. Funkční hodnota převodní charakteristiky odpovídá po průchodu dynamickým filtrem řídicímu signálu $g[n]$. Při zobrazení převodní charakteristiky dynamického procesoru je však zvykem uvádět na svislé ose úroveň odpovídající výstupnímu signálu Y .

Na obrázcích 2.2, 2.3 a 2.4 jsou uvedeny převodní charakteristiky různých systémů, které budou podrobně popsány dále. Je vhodné zmínit, že u každé uvedené převodní charakteristiky je vždy zakreslena tenká plná čára procházející bodem $[0; 0]$ a mající strmost 1. Ta popisuje převodní charakteristiku systému, při jehož použití by dynamika vstupního signálu $x[n]$ zůstala zachována. Jedinečnost každého dynamického procesoru je určena oblastí převodní charakteristiky, ve které se liší od lineární převodní charakteristiky.

2.1.1 Dělení dle tvaru převodní charakteristiky

Současným trendem v oblasti mixáže hudební nahrávky je docílení nejvyšší možné hlasitosti, čehož však není možné dosáhnout bez *snížení* dynamiky signálu. Cílem je snaha co nejvíce uzpůsobit charakter nahrávky vlastnostem přenosového řetězce, vlastnostem koncového zařízení (například rozhlasový vysílač) a hlučnosti prostředí, ve kterém je hudba v současnosti nejčastěji reprodukována.

Důvodem k používání systémů *zvyšujících* dynamický rozsah vstupního signálu je potlačení šumu. Tyto systémy slabé signály zeslabují a ostatní zachovávají v původní podobě. Signály nejslabší jsou odstraněny zcela.



Obr. 2.2: Převodní charakteristiky kompresoru a expanderu při poměru komprese a expanze 2 : 1 [7, 8].

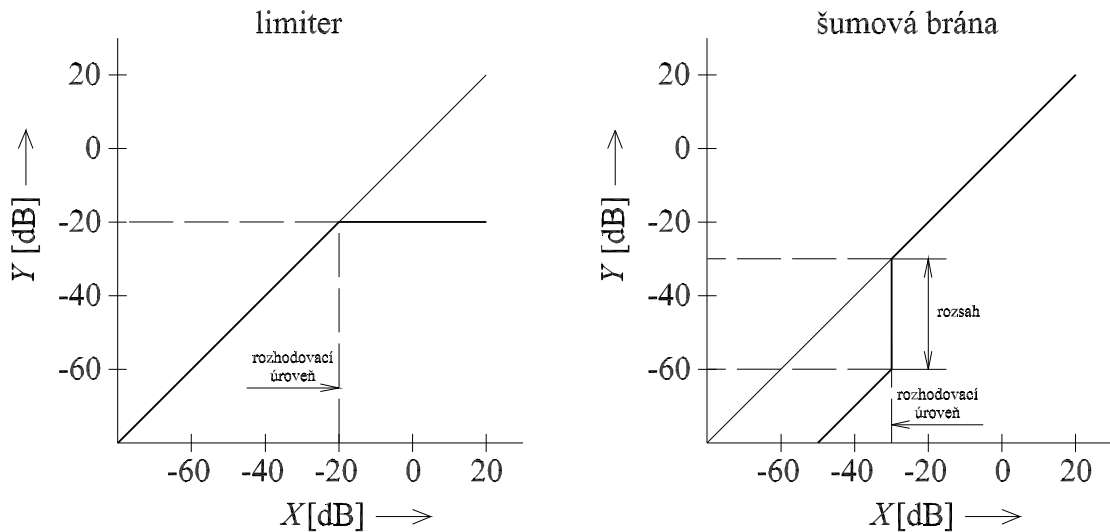
Dle tvaru převodní charakteristiky je možné rozeznávat čtyři základní typy dynamických procesorů. Jsou jimi kompresor, expander, limiter a šumová brána. Vhodným složením jednotlivých převodních charakteristik těchto jednoduchých systémů je možné získat systém kombinovaný.

Nastavení každého systému je možné provádět posunem rozhodovací úrovně, která definuje místo, v němž dochází ke zlomu převodní charakteristiky. U kompresoru a expanderu je také možné ovlivnit strmost převodní charakteristiky v oblasti, ve které se liší od převodní charakteristiky lineárního systému a určit tím poměr komprese respektive expanze.

Kompresor *snižuje* dynamický rozsah vstupního signálu. Signály s úrovní vyšší než rozhodovací jsou zeslabeny, a to v nastaveném poměru komprese. Signály s úrovní nižší než rozhodovací jsou ponechány beze změny (obrázek 2.2).

Expander *zvysuje* dynamický rozsah vstupního signálu. Signály s úrovní nižší než rozhodovací jsou zeslabeny, a to v nastaveném poměru expanze. Signály s úrovní vyšší než rozhodovací jsou ponechány beze změny (obrázek 2.2).

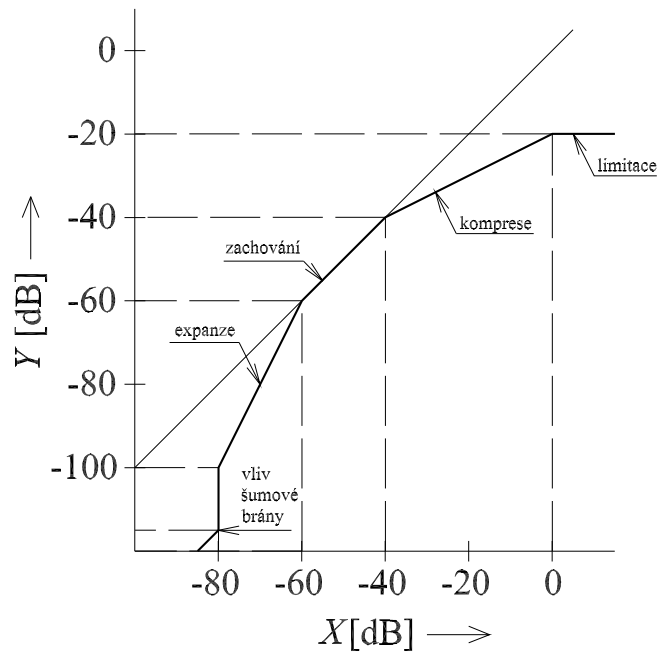
Limiter *snižuje* dynamický rozsah vstupního signálu. Jedná se o krajní případ kompresoru. Signály s úrovní vyšší než rozhodovací jsou zeslabeny, a to vždy v poměru komprese $\infty : 1$ (obrázek 2.3). Teoreticky, úroveň výstupního signálu limiteru nikdy nemůže přesáhnout nastavenou maximální funkční hodnotu převodní charakteristiky. V praxi ovšem může při rychlých změnách obálky vstupního signálu dojít k určitým překmitům výstupního signálu nad stanovenou hranici způsobených nenulovou setrvačností systému. Limiter bývá používán i k ochraně vstupů zařízení.



Obr. 2.3: Převodní charakteristiky limiteru a šumové brány [7, 8].

Šumová brána zvyšuje dynamický rozsah vstupního signálu. Literatura [7], z níž byl obrázek 2.3 převzat, popisuje funkci šumové brány jako zeslabení signálu s úrovní nižší než rozhodovací, a to o velikost danou parametrem označeným jako *rozsah*, přičemž signál s úrovní vyšší než rozhodovací zůstane nezměněn. U popisu šumové brány v literatuře [8] ve vztahu k [7] je velikost parametru „rozsah“ rovna nekonečnu. Veškeré vstupní vzorky mající úroveň nižší než rozhodovací jsou tak v tomto případě nulovány (v absolutní míře – nikoli logaritmické). Nicméně v reálných digitálních efektech pro úpravu dynamiky je používán popis dle [7]. Účelem šumové brány je potlačení šumu a bývá používána i ke zkrácení doby dozvuku signálu.

Kombinovaný systém spojuje do jednoho celku více jednoduchých systémů popisovaných výše, jejichž zastoupení v něm záleží na aplikaci, pro kterou je systém použit. Důležité však je dodržovat logické navazování zlomových bodů převodní charakteristiky. Je proto vhodné definovat uživatelské rozhraní, které redukuje počet nastavovaných parametrů a zavádí parametry nové – vhodnější, ze kterých jsou původní parametry zpětně získány výpočtem, a přitom zajistí jejich bezpečné nastavení. Těmito abstraktními parametry jsou například šířka a posunutí oblasti odpovídající lineárnímu systému či poměry komprese a expanze. Některé z původních parametrů (například rozhodovací úroveň limiteru) mohou být nastaveny pevně. Převodní charakteristika tohoto systému je zobrazena na obrázku 2.4.



Obr. 2.4: Převodní charakteristika kombinovaného systému [8, 7].

2.2 Dynamický filtr

K dokončení popisu veškerých částí dynamického procesoru již zbývá pouze poslední blok řídicí větve (viz. obrázek 2.1). Je jím dynamický filtr. Jednoduše jej lze charakterizovat jako integrátor s nastavitelnou dobou náběhu i poklesu. Obecně lze spatřovat jeho účel ve vyhlazení průběhu výstupního signálu bloku převodní charakteristiky a získání tak řídicího signálu $g[n]$. Konkrétní důvody jeho použití jsou dva, a to eliminace kolísavosti měřené efektivní či špičkové hodnoty či jiného parametru vstupního signálu $x[n]$ a jí způsobeného častého skokového přechodu mezi různými oblastmi převodní charakteristiky znatelného zejména pro hodnoty úrovně změřeného parametru vstupního signálu blízké úrovni rozhodovací. Druhý uvedený důvod je obzvláště kritický u šumové brány, kde by za popsanych okolností a současném nepoužití dynamického filtru mohlo neustále docházet k přepínání mezi dvěma mezními stavy propustit/nepropustit. Diferenční rovnice systému je následující [8]

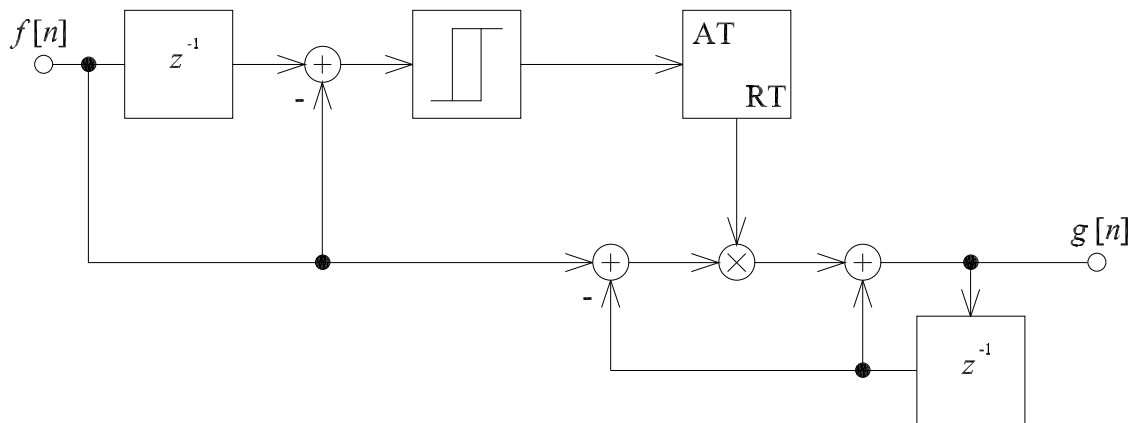
$$g[n] = (1 - k_{AR})g[n-1] + k_{AR}f[n], \quad (2.1)$$

kde $f[n]$ je vstupní signál filtru. Pro koeficient k_{AR} platí

$$\begin{aligned} k_{AR} &= AT \quad \text{pro } f[n] - f[n-1] > 0 \quad \text{nebo} \\ k_{AR} &= RT \quad \text{pro } f[n] - f[n-1] < 0, \end{aligned} \quad (2.2)$$

kde AT a RT jsou parametry se stejným významem jako v části 1.1.2, definované vztahem (1.3).

Blokové schéma dynamického filtru na obrázku 2.5 je možné rozdělit na řídicí a přímou větev obdobně jako u obrázku 2.1. Řídicí větev v tomto případě provádí jednoduchou analýzu vstupního signálu k získání informace o tom, který z parametrů AT a RT použít k výpočtu v přímé větvi. Do řídicí větve je k tomuto účelu mimo jiné umístěn blok hystereze, který zabraňuje příliš častému nežádoucímu přepínání mezi zmíněnými parametry AT a RT na základě hodnoty rozdílu současného a předchozího vstupního vzorku (literatura [9] uvádí šířku hystereze jako „malou“). V přímé větvi je pouze zrealizován jednoduchý integrátor, který vyhlazuje průběh signálu $f[n]$ s časovou konstantou dle aktuálně použitého parametru AT či RT . Výstupem je již řídicí signál $g[n]$ přímo ovlivňující vstupní signál dynamického procesoru $x[n]$.



Obr. 2.5: Blokové schéma dynamického filtru [8, 9]

3 POPIS IMPLEMENTACE

V této kapitole je popsána implementace všech dílčích částí dynamického procesoru i jejich spojení do výsledného funkčního celku.

3.1 Implementace metod měření

3.1.1 Implementace metody měření efektivní hodnoty pomocí výpočtu z matematické definice

Pro účely doplnění textu části 1.2 obrázky byl v prostředí Matlab vytvořen jednoduchý skript implementující metodu měření efektivní hodnoty signálu pomocí výpočtu z matematické definice dle vztahu (1.7), který poskytuje rozhraní pro nastavení délky zpracovávaného úseku a velikosti překrytí oken. Pro správnou interpretaci časové osy je nutné zadat jako jeden ze vstupních parametrů vzorkovací kmitočet vstupního signálu. Výstupem je vektor efektivních hodnot a vektor časové osy. Nicméně algoritmus pracuje pouze s jedním blokem dat.

```
function [y t] = rmsWin(x, winDur, overlape, fs)
% x.....vstupní signál
% winDur.....délka okna [ms]
% overlape...<0;1>
% fs.....vzorkovací kmitočet [Hz]
% y.....výstupní signál
% t.....časová osa

% převod délky okna v ms na počet vzorků
winLen = floor( fs * winDur / 1000);

% počet vzorků signálu nepřekrytých okny
% - krok, po kterém se jednotlivá okna posouvají
nooverlapedSamples = floor( (1 - overlape) * winLen);

j = 1;
% výpočet efektivní hodnoty pro každé okno přes celý vstupní signál
for i = winLen:nooverlapedSamples:length(x)
    y(j) = sqrt( sum( x(i-winLen+1:i).^2) / winLen); % RMS
    t(j) = (i-winLen) / fs; % časová osa pro správné vykreslování
    j = j+1;
end
```

3.1.2 Implementace základních metod měření odhadu efektivní a špičkové hodnoty

Základní metody měření odhadu efektivní a špičkové hodnoty byly implementovány dle textu části 1.1 čerpajícího z [9]. V jazyce C++ byla vytvořena základní třída poskytující funkce pro nastavení časových konstant a práci se sdílenými proměnnými. Od této třídy byly odvozeny třídy pro měření efektivní a špičkové hodnoty. Ty svým vlastním způsobem implementují virtuální metodu základní třídy, která vrací odhad měřeného parametru na základě vzorku vstupního signálu a stavu systému.

3.1.3 Implementace metody měření úrovně hlasitosti dle doporučení ITU-R BS.1770-1

Metoda měření úrovně hlasitosti dle BS.1770-1 byla implementována jako třída v jazyce C++. Přesně dle doporučení BS.1770-1 pracuje pouze se signálem o vzorkovacím kmitočtu 48 kHz (důvod viz část 1.3.1). Implementována byla pouze jednokanálová metoda, kterou bude možné využít pro implementaci metody vícekanálové.

Řízení výpočtu při blokovém zpracování

Aby bylo možné metodu používat nezávisle na velikosti bloku dat, po kterých probíhá zpracování u technologie VST, bylo nutné zavést algoritmus, který samotný výpočet měřeného parametru řídí. Blokové zpracování se poté stává pro metodu měření transparentním.

Metoda měření úrovně hlasitosti byla navržena pro zpracování po úsecích popsaném v části 1.2.1. Jejimi parametry jsou tedy délka okna a velikost překrytí oken. Řídící algoritmus musí společně s délkou bloku tyto parametry znát. Sám pak určuje konkrétní část bloku dat, která bude zahrnuta do patřičného (mezi)výpočtu. Pokud je pro výpočet měřeného parametru pro dané okno třeba více bloků dat, je výpočet rozdělen do částí, které se provádí dle dostupnosti dat. Není tak třeba ukládat vzorky signálu aktuálního bloku do paměti a přitom je zatížení procesoru v čase rozloženo rovnoměrně. Mezivýpočty jsou prováděny dle vztahu (1.9) v té úpravě, že součet druhých mocnin neprobíhá přes celé okno, ale jen přes část dat, která je v tu chvíli dostupná. Postupně se mezivýsledky sčítají a výsledná úroveň hlasitosti pro dané okno je počítána z celkové hodnoty dle vztahu (1.10).

Pro popsané výpočty pro dané okno používá řídicí algoritmus instanci třídy měření úrovně hlasitosti, která vzniká se začátkem a zaniká s koncem daného okna.

Kmitočtové vážení (filtrace) vstupního signálu

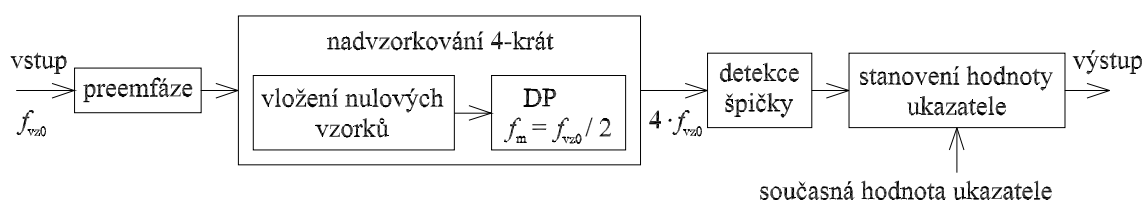
Jak bylo vidět na blokovém schématu 1.12, před před výše popsané výpočty je zařazena kmitočtová filtrace vstupního signálu. Zde není nutné ani vhodné postupovat v závislosti na délce okna a velikosti překrytí oken. Je tak provedena kmitočtová filtrace celého bloku dat najednou. Dva dílčí filtry s danými kmitočtovými charakteristikami (viz obr. 1.13) jsou řazeny za sebou. Každý z nich si pamatuje svůj stav, aby byla zaručena plynulá návaznost jednotlivých bloků dat. Oba filtry byly implementovány ve druhé kanonické formě, která je doporučena v [3].

3.1.4 Implementace metody měření skutečné špičkové hodnoty dle doporučení ITU-R BS.1770-1

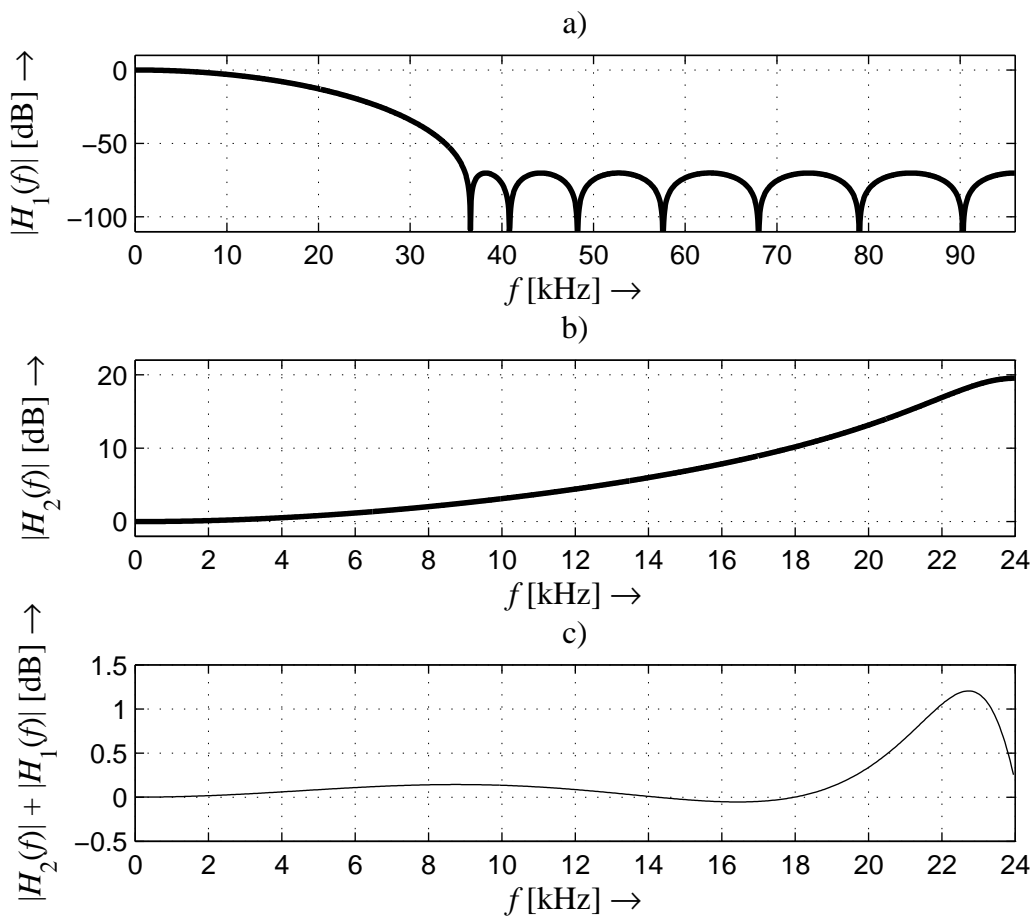
Metoda měření skutečné špičkové hodnoty dle BS.1770-1 byla implementována jako třída v jazyce C++. Na obrázku 3.1 je vidět blokové schéma znázorňující konkrétní způsob implementace metody měření skutečné špičkové hodnoty popsané v části 1.3.2. Schéma je dále postupně popisováno.

Nadvzorkování

Nadvzorkování je obecně prováděno vložení $L - 1$ nulových vzorků za každý vzorek původního signálu se vzorkovacím kmitočtem f_{vz0} , kde L je celé číslo určující poměr nadvzorkování. Dále je prováděna kmitočtová filtrace signálu dolní propustí s mezním kmitočtem $f_m = f_{vz0}/2$. Vzorkovací kmitočet nadvzorkovaného signálu je $f_{vz1} = Lf_{vz0}$. Charakter výsledného signálu zcela závisí na kmitočtové charakteristice použitého filtru. Byla upřednostněna věrná rekonstrukce původního signálu bez vzniku aliasingu. Při návrhu interpolačního filtru bylo dosaženo řádu 14. Jeho amplitudovou kmitočtovou charakteristiku je vidět na obrázku 3.2 v části a). Poměr nadvzorkování byl zvolen $L = 4$. Pro signál se vzorkovacím kmitočtem $f_{vz0} = 48$ kHz je tak splněna podmínka pro měření skutečné špičkové hodnoty při vzorkovacím kmitočtu nejméně 192 kHz.



Obr. 3.1: Blokové schéma implementace metody měření skutečné špičkové hodnoty dle doporučení ITU-R BS.1770-1.



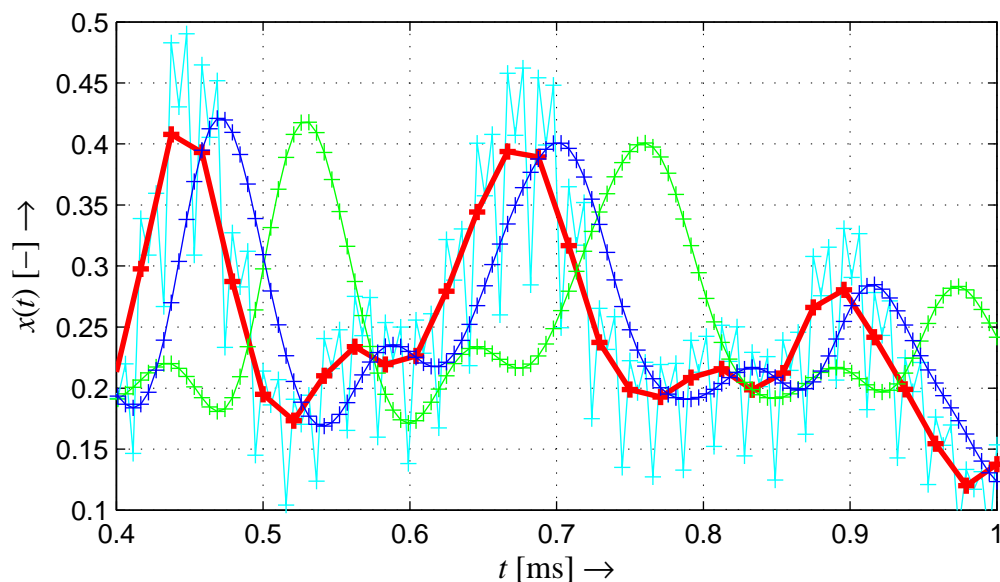
Obr. 3.2: Amplitudové kmitočtové charakteristiky interpolačního filtru (a)) a filtru preemfáze (b)) a chyba kompenzace útlumu interpolačního filtru pro kmitočty od 0 Hz do 24 kHz (c)).

Preemfáze

Aby byl vykompenzován útlum interpolačního filtru v propustném pásmu, který na kmitočtu $f_{vz0}/2$ dosahuje hodnoty $-19,4$ dB, byl navržen filtr, na obrázku 3.1 zastoupený blokem **preemfáze**. Jeho amplitudovou kmitočtovou charakteristiku je vidět na obrázku 3.2 v části b). Část c) znázorňuje velikost chyby výsledné kompenzace. Pro dané použití je vhodnější, pokud je výsledný signál po průchodu oběma filtry spíše zesílený. Sníží se tím možná chyba podečtením. Filtr je typu IIR a prvního řádu. Vzhledem k tomu, že výpočet probíhá v plovoucí řádové čárce, zařazení zesilujícího filtru na první místo sériového zapojení článků není závadou.

Na obrázku 3.3 je výsledek popsaného způsobu nadvzorkování porovnán s výsledky ideálního rekonstrukčního filtru získaného pomocí funkce `interp` v prostředí

Matlab a filtru navrženého s požadavkem na nízký řád. Řád ideálního filtru je 32, řád druhého navrženého filtru je 6. Na zvolený způsob nadvzorkování je možné pohlížet jako na kompromis mezi těmito krajními možnostmi. Zejména vzhledem k nízké výpočetní náročnosti, by bylo vhodné zkusit interpolaci filtrem s nízkým řádem.



Obr. 3.3: Srovnání původního signálu o vzorkovacím kmitočtu $f_{vz0} = 48 \text{ kHz}$ (červený) se signálem nadvzorkovaným zvoleným způsobem (modrý) a se signálem bez preemfáze – pouze interpolované ideálním filtrem (zelený) a filtrem s nízkým řádem (tyrkysový).

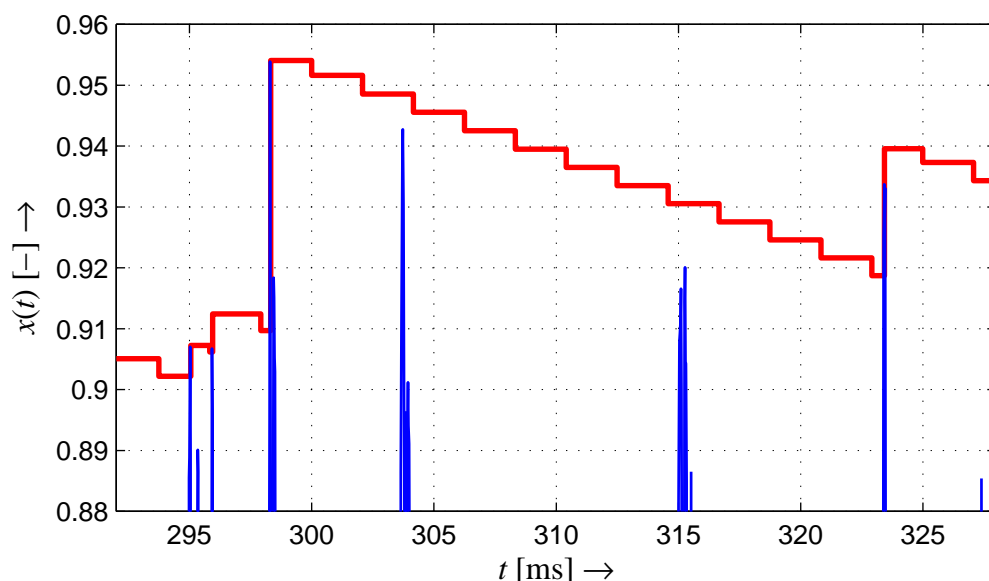
Detekce špičky v signálu a stanovení hodnoty ukazatele

Vzhledem k vysoké hodnotě vzorkovacího kmitočtu, při kterém měření skutečné špičkové hodnoty probíhá, není vhodné aktualizovat hodnotu ukazatele měřidla s každým vzorkem signálu. Pokud by aktualizace probíhala s každým vzorkem signálu, četnost přenosu informací z místa měření na místo zobrazování by tvořilo značnou zátěž pro procesor.

Implementovaný algoritmus tak vnitřně simuluje průběh ukazatele měřidla, a to včetně doby poklesu, a přenáší informace do místa zobrazování pouze pokud je detekována špička s hodnotou větší než stávající hodnota ukazatele. Aby nedocházelo k „zamrnutí“ ukazatele na jedné hodnotě v případě kdy není výše popsaným způsobem detekována žádná další špička, aktualizuje se hodnota ukazatele alespoň na konci každého zpracovávaného bloku.

Na obrázku 3.4 je zobrazen průběh ukazatele měřidla společně se vstupním signálem s původním vzorkovacím kmitočtem. V levé části jsou vidět dvě špičky signálu, které vedou k aktualizaci ukazatele v rámci jednoho bloku dat. Následně se vyskytne lokálně výrazně největší špička a ukazatel poté přes několik bloků pouze exponenciálně klesá. Rychlost poklesu byla stanovena na 20 dB za 1,5 s [7]. Přepočtená na hodnotu odpovídající jednomu vzorku při daném vzorkovacím kmitočtu včetně převodu z logaritmické míry, byl proveden dle vztahu 3.1

$$k = 10^{[-20/(1,5f_{vz1})]/20}. \quad (3.1)$$



Obr. 3.4: Průběhu ukazatele skutečné špičkové hodnoty signálu (červeně) a průběh vstupního zvukového signálu s původním vzorkovacím kmitočtem v oblasti špičkových hodnot (modře).

3.2 Implementace převodní charakteristiky

Převodní charakteristika byla implementována jako třída v jazyce C++. Byla vytvořena dvě rozhraní pro přístup k metodám třídy popisující převodní charakteristiku, a to uživatelské a funkční, sloužící pro modelaci obecné neklesající funkce, respektive pro získání její funkční hodnoty.

3.2.1 Uživatelské rozhraní a jeho vnitřní implementace

Průběh převodní funkce je tvořen z jedné či více křivek, jejichž tvar je dán polohou kontrolních bodů náležících k dané křivce. Uživatelské rozhraní poskytuje veškeré

funkce potřebné k pohodlnému tvarování převodní charakteristiky. Jsou jimi:

- pohyb kontrolními body,
- nastavení módu křivky,
- přidání křivky,
- odebrání křivky.

Pohyb kontrolními body

Vzhledem k tomu, že se jedná o modelaci *křivkami*, je bezpodmínečně nutné dodatečně zajistit, aby tvar křivky zachovával charakter matematické *funkce*; tedy pro každou hodnotu z definičního oboru funkce musí být dána právě jedna hodnota z oboru hodnot funkce. Zadáním této práce je však stanovena i nutnost dodržení *neklesajícího* průběhu funkce. Splnění těchto dvou podmínek je zajištěno umožněním pohybu kontrolních bodů pouze v určité (obdelníkové) oblasti dané souřadnicemi patřičných sousedních kontrolních bodů. Dodržení první podmínky (obecné matematické funkce) je dosaženo kontrolou souřadnic v ose definičního oboru. Obdobně dodržení druhé podmínky (neklesající funkce) je dosaženo kontrolou souřadnic v ose oboru hodnot. Aby modelovaná funkce byla spojitá, je nutné zamezit případu, kdy dva kontrolní body mají shodné souřadnice v ose definičního oboru a různé v ose oboru hodnot.

Vzhledem k neustálé potřebě provádění kontroly dodržení zmíněných podmínek při jakémkoli pohybu kontrolního bodu, byl zvolen způsob implementace provázáním jednotlivých kontrolních bodů ukazateli na souřadnice patřičných jiných kontrolních bodů. Výhodou je rychlost algoritmu ověřování souřadnic. Je však nutné zajistit, aby při spuštění proběhla logicky správná inicializace u všech kontrolních bodů, a byly tak přiřazeny patřičné ukazatele na patřičná místa. Při odebírání či přidávání křivky musí být provedena stejná procedura jen u kontrolních bodů, kterých se taková změna týká.

Nastavení módu křivky

Křivka samotná není určena pouze polohou kontrolních bodů, ale i algoritmem, pro každý druh křivky jedinečným, který na základě souřadnic kontrolních bodů přiřadí křivce její tvar. V této práci byly použity úsečky a Bézierovy kubiky. Algoritmus pro získání tvaru obecné Bézierovy křivky, který byl použit i zde, se nazývá *de Casteljau*. Implementován byl následovně.

```
Point2D getDeCasteljau(float t, Point2D ctrlPts[ORDER+1])
```

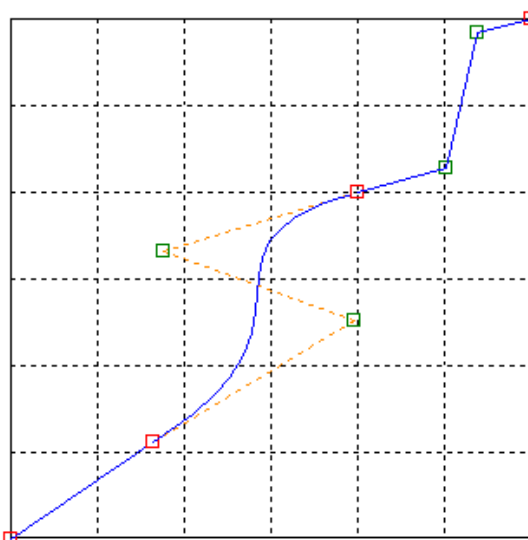
```

{ // #define ORDER 3 ... definovaná konstanta ... Bézierova KUBIKA
  for(short k = 0; k < ORDER; k++) {
    for(short i = 0; i < ORDER-k; i++) {
      ctrlPts[i].x = t * ctrlPts[i+1].x + (1 - t) * ctrlPts[i].x;
      ctrlPts[i].y = t * ctrlPts[i+1].y + (1 - t) * ctrlPts[i].y;
    }
  }
  return ctrlPts[0]; // bod na křivce pro relativní čas t
}

```

Vstupními parametry jsou souřadnice kontrolních bodů křivky (`ctrlPts`) a relativní čas $t \in \langle 0; 1 \rangle$ určující místo na křivce, kde se bude hledaný bod nacházet. Algoritmus de Casteljau provádí rozdělování úseček spojujících sousední kontrolní body křivky v poměru odpovídajícím času t a pro takto získané body použije stejný postup až do té doby, kdy zůstane jediný bod. Ten leží na dané křivce v místě odpovídajícím času t . Podrobný popis algoritmu de Casteljau je možné najít v [1].

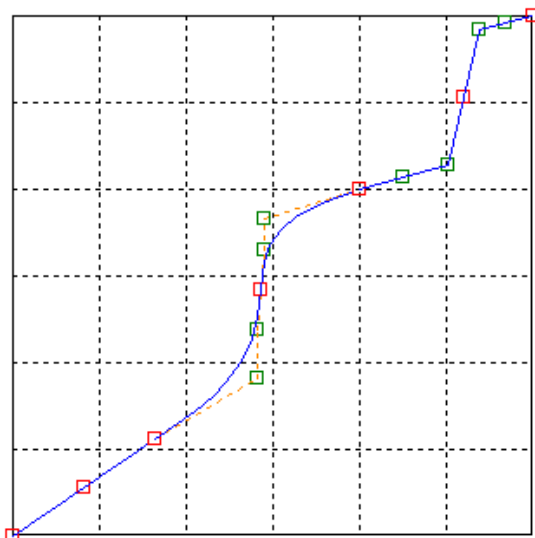
Při nastavení módu křivky využívajícím pouze úsečky je možné volit mezi možnostmi využívajícími všech kontrolních bodů (u kubiky jsou to 4) nebo pouze krajních, čímž je získána lomená čára, respektive jedna úsečka. Jednotlivé módy jsou zobrazeny na obrázku 3.5.



Obr. 3.5: Převodní funkce rozdělená na tři křivky v různých módech.

Přidání křivky

Přidání nové křivky je řešeno rozdělením původní křivky na dvě nové, přičemž tvar křivky zůstane zachován. Vizualně dojde pouze k přidání nových kontrolních bodů.



Obr. 3.6: Převodní funkce po rozdělení jednotlivých křivek na dvě nové v relativním čase $t = 0,5$.

Tato skutečnost je vidět na obrázku 3.6, kde byly jednotlivé křivky tvořící samotnou převodní funkci z obrázku 3.5 rozděleny v polovině své délky (odpovídá relativnímu času $t = 0,5$). U Bézierovy křivky je tato funkce řešena opět algoritmem de Casteljau, přičemž od jeho formy pro získání bodu na křivce popisované výše se liší pouze nutností zaznamenávat souřadnice určitých bodů získávaných při mezivýpočtech algoritmu. Počet nových kontrolních bodů pro kubiku je 7, přičemž souřadnice prvního a posledního zůstávají nezměněny a stejně jako prostřední kontrolní bod leží na křivce. Je vhodné podotknout, že souřadnice prostředního kontrolního bodu by byly výsledkem algoritmu de Casteljau pro získání bodu na křivce pro stejnou hodnotu relativního času t . Využití dělení Bézierovy kubiky v metodě implementující výpočet bodů celé křivky způsobem přizpůsobujícím se míře jejího zakřivení bude popsáno v části adaptivní vykreslování Bézierovy křivky. Implementace dělení Bézierovy křivky je následující.

```
void divideBezier(float t, Point2D toDivide[], Point2D divided[])
{
    // #define ORDER 3 ... definovaná konstanta ... Bézierova KUBIKA
    // toDivide...kontrolní body původní křivky
    // divided...kontrolní body dvou nových křivek
    short num = 2 * ORDER;
    divided[0] = toDivide[0]; // první i poslední kontrolní bod...
    divided[num] = toDivide[ORDER]; // ... zůstává stejný
    for(short k = 0; k < ORDER; k++) { // algoritmus de Casteljau
```

```

for(short i = 0; i < ORDER-k; i++) {
    toDivide[i].x = t * toDivide[i+1].x + (1 - t) * toDivide[i].x;
    toDivide[i].y = t * toDivide[i+1].y + (1 - t) * toDivide[i].y;
}
divided[k+1]      = toDivide[0];          // nové kontrolní body...
divided[num-k-1] = toDivide[ORDER-k-1]; //...přidávané symetricky
} }

```

Odebrání křivky

Při odebrání křivky dojde k odstranění patřičných kontrolních bodů a spojení dvou křivek v jednu novou.

Adaptivní vykreslování Bézierovy křivky

Vzhledem k tomu, že Bézierova křivka může nabývat libovolného množství velmi odlišných tvarů, bylo by nevhodné stanovit konstantní krok relativního času t a tím i pevný počet bodů, pomocí kterých je křivka vykreslována. V takovém případě by úsečka vytvořená Bézierovou křivkou byla vykreslena pomocí většího počtu bodů i přesto, že by mohla být vykreslena pouze pomocí bodů dvou. Naopak velmi zakřivené části křivky by mohly být vykreslovány s příliš malým rozlišením. Proto je vhodné použít metodu, která přizpůsobí počet a rozmístění bodů pro vykreslování tvaru křivky.

Implementována byla rekurzivní metoda využívající půlení Bézierovy křivky (dělení pro $t = 0,5$). Je však nutné definovat jisté kritérium plochosti (v následující ukázce kódu pojmenované jako `flatnessCriterion`), které označí již dostatečně plochou křivku a zastaví tím zanořování. Z každé takovéto křivky jsou uloženy souřadnice jejího prvního kontrolního bodu. Po vykonání celého algoritmu je v souboru těchto bodů dáno, že mezi dvěma sousedícími body je možné proložit úsečku s přesně definovanou chybou, na jejímž základě provádí funkce `flatnessCriterion()` své rozhodování. Implementace je tedy následující.

```

void getAdaptBezier(Point2D ctrlPts[], short numOfCurves, T &points)
{
    // ctrlPts.....kontrolní body křivek pro vykreslení
    // numOfCurves...počet křivek k vykreslení
    // points.....kontejner pro uložení bodů k následnému vykreslení
    for(short j = 0; j < numOfCurves; j++) {
        short index = j*ORDER;
        // &actCurve...adresa prvního kontrolního bodu zkoumané křivky
    }
}

```

```

Point2D &actCurve = ctrlPts[index];
Point2D divided[2*ORDER+1];
if(!flatnessCriterion(&actCurve)) { // pokud není křivka plochá
    divideBezier(0.5f, &actCurve, divided); // ... rozpul ji
    getAdaptBezier(divided, 2, points); // ... a zanoř se
} else { // pokud je přívka plochá, ulož první kontrolní bod
    points.insert(ctrlPts[index]);
} } }

```

3.2.2 Funkční rozhraní a jeho vnitřní implementace

Uživatelské rozhraní umožňuje ovládat tvar převodní funkce pomocí křivek, funkční rozhraní poskytuje explicitní vyjádření převodní funkce $y = f(x)$. Zvnějšku se tak převodní charakteristika jeví jako běžná matematická funkce.

Explicitní Bézierovy křivky

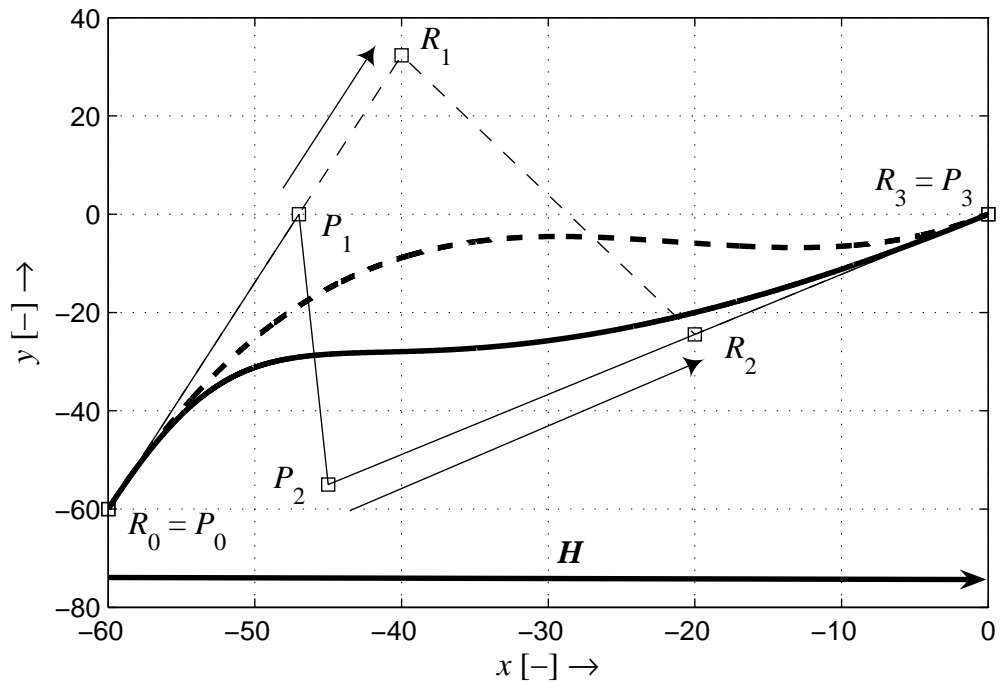
Z principu jsou všechny křivky, nejen ty Bézierovy, určeny pro použití v oblasti grafiky, kde je omezení na modelaci pouze matematických funkcí nežádoucí. Ne tak v této práci. V literatuře byla nalezena podmínka, po jejímž splnění je možné Bézierovy křivky používat jako matematické funkce s explicitním vyjádřením $y = f(x)$ [6]. Touto podmínkou je rovnoměrné rozložení kontrolních bodů v ose X.

Na obrázku 3.7 je zobrazena obecná Bézierova křivka a od ní odvozená explicitní Bézierova křivka. Krajní kontrolní body obou křivek se shodují. Poloha vnitřních kontrolních bodů explicitní Bézierovy křivky je získána dle [6]

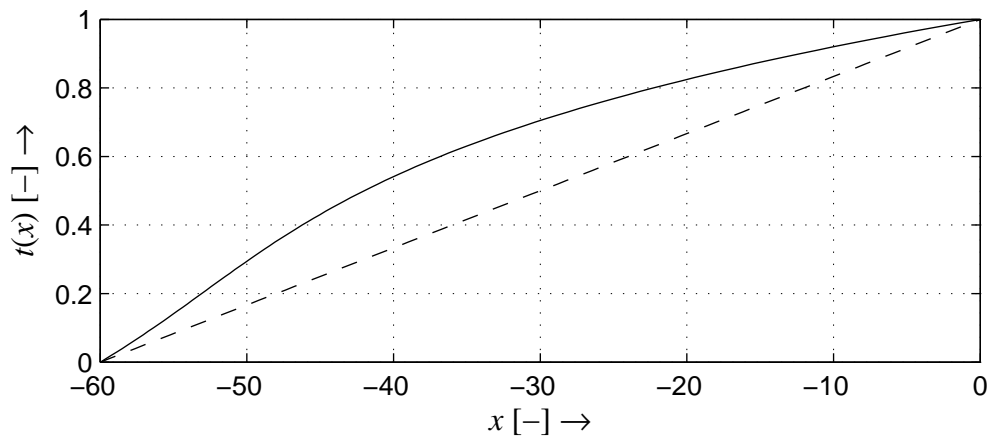
$$\begin{aligned}
 R_1 &= P_0 + \frac{\mathbf{H} \cdot \mathbf{H}}{3(P_1 - P_0) \cdot \mathbf{H}}(P_1 - P_0), \\
 R_2 &= P_3 - \frac{\mathbf{H} \cdot \mathbf{H}}{3(P_2 - P_3) \cdot \mathbf{H}}(P_2 - P_3),
 \end{aligned}
 \tag{3.2}$$

kde \mathbf{H} je vektor rovnoběžný s osou X o velikosti rovné vzdálenosti krajních kontrolních bodů křivky v ose X. Kontrolní body R_1 a R_2 jsou tak rovnoměrně rozmístěny v ose X, přičemž směr posunutí odpovídá směru vektoru $P_1 - P_0$ případně $P_2 - P_3$.

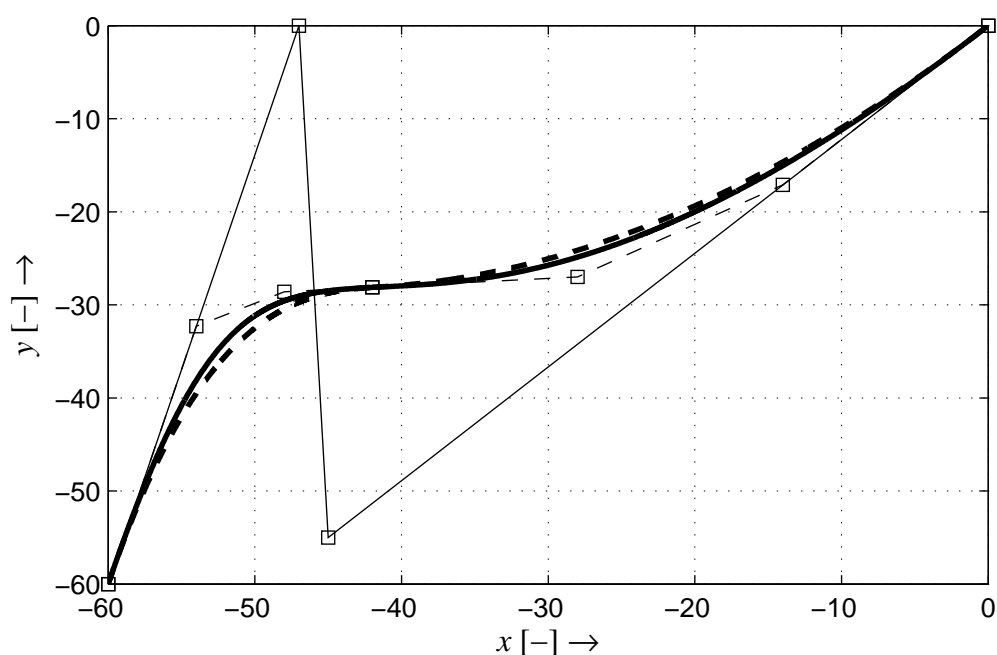
Důsledek rovnoměrného rozložení kontrolních bodů v ose X je vidět na obrázku 3.8. Z původní nelineární závislosti se stala závislost lineární a je tak možné pouhou lineární interpolací získat pro danou hodnotu x hodnotu relativního času t , jejíž nalezení by jinak bylo možné pouze numerickou metodou. Poté je možné již běžným způsobem získat souřadnice bodu v ose Y pro danou hodnotu relativního času t . Tím je získána funkční hodnota y v daném bodě x .



Obr. 3.7: Obecná Bézierova křivky (plnou čarou) s kontrolními body P a odvozená explicitní Bézierova křivka (čárkovaně) s kontrolními body R .



Obr. 3.8: Závislost relativního času t na poloze bodu na Bézierově křivce v ose X pro obecnou Bézierovu křivku (plnou čarou) a explicitní Bézierovu křivku (čárkovaně).



Obr. 3.9: Obecná Bézierova křivka (plnou čarou) a dvě odvozené explicitní Bézierovy křivky (čárkovaně).

Na obrázku 3.9 je zobrazena obecná Bézierova křivka z obrázku 3.7 a aproximovaná křivka složená ze dvou explicitních Bézierových křivek. Je vidět značné snížení chyby aproximace oproti obrázku 3.7.

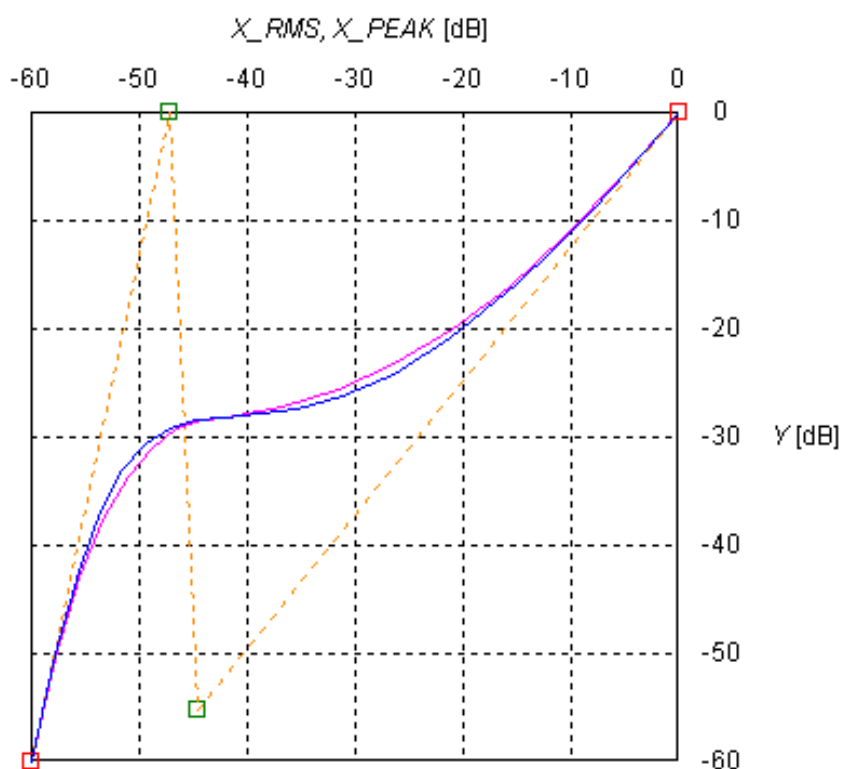
Převod obecné Bézierovy křivky na dílčí explicitní Bézierovy křivky je prováděn obdobným algoritmem jako adaptivní vykreslování popisované v části 3.2.1. Odhad maximální chyby aproximace (podmínka pro zastavení rekurze algoritmu) je počítán dle (3.3) [6]. Odhad je odvozen ze vzdálenosti, o kterou jsou posunuty vnitřní kontrolní body explicitní Bézierovy křivky R_1 a R_2 jako

$$\epsilon_{\max} = \frac{3}{4} \max(|R_1 - P_1|, |R_2 - P_2|). \quad (3.3)$$

V případě z obrázku 3.9 bylo nutné provést jedno zanoření, aby byl odhad chyby aproximace snížen pod danou mez. Ve výsledku tedy stačilo, že původní obecná Bézierova křivka byla rozdělena pouze na dvě nové explicitní křivky.

Pro uživatele modelujícího tvar křivky je převod na explicitní Bézierovy křivky transparentní. Možnost svobodného pohybu kontrolními body P obecné Bézierovy křivky zůstává uživateli zachována. Převod na explicitní vyjádření je prováděn při pohybu kontrolním bodem vedle výpočtu pro vykreslení původní křivky. Aby byla uživateli poskytnuta zpětná vazba, je explicitní vyjádření křivky použito nejen pro výpočet funkční hodnoty, ale i pro vykreslování skutečného tvaru převodní charakteristiky.

Na obrázku 3.10 je zobrazeno uživatelské rozhraní modelace převodní charakteristiky s Bézierovou křivkou z obrázku 3.9. Modře je vykreslena obecná Bézierova křivka odpovídající poloze kontrolních bodů, purpurově její aproximace explicitními Bézierovými křivkami. Osám je již přiřazen význam i jednotka. Ose definičního oboru je přiřazena úroveň efektivní či špičkové hodnoty signálu X_{RMS} či X_{PEAK} v dB. Ose oboru hodnot je přiřazena úroveň odpovídající výstupnímu signálu dynamického procesoru Y také v dB. Je vhodné zmínit, že z důvodu nepohodlného formátování textu prostředky poskytovanými technologií VST, nejsou popisky os naformátovány správně.



Obr. 3.10: Uživatelské rozhraní pro modelaci převodní charakteristiky.

Funkční rozhraní jako celek

Jak bylo ukázáno výše v části 3.2.1, modelovaná převodní funkce se může skládat z více křivek (funkcí) různých druhů. Pro danou hodnotu úrovně efektivní nebo špičkové hodnoty je nutné vyhledat funkci, do jejíhož definičního oboru daná hodnota spadá. Jelikož se průměrný počet křivek pohybuje v řádu jednotek, za datovou strukturu, v níž se vyhledává, bylo zvoleno pole s informacemi o poloze krajních bodů křivek v ose definičního oboru. Samotné křivky jsou uloženy v lineární datové

struktury `std::vector<>` poskytované standardní knihovnou jazyka C++, do níž ukazuje výsledek algoritmu vyhledávání.

Každá křivka, nezávisle na jejím módu (viz část 3.2.1), vystupuje navenek jako jedna křivka, přestože vnitřně může být reprezentována různým počtem částí. Jak bylo popsáno výše, pro obecnou Bézierovu křivku byl zvolen způsob reprezentace pomocí jedné či více explicitních Bézierových křivek. Mezi jednotlivými explicitními křivkami je nutné vyhledat tu správnou stejným způsobem jako na úrovni převodní funkce. Algoritmus i typ datové struktury byl zvolen v obou případech obdobný. Odlišností je reálně možný maximální počet prvků. Bylo zjištěno, že při nejvyšší nastavitelné přesnosti aproximace, složitém tvaru křivky a při maximální délce křivky (celá převodní funkce složená z jedné Bézierovy křivky), se počet explicitních Bézierových křivek pohybuje kolem čísla 20. Díky tomu, že jsou jednotlivé explicitní křivky v datové struktuře seřazeny dle polohy v ose definičního oboru, bylo by vhodné pro vyhledávání implementovat metodu půlení intervalu, aby jeho časová složitost nerostla s rostoucím počtem křivek lineárně, ale logaritmicky [2].

3.3 Implementace dynamického filtru

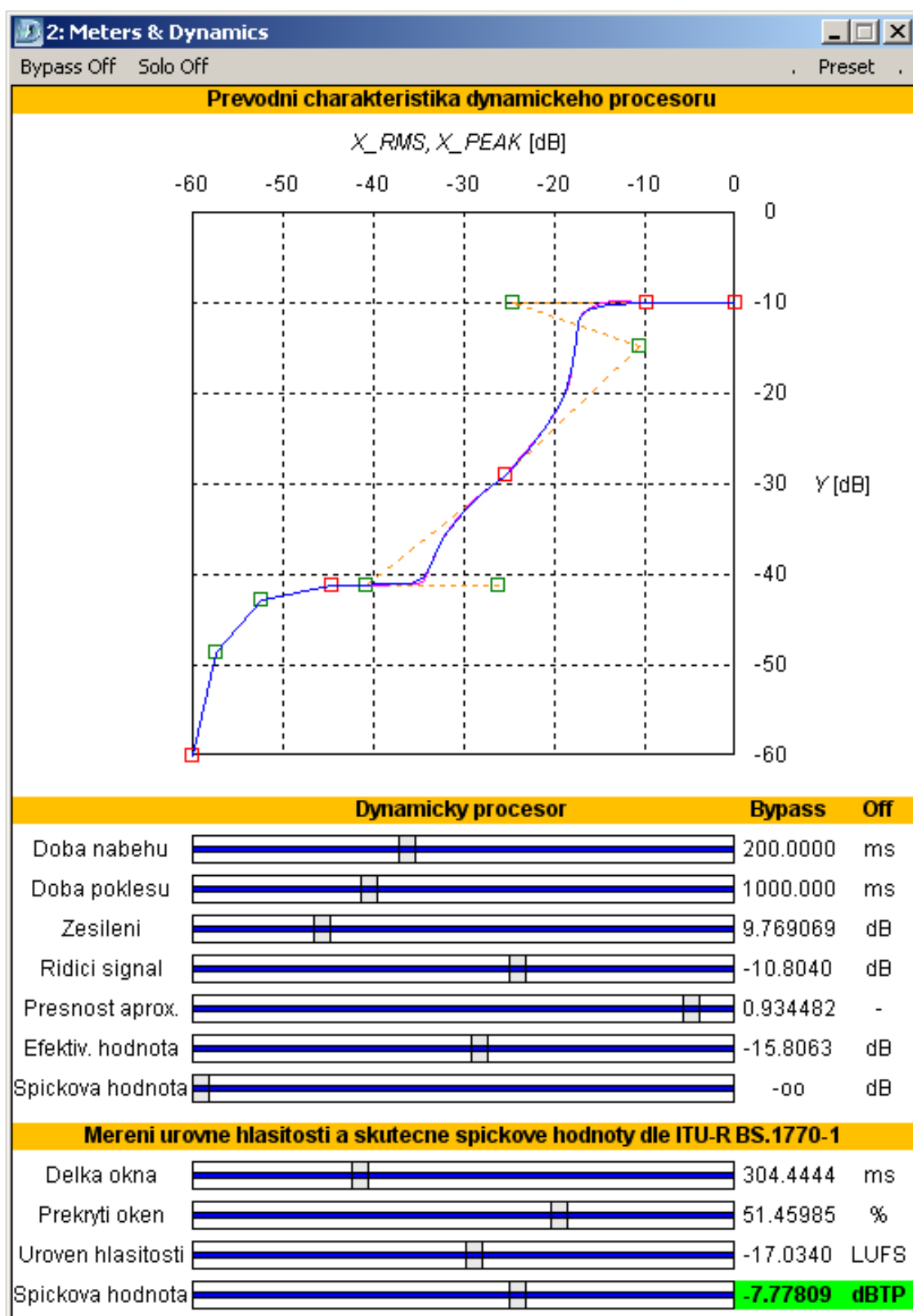
Dynamický filtr byl implementován jako třída jazyka C++ odvozená od třídy popsané v části 3.1.2. Virtuální metoda třídy definující chování její instance byla implementována dle blokového schématu na obrázku 2.5.

3.4 Implementace dynamického procesoru

Dynamický procesor byl implementován v jazyce C++ jako plug-in modul technologie VST. Je tak umožněno použít modul pro zpracování zvukového signálu v reálném čase. Provedená implementace dynamického procesoru spojuje v jeden celek veškeré implementace v jazyce C++ popsané v této kapitole.

Struktura implementace dynamického procesoru odpovídá obrázku 2.1. Měření efektivní a špičkové hodnoty vstupního signálu, které jsou vstupem pro blok převodní funkce, je prováděno metodami měření pomocí plovoucího odhadu popsanými v části 1.1. Měření úrovně hlasitosti a skutečné špičkové hodnoty dle doporučení ITU-R BS.1770-1 jsou zapojena na výstup dynamického procesoru, nicméně je umožněno přemostění celého dynamického procesoru a přivedení vstupního signálu přímo na vstup metod měření úrovně hlasitosti a skutečné špičkové hodnoty.

Na obrázku 3.11 je vidět uživatelské rozhraní systému. Pro ukazatele měřených veličin i tahové potenciometry byl použit shodný způsob grafické reprezentace. Parametry dynamického procesoru kromě převodní funkce se nachází ve střední části.



Obr. 3.11: Uživatelské rozhraní dynamického procesoru a měření dle doporučení ITU-R BS.1770-1.

3.4.1 Parametry dynamického procesoru

Doba náběhu a **doba poklesu** ovlivňují odpovídající časové parametry dynamického filtru, který vyhlazuje průběh řídicího signálu systému.

Zesílení je parametr, který obecně slouží pro nastavení hlasitosti. Důvodem k jeho zavedení byla potřeba kompenzace případného útlumu dynamického procesoru.

Řídicí signál je měřený parametr. Ukazuje hodnotu v decibelech, kterou je vstupní signál násoben. Je vhodné, aby parametr **zesílení** byl přibližně dorovnán k jeho průměrné hodnotě. Řídicí signál je získán z výstupního signálu převodní funkce filtrací dynamickým filtrem.

Efektivní a **špičková hodnota** jsou měřené parametry. Dynamický procesor reaguje vždy pouze na jeden z nich. Jsou vstupním signálem pro převodní funkci a jejich přepínání se děje kliknutím myši na název příslušného parametru.

Přesnost aproximace reprezentuje hodnotu s níž se porovnává odhad velikosti chyby ϵ_{\max} při převodu obecné Bézierovy křivky na její explicitní vyjádření. Její nastavení se promítá pouze do křivky, jejímž kontrolním bodem bylo naposledy pohnuto. Při výběru kontrolního bodu Bézierovy křivky myší, zobrazí hodnotu pro odpovídající křivku.

Bypass je společně s vedlejší popiskou indikátorem přemostění dynamického procesoru. Kliknutím myši do oblasti textu **on**, případně **off**, dojde ke změně současného stavu.

Je nutné uvést, že do systému je zavedena automatická šumová brána s parametrem rozsah o velikosti 40 dB, jejíž rozhodovací úroveň odpovídá poloze prvního kontrolního bodu v ose X. Podobně je zaveden automatický limiter, který vstupní signál větší než 0 dBFS omezí na úroveň odpovídající poloze posledního kontrolního bodu v ose Y.

3.4.2 Parametry měření dle doporučení ITU-R BS.1770-1

Parametry měření dle ITU-R BS.1770-1 jsou umístěny ve spodní části okna uživatelského rozhraní.

Délka okna a **překrytí oken** nastavují odpovídající parametr u metody měření úrovně hlasitosti, jejíž výsledná hodnota je počítána z úseku signálu o délce dané prvním z parametrů.

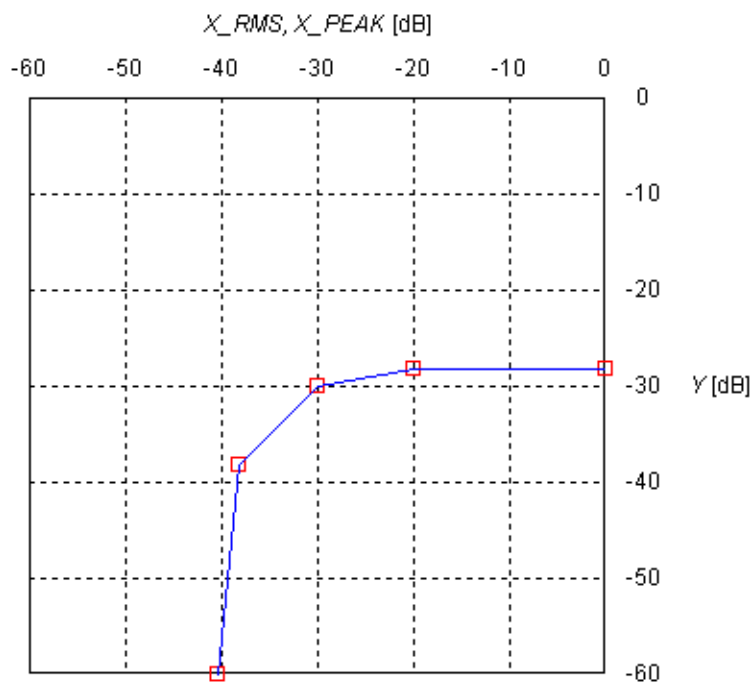
Úroveň hlasitosti a **špičková hodnota** jsou měřené parametry. Reagují buď na výstupní nebo vstupní signál dynamického procesoru, a to dle stavu zmíněného přepínače **bypass**.

3.4.3 Kombinovaná stupnice

U měřených parametrů (řídící signál a efektivní a špičková hodnota v části dynamický procesor a úroveň hlasitosti a špičková hodnota v části měření dle doporučení ITU-R BS.1770-1) je použita kombinovaná stupnice. Exponenciální část zabírá spodní polovinu rozsahu, lineární horní polovinu. Lineární část má rozsah 20 dB. U měření špičkových hodnot a řídicího signálu je zobrazitelné maximum 3 dBFS. U ostatních měřených parametrů je nejvyšší hodnota 0 dBFS.

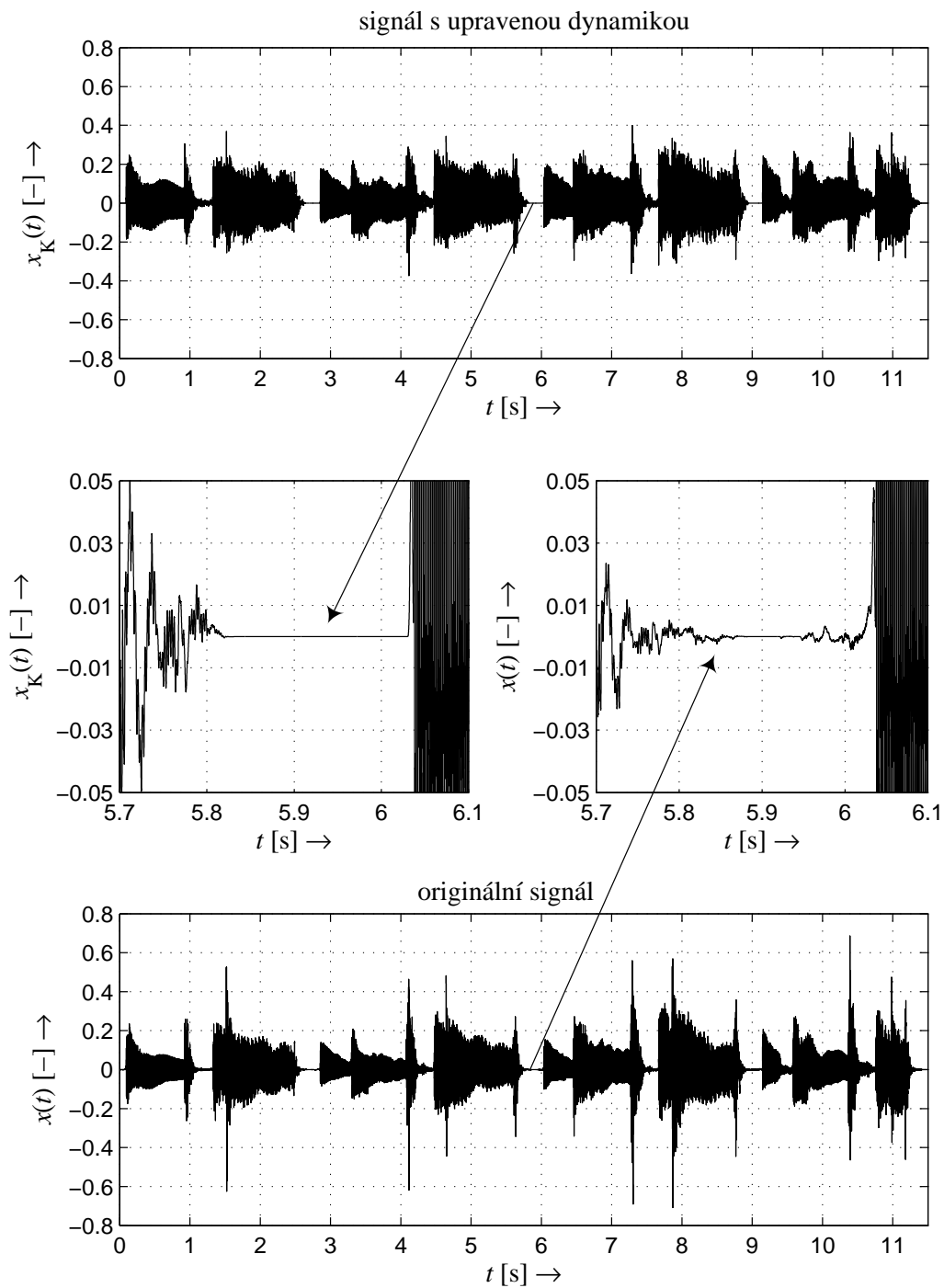
3.5 Výsledky implementace

Z důvodu, že úprava zvukového signálu dynamickým procesorem nemusí být při pouhém poslechu zcela zřejmá, byl v jednom případě výstupní signál systému zaznamenán. Na obrázku 3.12 je použitá převodní funkce dynamického procesoru. Časový průběh upraveného a původního signálu je na obrázku 3.13. V prostřední části obrázku jsou zvětšené oblasti z každého průběhu, na nichž je vidět potlačení signálu s nízkou úrovní.



Obr. 3.12: Převodní funkce kombinovaného systému použitá pro úpravu signálu na obrázku 3.13 pro dobu náběhu 5 ms, dobu poklesu 15 ms a pro reakci systému na efektivní hodnotu.

Původní signál obsahuje nahrávku, která byla pořízena hrou na elektroakustickou kytaru s cílem dosáhnout velké dynamiky. Nahrávka byla poté normalizována



Obr. 3.13: Časový průběh signálu s upravenou dynamikou kombinovaným systémem z obrázku 3.12 (nahore) a časový průběh původního signálu (dole) při shodné úrovni hlasitosti obou signálů $L_K = -25,1$ LUFS.

ke špičkové hodnotě s odstupem 3 dB od plného rozsahu. Následně byla změřena úroveň hlasitosti nahrávky $L_K = -25,1$ LUFS za použití délky okna rovné délce nahrávky. Úroveň hlasitosti výstupního signálu byla k této hodnotě dorovnána.

Na časovém průběhu původního signálu jsou vidět krátké výrazné špičky. Zařazenou limitací od úrovně -20 dB byly tyto ve výsledném signálu do značné míry potlačeny. Dále byl vstupní signál s úrovní od -30 dB do -20 dB komprimován v poměru 4 : 1. Tímto bylo dosaženo menšího rozdílu v hlasitosti mezi jednotlivými silnějšími a slabšími úseky užitečného signálu. Do oblasti od -41 dB do -38 dB byl zařazen expander, který se společně s automatickou šumovou bránou podílí na potlačení slabých částí signálu. V oblasti od -38 dB do -30 dB byl vstupní signál ponechán beze změny.

3.6 Hodnocení výpočetní náročnosti

Na osobním počítači bylo provedeno měření procentuálního zatížení procesoru v čase (procesor Intel Core 2 Duo 2, 13 GHz, operační paměť 1 GB 800 MHz DDR2, operační systém Microsoft Windows XP SP3). Tabulky naměřených hodnot a vypočítaných průměrných hodnot je možné najít v příloze A.

Díličí části implementovaného systému, tedy dynamický procesor i jednotlivá měření dle ITU-R BS.1770-1, byly testovány samostatně, a to při zavřeném uživatelském rozhraní. Zatížení procesoru při vypnutém zpracování signálu se pohybovalo mezi 0–1 %. Doba trvání každého měření byla stanovena na dvě minuty, přičemž každé dvě sekundy byla zapisována hodnota procentuálního zatížení procesoru. Tato měření dokumentují tabulky A.1–A.4.

Při zpracování pouze dynamickým procesorem byla převodní funkce nejprve vy-modelována pomocí jedné křivky (tabulka A.1) a následně pomocí osmi křivek (tabulka A.2). Různé módy křivek jsou značeny jako LM (úsečka), PM (lomená čára), BM1 (Bézierova křivka s nízkou přesností aproximace v jednoduchém tvaru), BM2 (Bézierova křivka s vysokou přesností aproximace ve složitém tvaru). Bylo ověřeno, že výpočetní náročnost roste se složitostí i počtem křivek. Hodnoty průměrného procentuálního zatížení procesoru stoupaly od 10,47 % k 10,97 %.

V případě, kdy byla v činnosti pouze metoda měření úrovně hlasitosti, byl zaznamenán nárůst procentuálního zatížení procesoru jak při zvýšení velikosti překrytí oken, tak i při zvýšení jejich délky, přičemž u parametru překrytí byl tento nárůst podstatně vyšší. Bylo ověřeno, že kombinace nejdelšího okna a největší hodnoty překrytí dosahuje nejvyšší výpočetní náročnosti. Hodnoty průměrného procentuálního zatížení procesoru stoupaly od 9,8 % k 10,13 % (viz tab. A.3).

Nejvyšší absolutní velikost průměrného procentuálního zatížení procesoru dosáhla metoda měření skutečné špičkové hodnoty, a to 11,47% (viz tab. A.4). Je tedy více než vhodné, aby byl snížen řád interpolačního filtru.

Tabulka A.5 uvádí výsledky pro aktivní všechny dílčí části plug-in modulu, a to při zavřeném (12,57%) i otevřeném (15,72%) okně uživatelského rozhraní. Výsledky ukazují, že uživatelské rozhraní tvoří značnou zátěž pro výpočetní systém.

Závěrečné měření (viz tab. A.6) ukazuje vliv uživatelské interakce pro pohyb kontrolními body i tahovými potenciometry. Měření bylo prováděno při vypnutém zpracování signálu stálým pohybem daným objektem po dobu přibližně 15 s. U křivek v módu úsečky a lomené čáry se projevil nárůst výpočetní náročnosti s rostoucím počtem křivek. Naproti tomu u Bézierových křivek tomu tak není, a to proto, že počet bodů nutných pro vykreslení v obou daných situacích je srovnatelný – kratší křivky jsou vykreslovány s nižším rozlišením než delší, jelikož podmínka plochosti je dána v absolutní míře nikoli relativní (např. jeden pixel). Nárůst výpočetní náročnosti při zvýšení přesnosti aproximace Bézierovy křivky se neprojevil.

3.7 Slabá místa a chyby implementace

Již při práci na semestrálním projektu bylo odhaleno slabé místo prováděné implementace. Je jím přístup ke sdíleným datům, které nemohou být reprezentovány jako VST parametry, a k nimž tak není technologií VST zajištěn bezpečný vícevláknový přístup. Např. pokud je vlákno spravující uživatelské rozhraní přerušeno v operaci odebrání křivky a běh předán vláknu provádějící zpracování signálu, může dojít k přístupu k neplatným datům a pádu programu. Byla přijata dvě opatření, které riziko pádu programu snižují. Zápis a čtení jsou prováděny nad různými úložišti, jejichž obsah je však udržován stejný. Riziko pádu je tak omezeno na okamžik, kdy je nutné po zápisu aktualizovat úložiště pro čtení. Kontejnery standardní knihovny jazyka C++ poskytují funkci `swap()` pro výměnu dat se stejným typem kontejneru. Rychlost této funkce se však ukázala jako nedostatečná a proto byl přístup k datům umožněn pouze přes ukazatele, které se po zápisu prohodí. Rychlost této metody je již dostatečná. Riziko pádu programu je tak sníženo na minimum, ne však na nulu.

Druhým opatřením je zachytávání výjimek standardní knihovny jazyka C++. Toto opatření neřeší problém přístupu k neplatným datům, ale pouze zamezí provedení neplatné operace nad platnými daty. Příkladem může být snaha o přístup na neplatnou pozici kontejneru v důsledku neaktuálnosti dat. Pokud je pro přístup k prvkům kontejneru typu `std::vector<>` v kritických místech použita funkce `at()`, je kontrolována platnost dané pozice a v případě její neplatnosti je vyvolána výjimka `std::out_of_range`. Tu je možné výše zachytit a přizpůsobit běh programu nastalé

situaci. Popsaná opatření zajistí relativně spolehlivý běh programu při spuštěném zpracování signálu a potlačí slabé místo zvoleného způsobu implementace.

Jednoznačnou chybou implementace, jejímž důsledkem je občasný pád programu při zavření či otevření okna uživatelského rozhraní případně při zavření celého plugin modulu, je špatný způsob přístupu k parametrům ve třídě `CStaticCurveEditor` odvozené od třídy `AEffGUIEditor` zastupující standardní uživatelské rozhraní technologie VST. Při přerušení vlákna spravujícího uživatelské rozhraní a předání běhu vláknu, v němž probíhá zpracování signálu, hrozí, že v uvedených případech dojde ve snaze aktualizovat např. parametr efektivní hodnoty k přístupu k neplatným datům a pádu programu. Tato chyba nebyla odstraněna a přitom jí mohlo být předejito pokud by pro grafickou reprezentaci VST parametrů bylo použito standardních prostředků poskytovaných technologií VST.

4 ZÁVĚR

Zadání diplomové práce nebylo zcela splněno. Bylo upřednostněno dokončení systému změny dynamiky signálu jako celku před implementací všech zadaných metod měření. Metody měření stejnosměrné složky a sledování amplitudové obálky signálu nebyly ani prostudovány ani implementovány. Metody měření úrovně signálu typu Volume Unit a Peak Program podle standardů DIN a EBU byly nahrazeny nejnovějšími metodami měření úrovně hlasitosti a skutečné špičkové hodnoty dle doporučení ITU-R BS.1770-1. Prostudovány a implementovány byly také základní metody měření efektivní a špičkové hodnoty. Implementován byl i dynamický filtr použitý v systému změny dynamiky signálu. Algoritmy byly vždy nejprve testovány v prostředí Matlab a následně implementovány v jazyce C++.

Byl navržen a implementován algoritmus plovoucího okna s nastavitelnou délkou okna a velikostí překrytí oken. Algoritmus zajišťuje správnou práci s okny nezávisle na velikosti bloku zpracovávaných dat. Tím je umožněno nastavení libovolné délky okna omezené pouze rozsahem primitivního datového typu `long`, tedy rozsahem 32 bitů. Algoritmus byl použit v metodě měření úrovně hlasitosti dle doporučení ITU-R BS.1770-1.

Byl navržen a implementován systém modelování převodní funkce. Systém poskytuje uživateli značné možnosti při tvarování požadovaného průběhu, a to zejména modelováním hladkých tvarů pomocí Bézierových křivek. Pro reprezentaci obecné Bézierovy křivky jako matematické funkce byl navržen a implementován algoritmus rozkladu křivky na explicitní Bézierovy křivky s nastavitelnou přesností aproximace.

Byl navržen a implementován systém změny dynamiky zvukového signálu pomocí obecné po částech lineární neklesající převodní funkce včetně možnosti nastavení doby náběhu i poklesu prahových úrovní. Systém využívá a spojuje do jednoho funkčního celku veškeré implementované algoritmy, a to základní metody měření efektivní a špičkové hodnoty pro řízení systému dle charakteru vstupního signálu, dále systém pro modelaci převodní funkce k získání požadovaných dynamických vlastností výstupního signálu a nakonec metody měření dle doporučení ITU-R BS.1770-1, které monitorují výstupní signál. Bylo vytvořeno i uživatelské rozhraní, kde probíhá nastavení parametrů i zobrazení měřených hodnot.

Jakákoli další snaha v rozvíjení této práce by měla být nejprve směřována ke snížení výpočetní náročnosti. V části 3.6 byl prokázán mimo jiné značný podíl metody měření skutečné špičkové hodnoty na vytížení procesoru. Cestou je snížení řádu interpolačního filtru. Dalším krokem by mělo být opravení způsobu grafické reprezentace VST parametrů popsaného v části 3.7, které by mělo mimo napravení uvedené chyby snížit podíl uživatelského prostředí na zatížení procesoru. Cestou je správné použití standardních prostředků poskytovaných technologií VST.

LITERATURA

- [1] ALEXANDR, L. *Výuka počítačové grafiky cestou WWW* [online]. [cit. 1. 12. 2008]. Dostupné z: <<http://www.hyperkrychle.cz/curves/obsah.html>>.
- [2] BURGET, R. *Algoritmy vyhledávání* [online]. Poslední úpravy 24. 9. 2009 [cit. 17. 5. 2010]. Dostupné přes informační systém Vysokého učení technického: <https://www.vutbr.cz/elearning/file.php/87027/prednasky/07_algoritmy_vyhledA_vA_nA_.pdf>.
- [3] International Telecommunication Union. Radiocommunication Sector. *Recommendation ITU-R BS.1770-1. Algorithms to measure audio programme loudness and true-peak audio level (Question ITU-R 2/6)* [online]. [cit. 11. 5. 2010]. Dostupné z: <<http://webs.uvigo.es/servicios/biblioteca/uit/rec/BS/R-REC-BS.1770-1-200709-I!!PDF-E.pdf>>.
- [4] JOSUTTIS, N. M. *C++ Standardní knihovna a STL Kompletní průvodce*. 1. vyd. Brno: CP Books, a.s., 2005. ISBN 80-251-0511-1.
- [5] Nielsen, S. H.; TC Group Research. *Note on measurement units for loudness* [online]. 8.10.2009 [cit. 11. 5. 2010]. Dostupné z: <http://www.tcelectronic.com/media/nielsen_loudness_units.pdf>.
- [6] SEDERBERG, T. W.; FALAI, C.; KLIMASZEWSKI, K. S. *Rotated Explicit Curves* [online]. [cit. 15. 5. 2010]. Dostupné z: <<http://tom.cs.byu.edu/tom/papers/rec.pdf>>.
- [7] SCHIMMEL, J. *Syntéza zvukových efektů s využitím nelineárního zpracování signálů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2006. Vedoucí disertační práce Prof. Ing. Zdeněk Smékal, CSc.
- [8] ZÖLZER, U. *DAFX – Digital Audio Effects*. 1st ed. Chichester: John Wiley & Sons, Ltd, 2002. ISBN 0-471-49078-4.
- [9] ZÖLZER, U. *Digital Audio Signal Processing*. 1st ed. Chichester: John Wiley & Sons, Ltd, 1997. ISBN 0-471-97226-6.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AT	koeficient systému odvozený z doby náběhu t_{AT}
BM1	křivka převodní funkce složená z jedné nebo více Bézierových křivek ve tvaru úsečky pro nejnižší přesnost aproximace
BM2	křivka převodní funkce složená z jedné nebo více Bézierových křivek ve složitějším tvaru pro nejvyšší přesnost aproximace
BS.1770-1	doporučení organizace ITU-R pro měření úrovně hlasitosti a skutečné špičkové hodnoty zvukového signálu
CPU	procentuální zatížení procesoru
CPU_O	procentuální zatížení procesoru při otevřeném okně uživatelského rozhraní
dBFS	jednotka decibel vztažená k rozsahu AD a DA převodníku
dBTP	jednotka decibel vztažená k rozsahu AD a DA převodníku používaná při měření skutečné špičkové hodnoty (true-peak)
ϵ_{max}	největší možná chyba aproximace Bézierovy křivky explicitní Bézierovou křivkou
f	kmitočet signálu v jednotkách Hertz
$f[n]$	vstupní signál dynamického filtru
f_m	mezní kmitočet filtru
f_{vz}	vzorkovací kmitočet
f_{vz0}	vzorkovací kmitočet před nadvzorkováním
f_{vz1}	vzorkovací kmitočet po nadvzorkování
$g[n]$	řídící signál
G	úroveň řídicího signálu v decibelech
H	vektor rovnoběžný s osou definičního oboru funkce o velikosti rovné vzdálenosti krajních kontrolních bodů křivky (funkce)
IIR	filtr s nekonečnou impulzní odezvou (infinite impulse response)

ITU-R	mezinárodní telekomunikační unie, odbor radiokomunikace
k	koeficient doby poklesu ukazatele špičkové hodnoty v absolutní míře
k_{AR}	koeficient dynamického filtru odpovídající koeficientu AT nebo RT podle směru změny vstupního signálu
L	poměr nadzvorkování
Leq	metoda měření efektivní hodnoty zvukového signálu (equivalent sound level)
Leq(K)	metoda měření úrovně hlasitosti, metoda Leq vážená křivkou RLB a křivkou zohledňující akustický stín hlavy
Leq(RLB)	metoda měření úrovně hlasitosti, metoda Leq vážená křivkou RLB
Leq(R2LB)	jiné označení pro Leq(K)
L_K	úroveň hlasitosti kmitočtově vážená křivkou K též označovanou jako R2LB
LM	křivka převodní funkce složená z jedné nebo více úseček
LU	jednotka úrovně hlasitosti bez referenční úrovně nebo pro vyjádření rozdílu v úrovních hlasitosti
LUFS	jednotka úrovně hlasitosti vztažená k rozsahu AD a DA převodníku
m	diskrétní čas
M	počet kanálů
n	diskrétní čas
N	délka okna
P	kontrolní body obecné Bézierovy křivky
PM	křivka převodní funkce složená z jedné nebo více lomených čar
R	kontrolní body explicitní Bézierovy křivky
RLB	váhovací křivka pro měření úrovně hlasitosti (revised low-frequency B-curve)
R2LB	váhovací křivka pro měření úrovně hlasitosti spojující RLB křivku s křivkou zohledňující akustický stín hlavy, také označována jako K

RMS	efektivní hodnota signálu (root mean square)
RT	koeficient systému odvozený z doby poklesu t_{RT}
T	vzorkovací perioda
t	relativní čas křivky ($t \in \langle 0; 1 \rangle$) nebo skutečný čas v sekundách
t_{AT}	dobu náběhu (attack time)
TAV	parametr určující dobu náběhu i dobu poklesu
t_{RT}	dobu poklesu (release time)
t_{TAV}	časový parametr určující dobu náběhu i dobu poklesu
VST	technologie zpracování zvukového signálu v reálném čase (virtual studio technology)
W_i	váhovací koeficient pro kanál i
x	poloha bodu v ose X
X	osa definičního oboru funkce
$x[n]$	vstupní signál
$x[n - D]$	zpožděný vstupní signál o D vzorků
X	úroveň vstupního signálu v decibelech
x_K	signál s upravenou dynamikou
X_{PEAK}	úroveň špičkové hodnoty vstupního signálu v decibelech
X_{RMS}	úroveň efektivní hodnoty vstupního signálu v decibelech
x_{PEAK}	špičková hodnota signálu
x_{RMS}	efektivní hodnota signálu
$x_w[n]$	kmitočtově vážený vstupní signál
y	poloha bodu v ose Y
Y	osa oboru hodnot funkce
$y[n]$	výstupní signál
Y	úroveň výstupního signálu v decibelech
z_i	aritmetický průměr druhých mocnin (mean-square energy), pro kanál i

SEZNAM PŘÍLOH

A	Tabulky naměřených hodnot	64
B	Obsah přiloženého CD	66

A TABULKY NAMĚŘENÝCH HODNOT

Tab. A.1: Naměřené a vypočítané hodnoty procentuálního zatížení procesoru v čase pro různé módy křivky převodní funkce dynamického procesoru.

	LM	PM	BM1	BM2		LM	PM	BM1	BM2		LM	PM	BM1	BM2		LM	PM	BM1	BM2	
t	CPU	CPU	CPU	CPU	t	CPU	CPU	CPU	CPU	t	CPU	CPU	CPU	CPU	t	CPU	CPU	CPU	CPU	
[s]	[%]	[%]	[%]	[%]	[s]	[%]	[%]	[%]	[%]	[s]	[%]	[%]	[%]	[%]	[s]	[%]	[%]	[%]	[%]	
0	13	13	6	6	30	17	14	13	13	60	4	5	17	16	90	7	10	17	17	
2	15	15	16	15	32	7	6	17	17	62	7	11	13	11	92	14	15	6	5	
4	5	4	16	12	34	6	7	8	7	64	15	17	6	6	94	15	11	9	9	
6	2	4	6	3	36	14	15	7	7	66	17	17	13	13	96	6	2	17	16	
8	9	11	7	10	38	18	17	16	16	68	11	6	18	18	98	2	4	13	11	
10	15	16	15	17	40	12	9	15	12	70	6	6	7	7	100	8	12	6	6	
12	14	12	8	9	42	4	5	5	4	72	9	13	7	6	102	16	15	14	13	
14	4	4	4	6	44	6	8	8	11	74	16	16	17	15	104	15	8	16	17	
16	6	7	13	16	46	13	13	14	17	76	16	12	13	13	106	4	3	6	6	
18	16	17	16	17	48	15	14	8	9	78	6	5	5	3	108	7	11	8	6	
20	18	19	5	6	50	5	3	4	7	80	6	8	9	9	110	15	18	17	15	
22	10	8	9	9	52	4	5	13	16	82	13	17	15	17	112	17	17	14	12	
24	6	6	17	16	54	9	13	16	17	84	17	17	7	9	114	11	6	5	4	
26	9	12	13	11	56	15	17	5	5	86	13	7	5	6	116	5	8	10	11	
28	15	17	5	5	58	15	9	9	8	88	4	4	14	16	118	9	15	10	18	
Průměr CPU [%]	LM				PM				BMI				BM2				Měřeno pro počet křivek rovný 1			
	10,47				10,6				10,8				10,92							

Tab. A.2: Naměřené a vypočítané hodnoty procentuálního zatížení procesoru v čase pro různé módy křivek převodní funkce dynamického procesoru.

	LM	PM	BM1	BM2		LM	PM	BM1	BM2		LM	PM	BM1	BM2		LM	PM	BM1	BM2	
t	CPU	CPU	CPU	CPU	t	CPU	CPU	CPU	CPU	t	CPU	CPU	CPU	CPU	t	CPU	CPU	CPU	CPU	
[s]	[%]	[%]	[%]	[%]	[s]	[%]	[%]	[%]	[%]	[s]	[%]	[%]	[%]	[%]	[s]	[%]	[%]	[%]	[%]	
0	15	12	16	15	30	15	17	9	18	60	5	5	6	9	90	6	11	8	3	
2	16	16	6	14	32	5	11	2	7	62	5	4	8	6	92	15	16	17	11	
4	7	13	8	6	34	6	5	12	8	64	11	12	15	15	94	18	13	14	18	
6	6	5	17	11	36	15	11	16	15	66	15	15	8	17	96	12	6	6	7	
8	11	7	16	17	38	17	17	4	13	68	10	8	4	6	98	6	8	9	7	
10	17	17	5	9	40	13	17	8	6	70	2	3	13	10	100	8	16	15	17	
12	14	17	11	6	42	6	7	17	13	72	5	10	16	16	102	17	17	7	15	
14	4	9	17	13	44	8	6	13	17	74	15	15	4	11	104	17	6	4	6	
16	4	5	11	15	46	17	14	5	8	76	17	15	10	5	106	6	5	13	12	
18	13	10	7	3	48	18	16	13	6	78	6	4	18	13	108	5	10	14	17	
20	16	15	16	8	50	9	11	18	13	80	4	6	11	17	110	9	15	5	9	
22	7	15	17	16	52	5	5	8	13	82	12	16	6	6	112	15	9	11	7	
24	4	4	5	11	54	11	9	6	3	84	17	18	13	5	114	12	3	17	16	
26	7	3	7	6	56	18	17	17	9	86	17	9	18	15	116	2	6	10	17	
28	15	12	15	14	58	17	17	15	17	88	5	5	7	11	118	4	15	6	4	
Průměr CPU [%]	LM				PM				BMI				BM2				Měřeno pro počet křivek rovný 8			
	10,57				10,68				10,83				10,97							

Tab. A.3: Naměřené a vypočítané hodnoty procentuálního zatížení procesoru v čase pro měření úrovně hlasitosti dle doporučení ITU-R BS.1770-1.

Délka okna 1,45ms; Překrytí 0%						Délka okna 1,45ms; Překrytí 75%						Délka okna 1s; Překrytí 0%						Délka okna 1s; Překrytí 75%					
<i>t</i>	<i>CPU</i>	<i>t</i>	<i>CPU</i>	<i>t</i>	<i>CPU</i>	<i>t</i>	<i>CPU</i>	<i>t</i>	<i>CPU</i>	<i>t</i>	<i>CPU</i>	<i>t</i>	<i>CPU</i>	<i>t</i>	<i>CPU</i>	<i>t</i>	<i>CPU</i>	<i>t</i>	<i>CPU</i>	<i>t</i>	<i>CPU</i>		
[s]	[%]	[s]	[%]	[s]	[%]	[s]	[%]	[s]	[%]	[s]	[%]	[s]	[%]	[s]	[%]	[s]	[%]	[s]	[%]	[s]	[%]	[s]	[%]
0	14	40	5	80	8	0	14	40	4	80	12	0	14	40	6	80	15	0	17	40	6	80	11
2	11	42	4	82	15	2	12	42	6	82	14	2	10	42	11	82	13	2	14	42	8	82	14
4	2	44	13	84	6	4	2	44	14	84	4	4	6	44	14	84	5	4	5	44	14	84	2
6	8	46	13	86	4	6	8	46	12	86	5	6	13	46	9	86	11	6	9	46	11	86	6
8	14	48	2	88	12	8	15	48	2	88	14	8	16	48	7	88	17	8	15	48	2	88	14
10	7	50	6	90	15	10	7	50	8	90	15	10	6	50	14	90	10	10	8	50	8	90	11
12	3	52	16	92	5	12	3	52	15	92	5	12	8	52	18	92	5	12	3	52	15	92	3
14	13	54	12	94	7	14	12	54	11	94	10	14	17	54	5	94	11	14	11	54	7	94	11
16	15	56	5	96	15	16	15	56	4	96	15	16	14	56	5	96	15	16	14	56	4	96	15
18	7	58	11	98	15	18	6	58	13	98	11	18	4	58	13	98	3	18	4	58	13	98	9
20	7	60	15	100	6	20	6	60	16	100	5	20	10	60	12	100	6	20	5	60	16	100	6
22	15	62	9	102	11	22	15	62	6	102	14	22	15	62	2	102	14	22	14	62	5	102	13
24	14	64	7	104	15	24	13	64	7	104	17	24	7	64	9	104	10	24	14	64	8	104	15
26	5	66	15	106	11	26	5	66	16	106	6	26	2	66	14	106	2	26	5	66	14	106	6
28	10	68	16	108	5	28	11	68	16	108	6	28	11	68	7	108	10	28	11	68	12	108	9
30	16	70	5	110	12	30	17	70	5	110	14	30	15	70	2	110	15	30	15	70	5	110	16
32	12	72	7	112	14	32	11	72	9	112	12	32	3	72	11	112	8	32	10	72	13	112	13
34	6	74	15	114	4	34	6	74	14	114	2	34	4	74	15	114	6	34	6	74	17	114	4
36	10	76	10	116	4	36	11	76	8	116	8	36	14	76	6	116	13	36	14	76	8	116	10
38	14	78	2	118	13	38	17	78	3	118	14	38	12	78	8	118	15	38	17	78	8	118	15
Průměr CPU [%]			9,8			Průměr CPU [%]			9,97			Průměr CPU [%]			9,88			Průměr CPU [%]			10,13		

Tab. A.4: Naměřené a vypočítané hodnoty procentuálního zatížení procesoru v čase pro měření skutečné špičkové hodnoty dle doporučení ITU-R BS.1770-1.

<i>t</i> [s]	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	Průměr CPU [%]
<i>CPU</i> [%]	9	17	10	6	13	16	7	9	17	14	6	9	17	10	7	15	17	6	11	17	
<i>t</i> [s]	40	42	44	46	48	50	52	54	56	58	60	62	64	66	68	70	72	74	76	78	11,47
<i>CPU</i> [%]	11	5	14	16	7	7	16	14	5	11	17	11	6	15	17	6	9	18	13	5	
<i>t</i> [s]	80	82	84	86	88	90	92	94	96	98	100	102	104	106	108	110	112	114	116	118	
<i>CPU</i> [%]	12	16	8	7	17	15	5	9	17	11	5	14	18	7	8	17	15	6	10	15	

Tab. A.5: Celkové zatížení pro otevřené a zavřené okno uživatelského rozhraní.

<i>t</i> [s]	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	Průměr CPU_o [%]
<i>CPU_o</i> [%]	20	21	16	9	15	22	22	13	10	15	20	19	9	10	18	21	16	9	8	18	
<i>CPU</i> [%]	13	17	15	6	8	15	19	10	7	11	18	18	7	8	14	17	15	8	9	18	
<i>t</i> [s]	40	42	44	46	48	50	52	54	56	58	60	62	64	66	68	70	72	74	76	78	15,72
<i>CPU_o</i> [%]	20	12	12	15	20	21	11	13	20	23	18	10	13	18	20	12	9	15	20	20	
<i>CPU</i> [%]	20	11	8	13	17	17	6	7	15	19	11	6	9	17	17	7	6	13	17	16	
<i>t</i> [s]	80	82	84	86	88	90	92	94	96	98	100	102	104	106	108	110	112	114	116	118	12,57
<i>CPU_o</i> [%]	8	9	15	20	17	11	13	21	22	14	11	17	23	22	12	13	17	20	16	9	
<i>CPU</i> [%]	8	11	19	21	11	8	13	17	16	6	8	16	18	10	7	11	18	18	6	7	

Tab. A.6: Rozsah hodnot procentuálního zatížení procesoru při uživatelské interakci za vypnutého zpracování signálu.

Pohyb kontrolním bodem				Pohyb kontrolním bodem				Pohyb táhlem
LM	PM	BM1	BM2	LM	PM	BM1	BM2	
<i>CPU_o</i> [%]	<i>CPU_o</i> [%]	<i>CPU_o</i> [%]	<i>CPU_o</i> [%]	<i>CPU_o</i> [%]	<i>CPU_o</i> [%]	<i>CPU_o</i> [%]	<i>CPU_o</i> [%]	<i>CPU_o</i> [%]
1	1-2	4-5	4-6	1-3	4	5-6	4-6	4-6
Počet křivek = 1				Počet křivek = 8				

B OBSAH PŘILOŽENÉHO CD

- Adresář se zdrojovými soubory v jazyce C++ a projektem ve vývojovém prostředí Microsoft Visual Studio 2005 (`CppProjekt`)
- Adresář s testovacími zdrojovými soubory prostředí Matlab (`Matlab`)
- VST plug-in modul (`83591.dll`)
- Text diplomové práce (`83591.pdf`)
- Mapa CD (`Ctime.txt`)
- Náповěda k programu (`Napoveda.pdf`)