



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

**KOMPRESSE SIGNÁLŮ EKG NASNÍMANÝCH POMOCÍ
MOBILNÍHO ZAŘÍZENÍ**

COMPRESSION OF ECG SIGNALS RECORDED USING MOBILE ECG DEVICE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Filip Had

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Andrea Němcová

BRNO 2017

Diplomová práce

magisterský navazující studijní obor Biomedicínské inženýrství a bioinformatika

Ústav biomedicínského inženýrství

Student: Bc. Filip Had

ID: 155574

Ročník: 2

Akademický rok: 2016/17

NÁZEV TÉMATU:

Komprese signálů EKG nasnímaných pomocí mobilního zařízení

POKYNY PRO VYPRACOVÁNÍ:

1) Seznamte se s problematikou komprese signálů EKG snímaných pomocí mobilních zařízení. 2) Navrhněte a vytvořte vhodný pokročilý algoritmus pro kompresi a rekonstrukci signálu EKG. 3) Otestujte algoritmus na standardní databázi signálů EKG (např. CSE nebo MIT-BIH Arrhythmia Database) a optimalizujte jej. 4) Vytvořte databázi signálů EKG a signálů z akcelerometru paralelně nasnímaných pomocí mobilního zařízení. 5) Otestujte kompresní algoritmus na vytvořené databázi, diskutujte výsledky a porovnejte je s výsledky testování na standardní databázi, výsledky také porovnejte s dalšími autory. 6) Zjistěte, zda by se signál z akcelerometru dal využít pro řízení komprese signálu EKG.

DOPORUČENÁ LITERATURA:

[1] MANIKANDAN, M. Sabarimalai a S. DANDAPAT. Wavelet-based electrocardiogram signal compression methods and their performances: A prospective review. Biomedical Signal Processing and Control [online]. 2014, 14(November 2014), 73-107. DOI: 10.1016/j.bspc.2014.07.002. ISSN 17468094.

[2] SALOMON, D. Data Compression: The Complete Reference. 4th ed. London: Springer, 2007. ISBN 18-46-8602-6.

Termín zadání: 6.2.2017

Termín odevzdání: 19.5.2017

Vedoucí práce: Ing. Andrea Němcová

Konzultant:

prof. Ing. Ivo Provazník, Ph.D.
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.

Fakulta elektrotechniky a komunikačních technologií, Vysoké učení technické v Brně / Technická 3058/10 / 616 00 / Brno

ABSTRAKT

Kompresí signálů je nezbytnou součástí pro snímání EKG, z důvodu poměrně velkého objemu dat, která je nutno především bezdrátově posílat na analýzu. Právě kvůli bezdrátovému odesílání je nutné objem dat co nejvíce minimalizovat. K minimalizaci objemu slouží bezztrátové nebo ztrátové kompresní algoritmy. Tato práce popisuje algoritmus SPIHT a nově vytvořenou experimentální metodu, založenou na kompresi pomocí PNG, a jejich testování. V této diplomové práci je také uvedena banka signálů EKG s paralelně snímanými daty z akcelerometru. V poslední části práce je rozebrána a realizována úprava algoritmu SPIHT, využívající data akcelerometru.

KLÍČOVÁ SLOVA

Kompresní metody, Huffmanovo kódování, SPIHT, PNG, ztrátová komprese, bezztrátová komprese, akcelerometr, mobilní EKG záznamník

ABSTRACT

Signal compression is necessary part for ECG scanning, because of relatively big amount of data, which must be transmitted primarily wirelessly for analysis. Because of the wireless sending it is necessary to minimize the amount of data as much as possible. To minimize the amount of data, lossless or lossy compression algorithms are used. This work describes an algorithm SPITH and newly created experimental method, based on PNG, and their testing. This master's thesis there is also a bank of ECG signals with parallel sensed accelerometer data. In the last part, modification of SPIHT algorithm, which uses accelerometer data, is described and realized.

KEYWORDS

Compression methods, Huffman coding, SPIHT, PNG, lossy compression, lossless compression, accelerometer, mobile ECG device

HAD, F. Komprese signálů EKG nasnímaných pomocí mobilního zařízení. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 66 s. Vedoucí diplomové práce Ing. Andrea Němcová.

PROHLÁŠENÍ

Prohlašuji, že svoji diplomovou práci na téma Komprese signálů EKG nasnímaných pomocí mobilního zařízení 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/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji mé vedoucí diplomové práce Ing. Andrei Němcové za odbornou pomoc a poskytnutí cenných rad při zpracování mé diplomové práce

Obsah

Úvod	1
1 Elektroardiogram	2
1.1 Způsoby záznamu EKG	2
1.2 Vlny EKG	3
1.3 Intervaly EKG.....	4
2 Komprese	5
2.1 Bezeztrátová komprese	5
2.1.1 Metoda potlačení nul	5
2.1.2 Proudové kódování	6
2.1.3 Huffmanovo kódování	6
2.1.4 Aritmetické kódování	8
2.1.5 Metoda LZ77	10
2.1.6 Metoda založená na ASCII kódování	11
2.1.7 Metoda založená na K-means clusteringu	13
2.2 Ztrátová komprese	14
2.2.1 Algoritmus Turning point	14
2.2.2 Algoritmus AZTEC	14
2.2.3 Kódování založené na lineární predikci vlnek.....	15
2.2.4 Metoda založená na vlnkové paketové transformaci	16
3 SPIHT	18
3.1 Spojitá vlnková transformace	18
3.2 Vlnková transformace s diskrétním časem	18
3.3 SPIHT algoritmus	20
3.3.1 SPIHT kódování příklad	22
3.3.2 SPIHT dekódování příklad	23
4 Hodnocení kvality	25
4.1 PRD – Percent root mean square difference	25
4.2 PRDN – Normalized percent root mean square difference	26
4.3 NMAE – Normalized maximum amplitude error.....	26

5	Výsledky	26
5.1	Rozdíly mezi použitými vlnkami.....	27
5.2	Rozdíly mezi stupni rozkladu	30
5.3	Zhodnocení	31
6	Experimentální část	34
6.1	PNG komprese	34
6.1.1	Delta filtrace	34
6.1.2	Deflate.....	35
6.2	Princip experimentální metody	35
6.3	Výsledky experimentální metody	36
7	Banka signálů	38
8	Testování nasnímané banky signálů	43
8.1	Testování algoritmu SPIHT na vytvořené bance signálů	43
8.2	Testování experimentální metody na vytvořené bance signálů	47
9	Využití dat z akcelerometru pro kompresi	50
9.1	Úprava algoritmu SPIHT pro využití dat z akcelerometru	50
9.2	Testování upraveného algoritmu SPIHT	51
10	Návody k použití gui	57
10.1	Experimentální metoda	57
10.2	Upravený algoritmus SPIHT	59
11	závěr	63
	Literatura	64

SEZNAM OBRÁZKŮ

Obr. 1 – Popis reálné EKG křivky - vlny	3
Obr. 2 – Popis reálné EKG křivky - intervaly	4
Obr. 3 – První krok Huffmanova stromu	6
Obr. 4 – Druhý krok Huffmanova stromu	7
Obr. 5 – Výsledný Huffmanův strom	7
Obr. 6 – Rychlá dyadická DTWT (převzato z [16])	19
Obr. 7 – Stromová struktura vstupních dat	20
Obr. 8 – Rozložený příkladový signál (převzato z [15])	22
Obr. 9 – Výsledky rekonstruovaného signálu 12621_01m pomocí vlnek haar a bior4.4 (červeně – rekonstruovaný signál, modře – originální signál, zeleně – chybový signál).....	28
Obr. 10 - Výsledky rekonstruovaného signálu 11442_03m pomocí vlnek haar a bior4.4 (červeně – rekonstruovaný signál, modře – originální signál, zeleně – chybový signál).....	29
Obr. 11 – Kvalitativní rozdíly signálů 1142_03m a 12621_01m pro různé stupně rozkladu.	31
Obr. 12 – Závislost PRDN na kompresním poměru	32
Obr. 13 – Sloupcový graf pro výběr délky bufferu.....	33
Obr. 14 – Výsledný obrázek po úpravách.....	36
Obr. 15 – Závislost kompresního poměru na signálu	37
Obr. 16 – Výřez signálu EKG s akcelerometrem při provádění kliků.....	40
Obr. 17 – Výřez signálu EKG s akcelerometrem při spánku v klidu	41
Obr. 18 – Výřez signálu EKG s akcelerometrem při změně stylu jízdy na kole	42
Obr. 19 – Sloupcové grafy pro výběr vlnky	43
Obr. 20 – Sloupcové grafy pro výběr stupně rozkladu	44
Obr. 21 – Sloupcové grafy pro výběr délky bufferu.....	45
Obr. 22 – Výběr délky bufferu pro experimentální metodu	48
Obr. 23 – Porovnání originálního signálu beh.mat s rekonstruovaným pomocí upraveného algoritmu SPIHT	53
Obr. 24 – Přiblížený průběh akcelerometru pro signál beh.mat	54
Obr. 25 – Porovnání originálního signálu kliky.mat s rekonstruovaným pomocí upraveného algoritmu SPIHT	55
Obr. 26 – Kompletní průběh akcelerometru pro signál kliky.mat	55

Obr. 27 - Porovnání originálního signálu spanek_klid.mat s rekonstruovaným pomocí upraveného algoritmu SPIHT	56
Obr. 28 – Úvodní obrazovka GUI experimentální metody.....	57
Obr. 29 – Vlevo – načtený signál z cdb, Vpravo – načtený signál z vytvořené banky signálů.....	58
Obr. 30 – GUI experimentálního algoritmu, karta „Dekomprese“	58
Obr. 31 – GUI experimentálního algoritmu – dokončená dekomprese.....	59
Obr. 32 – Úvodní obrazovka GUI pro upravený algoritmus SPIHT	60
Obr. 33 – Upravený algoritmus SPIHT – vlevo - průběh komprese, vpravo – komprese dokončena	61
Obr. 34 – Upravený algoritmus SPIHT – vlevo waitbar dekomprese, vpravo dekomprimovaný signál.....	61
Obr. 35 – Upravený algoritmus SPIHT – Dekomprimovaný signál s originálem.....	62

SEZNAM TABULEK

Tab. 1 – Tabulka četností znaků	6
Tab. 2 – Abeceda aritmetického kódování	8
Tab. 3 – Výpočet algoritmu aritmetického kódování	9
Tab. 4 – Výpočet algoritmu aritmetického dekódování	10
Tab. 5 – Příklad kódování metodou LZ77	10
Tab. 6 – Kompresní poměry a časy algoritmu ASCII [7].....	12
Tab. 7 – Výsledky kódování, založeném na K-means clusteringu [8]	13
Tab. 8 – Výsledky kódování, založeném na lineární predikci [10]	16
Tab. 9 – Výsledky kódování metody založené na vlnkové paketové transformaci převzato z [11]	17
Tab. 10 – Výstup z algoritmu SPIHT	24
Tab. 11 – Hodnoty PRDN [%] pro jednotlivé vlnky při CR = 1/5	27
Tab. 12 – Výpočetní časy při použití jednotlivých vlnek	30
Tab. 13 – Časové a kvalitativní rozdíly mezi počty stupňů rozkladu	30
Tab. 14 – Porovnání experimentálního algoritmu s ostatními	37
Tab. 15 – Banka signálů	39
Tab. 16 – Výsledné časy a PRDN po nastavení vybraných parametrů	46
Tab. 17 – Výsledné časy a PRDN před nastavením vybraných parametrů	47
Tab. 18 – Výsledky experimentálního algoritmu na vytvořené bance signálů.....	49
Tab. 19 – Výsledky upraveného algoritmu SPIHT.....	52

ÚVOD

Kompresie dat je nezbytnou součástí většiny přístrojů, které snímají a následně odesílají data, zejména pak přístroje přenosné. U přenosných přístrojů EKG jsou vysoké nároky na minimalizaci. Z důvodu vysoké minimalizace a jednoduchosti používání, jsou v podstatě všechny mobilní EKG jednokanálové, s výjimkou Holterovského. Jelikož je nutné z těchto přenosných přístrojů data odesílat ideálně bezdrátově, je zapotřebí dostatečné komprese dat pro urychlení procesu.

V první části diplomové práce byly popsány způsoby, jakými lze signál EKG získávat. Následně byly popsány jednotlivé útvary, které se v křivce vyskytují.

V druhé části je uvedeno několik jednoduchých bezztrátových kompresních algoritmů s uvedenými příklady, po nichž jsou uvedeny komplexnější bezztrátové algoritmy. V práci jsou dále uvedeny jednoduché algoritmy pro ztrátovou kompresi, následované algoritmy pro ztrátovou kompresi s využitím vlnkové, respektive vlnkové paketové transformace.

Ve třetí části je rozebrána vlnková transformace a dopodrobna ztrátový algoritmus SPIHT, včetně postupu pro kompresi a dekompresi signálu. Dále jsou rozebrány výsledky navrženého algoritmu v prostředí MATLAB.

Ve čtvrté části diplomové práce byla navržena nová experimentální metoda komprese, založená na kombinaci proudového kódování a PNG formátu, který je zde rovněž popsán.

Ve páté části je jednoduchý popis naměřené banky EKG signálů s paralelně snímanými daty z akcelerometru. V této části jsou dva popisované algoritmy rovněž testovány právě na dané bance signálů.

V poslední části diplomové práce je popsána a otestován upravený algoritmus SPIHT, tak aby využíval pro nastavení komprese data z akcelerometru. Testování probíhalo rovněž na naměřené bance signálů. V poslední podkapitole, která se věnuje právě testování navrženého algoritmu, jsou popsány příklady výhod a nevýhod.

1 ELEKTROKARDIOGRAM

Elektrokardiogram (dále EKG) je závislost rozdílů akčních potenciálů šířících se po srdci v závislosti na čase. Na EKG se rozlišuje několik úseků a vln, které v této kapitole budou popsány. Samotné EKG se měří pomocí elektrod umístěných na těle pacienta. Signál EKG se využívá pro určování srdečních onemocnění, popřípadě poškození srdce po infarktu myokardu. Samotný akční potenciál, jehož rozdíly měříme, vzniká v sinoatriálním uzlu, tudíž bývá označován jako primární pacemaker [1].

1.1 Způsoby záznamu EKG

Nejčastějším způsobem snímání EKG je standardní 12svodové EKG, či ortogonální svodový systém.

12svodové EKG je nejčastěji využívaným svodovým systémem v klinické praxi. Tento systém tvoří 3 končetinové elektrody (Einthovenovy) a 6 elektrod hrudních (Wilsonovy), pomocí končetinových elektrod se počítají 3 svody unipolární (Goldbergovy). Všechny svody jsou měřeny na povrchu těla na standardizovaných místech [1].

Bipolární svody se označují římskými číslicemi I, II a III a jsou měřeny mezi levou a pravou paží a levou nohou. Název Einthovenovy získaly tyto svody díky svému objeviteli, Willemu Einthovenovi, který určil právě tato umístění na těle jako nejvhodnější. Umístění mezi levou a pravou rukou a levou nohou tvoří takzvaný Einthovenův trojúhelník, což je rovnostranný trojúhelník se srdcem uprostřed [1].

Wilsonovy hrudní svody jsou označovány V1 až V6 a popisují detailní pohled na aktivitu srdce v různých rovinách. Tyto svody jsou umístěny na těle pacienta od pravé přes levou stranu hrudní kosti po levou zadní část hrudníku. Jelikož tyto svody jsou unipolární, mají vztažnou svorku, vytvořenou pomocí spojení končetinových elektrod přes odpor 5 k Ω [1].

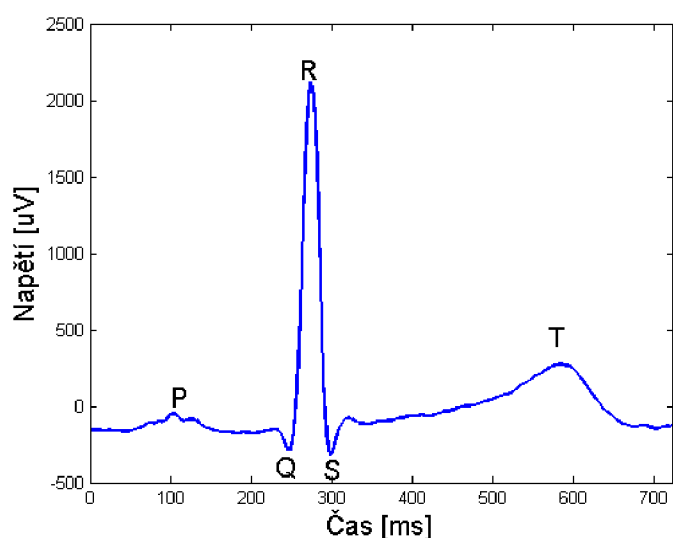
Goldbergovy svody se označují aVL, aVR a aVF, a jsou vypočítány jako rozdíl potenciálů na jedné z končetinových elektrod a průměru zbývajících dvou. Tyto svody jsou určeny k vyplnění informací, které není možné získat z Einthovenových svodů [1].

Ortogonální svodový systém poskytuje celoprostorový vjem. Základním principem tohoto systému je využití několika elektrod k vytvoření jednoho svodu. Elektrody jsou

následně napojeny na odporovou síť, ze které se odvozuji signály U_x, U_y a U_z . Nejznámější zapojením ortogonálního systému je systém Frankův, obsahující 7 elektrod, přičemž 5 elektrod je umístěných po obvodu hrudníku, jedna elektroda na noze a druhá v jícnu, popřípadě na zadní straně krku [2].

1.2 Vlny EKG

Pro diagnózu srdečních arytmií, nebo jiných onemocnění se využívá rozměření signálu EKG a to tak, že se měří doba trvání a výchylka jednotlivých vln a úseků. Na Obr. 1 Je příklad reálně naměřené křivky s označenými vlnami.



Obr. 1 – Popis reálné EKG křivky - vlny

Jako první se v EKG křivce objevuje **Vlna P**, která je způsobena depolarizací pravé a levé síně. Výchylka samotné vlny P ve většině případů nepřekračuje 300 μV , což v případě reálně naměřeného signálu na Obr. 1 odpovídá (120 μV). Dalším sledovaným parametrem vlny P je doba trvání, která bývá menší než 120 ms, v případě reálného signálu je přesně 120 ms. Repolarizace pravé a levé síně, v EKG signálu není vidět, jelikož probíhá ve stejném intervalu jako **komplex QRS**, tudíž je tímto výrazným útvarem zastíněna.

Při posunu po EKG křivce je další výrazný útvar, zvaný **komplex QRS**, který je způsoben depolarizací pravé a levé komory. V případě zdravého srdce, je trvání QRS komplexu mezi 70-110 ms, v případě reálného signálu se jedná přibližně o 100 ms. QRS komplex se skládá ze tří výrazných kmitů, podle kterých má i svůj název. Jako první je negativní **vlna Q**, následovaná výraznou pozitivní **vlnou R** a jako poslední část

komplexu je opět negativní **vlna S**. QRS komplex je nejvýraznějším útvarem EKG křivky, s velikostí 2-3 mV, v případě reálného signálu je napětí přibližně 2 500 μV [1].

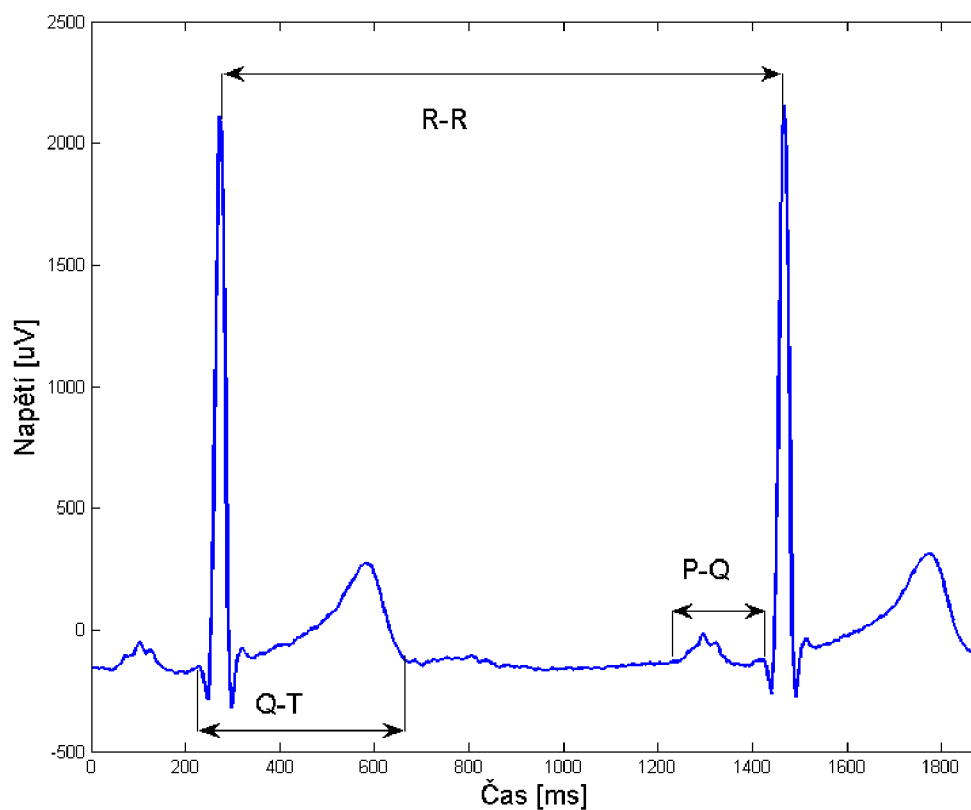
Poslední výrazná vlna v EKG křivce, vyskytující se v intervalu 300 ms za komplexem QRS, je vlna označovaná jako **T**, která je způsobena repolarizací komor. Interval mezi QRS komplexem a vlnou T a šířka vlny T bývá silně ovlivněn tepovou frekvencí (se zvyšující se tepovou frekvencí se interval a šířka zužují) [1]. V některých případech se za vlnou T objevuje i vlna U, jejíž původ je dosud neznámý, avšak existují tři teorie původu vlny U:

- Opožděná repolarizace Purkyňových vláken
- Prodloužená repolarizace midmyokardiálních buněk
- Potenciál vzniklý mechanickými silami ve stěně komory [3]

Jak je vidět na reálně naměřeném signálu se vlna U nevyskytuje.

1.3 Intervaly EKG

Dalším parametrem, kromě délek samotných vln, vyskytujících se v signálu EKG jsou délky intervalů mezi jednotlivými vlnami.



Obr. 2 – Popis reálné EKG křivky - intervaly

Jako první se měří **interval RR**, jelikož je poměrně jednoduché detekovat vlny R a z délky tohoto intervalu se zjišťuje aktuální tepová frekvence a popřípadě variabilita srdečního rytmu, pomocí které lze zjistit různé srdeční choroby.

Dále se měří **interval PQ**, který vyjadřuje dobu od síňové depolarizace po komorovou depolarizaci. Tento interval představuje šíření akčního potenciálu z primárního pacemakeru (sinoatriální uzel) do srdečních komor.

Jako poslední se měří **interval QT**, který vyjadřuje část křivky od začátku depolarizace až po konec repolarizace komor. Jak bylo řečeno v předchozí kapitole, jelikož vlna T je silně závislá na tepové frekvenci, je i interval QT závislý na tepové frekvenci [1].

2 KOMPRESSE

Kompresse signálů se dá rozdělit do dvou základních skupin. První skupinou je komprese bezztrátová. Tento typ komprese probíhá bez jakýchkoliv ztrát dat avšak zpravidla za cenu menšího kompresního poměru. Druhá skupina je komprese ztrátová, která je využívána častěji z důvodu velkého kompresního poměru a lze dosáhnout poměrně malé chyby v rekonstrukci [4].

2.1 Bezeztrátová komprese

Typickými zástupci bezztrátové komprese jsou například: potlačení nul (null suppression), proudové kódování nebo Huffmanovo kódování. Bezztrátová komprese se rovněž používá pro redukci objemu dat komprimovaných pomocí komprese ztrátové.

2.1.1 Metoda potlačení nul

Metoda je založena na vyhledávání mezer, či nul v posloupnosti čísel a následné odeslání indexu a hodnoty za nulami. Příkladem může být posloupnost: 100002001005023, která se po zakódování odešle jako: {1,1},{6,2},{9,1},{12,5},{14,2},{15,3}. Z výsledku je patrné, že pokud není v proudu dat větší množství nul, tak se tato metoda nevyplatí [5]. Kompresní poměr této metody nebývá udáván, jelikož se může pohybovat v libovolném rozmezí, v závislosti na signálu.

2.1.2 Proudové kódování

Tento typ kódování byl odvozen z metody potlačení nul, avšak nevyhledává pouze opakující se nuly, ale všechna opakující se čísla, či znaky. Výstupem z tohoto kodéru jsou tzv. zarážky (indikátory – v příkladu označeny jako *i*), následované opakovanou hodnotou a počtem opakování. V případě, že se hodnota neopakuje, odešle se na výstup pouze daná hodnota [5]. Např.:

Vstup do kodéru: 55511000012143334888888888

Výstup z kodéru: i53i12i041214i334i89

2.1.3 Huffmanovo kódování

Huffmanův kodér je rovněž založen na počtu opakování jednotlivých čísel, či znaků, s tím rozdílem, že není nutné, aby znaky byly za sebou a i jejich následné zpracování se od uvedených metod liší. Metoda spočívá v kódování znaku určitým počtem bitů v závislosti na četnosti znaků, toho se docílí konstrukcí takzvaného binárního stromu [5].

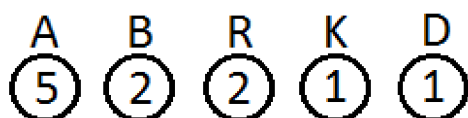
Příklad:

Vstupem do kodéru je posloupnost znaků „ABRAKADABRA“

Tab. 1 – Tabulka četností znaků

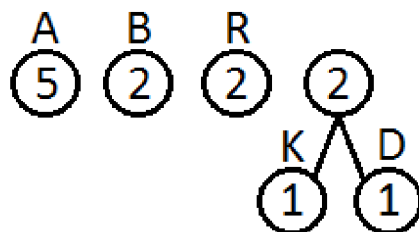
Znak:	A	B	R	K	D
Četnost	5	2	2	1	1

Následně se podle Tab. 1 vytvoří strom, který po prvním kroku vypadá následovně:



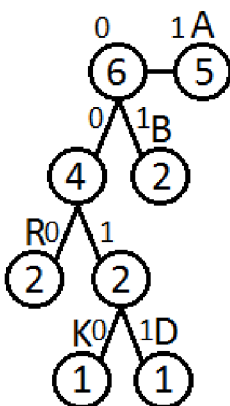
Obr. 3 – První krok Huffmanova stromu

V kruhu jsou zobrazeny četnosti znaků ve slově. Nyní se vyberou dva prvky s nejmenší četností a sloučí se do jednoho uzlu a jejich četnost se sečte:



Obr. 4 – Druhý krok Huffmanova stromu

Následně se opět vyberou dva prvky s nejmenší četností v nejvyšším patře a jejich četnost se sečte. Tyto kroky se opakují, dokud nevznikne následující strom, u něhož si jednotlivé větve označíme 0 a 1:



Obr. 5 – Výsledný Huffmanův strom

Po vytvoření stromu jako je na Obr. 5, se pokračuje rekonstrukcí kódu pro náš text následovně: **A** bylo nejčetnější písmeno, tudíž musí být kódováno nejmenším počtem bitů, v tomto případě 1 bit s hodnotou 1, **B** byl druhý nejčetnější znak, tudíž by měl mít jako druhý nejmeně bitů, v tomto případě 2 a jedná se o 01. Pro všechny znaky vzniká kód tím způsobem, že se jde po cestě, která se skládá z jednotlivých větví, označených „0“ nebo „1“. Jakmile dojdeme k požadovanému znaku, cesta z „0“ a „1“ se vypíše jako kód. Pro zadaný textový řetězec ABRAKADABRA vznikne kód 1 01 000 1 0010 1 0011 1 01 000 1.

Z popsaného principu vyplývá, že je nutno synchronizovat stromy (slovníky) mezi přijímací a odesílací stranou, jelikož **K** a **D** by šlo zaměnit, to stejné pro **B** a **R**. Z uvedených důvodů bývá strom odeslán společně se zakódovanými daty.

2.1.4 Aritmetické kódování

Jedná se rovněž jako u Huffmanova kódování o statistickou metodu, která kóduje jedno kódované slovo jako jedno desetinné číslo. Principem je zápis slova jako desetinného čísla z pevně daného intervalu, který se pomocí následujícího algoritmu vypočítá [6].

Kódování:

Obě strany (kódující i dekódující) musí znát abecedu, pomocí které kódování, či dekódování probíhá. Např.:

Tab. 2 – Abeceda aritmetického kódování

Písmeno	Rozsah
A	<0, 0.2)
B	<0.2, 0.3)
C	<0.3, 0.4)
D	<0.4, 0.5)
E	<0.5, 0.75)
F	<0.75, 0.8)
G	<0.8, 0.85)
H	<0.85, 1)

Následně je zvoleno kódované slovo např.: BAFA.

Poté již probíhá výpočet dolní (L) a horní (H) meze intervalu o délce R podle vzorců:

$$H = L + R * (\text{horní_hodnota_intervalu_znaku}), \quad (1)$$

$$L = L + R * (\text{spodní_hodnota_intervalu_znaku}), \quad (2)$$

kde L , H a R jsou spodní a horní meze výsledného intervalu a rozsah z předchozího kroku, dle zvoleného slova je výpočet následující:

Inicializační hodnoty: $L = 0$ a $H = 1 \Rightarrow R = 1$

Tab. 3 – Výpočet algoritmu aritmetického kódování

Vstup	Výpočet	Předchozí R	L	H
-	-		0	1
B	H=0+1*0.2=0.2 L=0+1*0.3=0.3	1	0.2	0.3
A	H=0.2+0.1*0=0.2 L=0.2+0.1*0.2=0.22	0.1	0.2	0.22
F	H=0.2+0.02*0.75=0.215 L=0.2+0.02*0.8=0.216	0.02	0.215	0.216
A	H=0.215+0.001*0=0.215 L=0.215+0.001*0.2=0.2152	0.001	0.215	0.2152

Výsledný interval, kódující slovo BAFA je $(0.215, 0.2152)$, to znamená, že na výstup z algoritmu můžeme poslat libovolné číslo z tohoto intervalu, např.: 0,215.

Dekódování:

Jak již bylo řečeno, je nutno aby i dekodovací strana znala abecedu uvedenou v Tab. 2. Následně se z abecedy vybere interval, do kterého vstupní číslo patří a proběhne výpočet dalšího vstupního čísla podle následujícího vzorce:

$$n = \frac{n - l_{znaku}}{R_{znaku}}, \quad (3)$$

kde n je vstupní číslo, l_{znaku} je spodní hranice intervalu znaku a R_{znaku} je rozsah intervalu daného znaku.

Tab. 4 – Výpočet algoritmu aritmetického dekódování

n - vstupní	Interval znaku:	Výpočet dalšího n
0.215	B <0.2, 0.3)	$n = \frac{0.215 - 0.2}{0.1} = 0,15$
0.15	A <0, 0.2)	$n = \frac{0.15 - 0}{0.2} = 0,75$
0,75	F <0.75,0.8)	$n = \frac{0.75 - 0.75}{0.05} = 0$
0	A <0, 0.2)	-

Z výpočtu je patrná jistá nevýhoda algoritmu, kterou je absence ukončení, to znamená, že algoritmus neví, kdy nastává konec řetězce. V případě, že by probíhal výpočet z následujících hodnot, výsledné slovo by mohlo být BAFA, ale i BAFAA, BAFAAA atd. Z uvedeného důvodu se většinou využívá ukončovacího znaku, na který když dekodér narazí, ukončí dekódování.[6]

2.1.5 Metoda LZ77

LZ77 je metoda s tzv. klouzavým oknem, které se skládá ze dvou částí, těmi jsou: vyhledávací okno a sledující okno. Výstupem pak jsou tzv. tokeny, které jsou rozděleny na 3 části a obsahují veškeré informace o vstupním řetězci. Token má podobu 3 znaků, přičemž první znak udává vzdálenost řetězce od konce vyhledávacího okna, druhý znak udává délku stejného řetězce a třetí znak je písmeno, které se na konec řetězce umístí [13].

Tab. 5 – Příklad kódování metodou LZ77

Krok	Vyhledávací okno	Sledující okno	Doplněný znak		Výstup
1		sir_sid_eastman	s	=>	(0,0,s)
2	s	ir_sid_eastman_	i	=>	(0,0,i)
3	si	r_sid_eastman_e	r	=>	(0,0,r)
4	sir	_sid_eastman_ea	_	=>	(0,0,_)
5	sir_	sid_eastman_eas	sid	=>	(4,2,d)
6	sir_sid	_eastman_easily	_e	=>	(4,1,e)

V Tab. 5 je uveden příklad z [13] s úpravou, kde jsou vypsány jednotlivé kroky při kódování řetězce „sir_sid_eastman_easily_teases_sea_sick_seals“. Inicializace vyhledávacího okna spočívá ve vytvoření, prázdného bloku s předem danou velikostí,

sledující okno má rovněž danou velikost avšak, ta by měla být menší než u vyhledávacího okna a výstup označuje zmíněné tokeny. V prvním až čtvrtém kroku se doplňuje jeden znak, v pátém kroku se již doplňují znaky tři. V kroku 5. Je výstupem (4,2,d), což znamená, že ve vyhledávacím okně 4 prvek od konce je počátkem řetězce a délka řetězce je 2 => si. Posledním znakem tokenu v pátém kroku je „d“, což značí poslední přidáný znak aktuálního kroku => si„d“. Stejným způsobem pracuje algoritmus nadále.

2.1.6 Metoda založená na ASCII kódování

Klíč ke kódování a získání vysokého kompresního poměru je ve vhodném způsobu kódování dat a jejich výsledném spojení do výstupního textového souboru. Celé kódování je rozděleno do 6 kroků, které jsou [7]:

1. Výběr okna 8 vzorků

Prvním krokem je separace časové osy ze samotného signálu, jelikož je známá vzorkovací frekvence, není nutno čas ukládat. Následující kroky jsou aplikovány na 8 po sobě jdoucích vzorků.

2. Delta kódování

Delta kódování je prováděno tím způsobem, že první vzorek z okna se uchová a poté se zbývajících 7 vzorků od prvního hodnoty odečítá. Důvodem tohoto kroku je snaha o dosažení co nejmenších čísel, díky tomu je možno dosáhnout vysokého kompresního poměru.

3. Generování absolutních hodnot celých čísel

V tomto kroku je ověřeno znaménko všech čísel z předchozího kroku a jejich hodnoty jsou uloženy jako binární čísla (kladné = 0, záporné = 1). Následně jsou všechny hodnoty vynásobeny 1 000 a je na ně aplikována absolutní hodnota.

4. Normalizace

V kroku delta kódování, byl první vzorek uchován bez jakékoliv změny, což v případě, že okno začíná vysokou hodnotou (vyšší než $|0.255|$), po vynásobení přesáhne hodnotu 255, což nemůže být převedeno na text. K předcházení tohoto problému je hodnota vydělena 200 a zbytek po dělení je ponechán na původní pozici, přičemž celé číslo je uloženo do speciální proměnné.

5. Kódování dat

Pro redukci velikosti dat je na 8 vzorků v okně použita následující logika:

```
if ([a(i)*100]+a(i+1)<255)
    e(j)=[a(i)*100]+a(i+1)
    j=j+1
else if (a(i)+[a(i+1)*100]<255)
    e(j)=a(i)+[a(i+1)*100]
    j=j+1
end
```

Ve výše uvedeném kódu je proměnná a hodnotou i -tého vzorku z aktuálního okna a e je hodnota j -tého výstupního vzorku, přičemž hodnota i je zvyšována v každém kroku o 2.

Vzorky jsou kódovány do tří kategorií dle uvedeného kódu.

- Pokud dvě po sobě jdoucí hodnoty splňují první podmínku, sudé pozice speciální proměnné uložena logická 1
- Pokud dvě po sobě jdoucí hodnoty splňují podmínku druhou, liché pozice speciální proměnné uložena logická 1
- Pokud dvě po sobě jdoucí hodnoty nesplňují ani jednu podmínku, ponechají se vzorky nezměněné a uloží se do nové proměnné

6. Standardizace dat do textového souboru

V tomto kroku jsou všechny proměnné uloženy v pevně nastaveném pořadí do textového souboru. Právě díky tomu, že hodnoty nepřesahují 255, je možné je uložit jako ASCII znaky. Pokud program nedošel na konec souboru s EKG, pokračuje bodem 1 [7].

Tab. 6 – Kompresní poměry a časy algoritmu ASCII [7]

Délka signálu [s]	15	15	15
Kompresní poměr [-]	1/18,27	1/18,86	1/17,87
Kompresní čas [ms]	329,34	417,67	389,83

Jak je vidět v Tab. 6, kompresní poměry uvedené metody jsou na velmi dobrých hodnotách. Nutno však podotknout, že v případě delších signálů uvedených v [7], začala vznikat jistá chyba, avšak jedná se o desetiny promile v případě parametru PRDN.

2.1.7 Metoda založená na K-means clusteringu

Tato metoda je založená na podobnosti jednotlivých úseků v EKG křivce. Právě díky podobnosti lze využít K-means clustering ve prospěch lepšího kompresního poměru. Celá metoda probíhá ve 4 krocích, které lze rozdělit na další kroky [8].

V prvé řadě je nutno detekovat **vlny R**, aby bylo možno provádět shlukování. Po detekci následuje právě zmiňovaný K-means clustering, který probíhá v 6 krocích:

1. Vlna R se označí jako střed a průměrné hodnoty RR intervalů vynásobených hodnotou 0,8 se označí jako šířka QRS komplexu. Následně se komplexy QRS oddělí do samostatných polí.
2. Rozdělení všech komplexů do 16 clusterů, přičemž prostřední komplex QRS je určen jako iniciační průměrná hodnota clusteru
3. Spočítá se Euklidovská vzdálenost mezi komplexy a průměrnými hodnotami clusterů a následně se zařadí do nejbližšího.
4. Aktualizuje se průměrná hodnota všech 16 clusterů.
5. Iterace pokračuje krokem 3 a 4, dokud se průměrná hodnota clusterů nemění, nebo počet iterací nedosáhne čísla 3 000.
6. Po rozdělení všech komplexů QRS mezi jednotlivé clustery, kodér pokračuje výpočtem rozdílů mezi všemi a průměrnými komplexy QRS pro jednotlivé clustery.

Kodér pokračuje třetí částí, ve které provádí kódování střední hodnoty jednotlivých komplexů pomocí prvního řádu diferenciálního kódování. Díky tomu lze střední hodnoty z původních 12 bitů snížit až na 8 bitů.

V závěrečném kroku kodér aplikuje Huffmanovo kódování na rozdíly mezi komplexy QRS a jejich korespondující průměrné vlny v jednotlivých clusterech. Dále se uloží všech 16 clusterů bez jakýchkoliv změn. Jako poslední se aplikuje opět Huffmanovo kódování na uložené hodnoty vln R, a částí se středními hodnotami. Algoritmus byl testován na databázi MIT-BIH ARRHYTHMIA, podle čehož je i číslování v Tab. 7 [8].

Tab. 7 – Výsledky kódování, založeném na K-means clusteringu [8]

Signál č.	100	101	102	103	104	105	106
CR	1/3,552	1/3,594	1/3,037	1/3,287	1/3,071	1/3,093	1/2,834

2.2 Ztrátová komprese

Ztrátová komprese je zřejmě nejvyžívanější kompresí v běžném životě. Nejznámější je zřejmě komprese JPEG, avšak tento typ se nehodí pro technické výkresy, či kompresi EKG signálů. Z uvedených důvodů jsou zde uvedeny jednoduché avšak poměrně účinné algoritmy pro kompresi právě časově závislých signálů, jako je například EKG. Tyto metody lze rozdělit na dvě kategorie: Přímá komprese v časové oblasti a komprese založená na vlnkové transformaci.

2.2.1 Algoritmus Turning point

Turning point (bod zvratu), je velmi jednoduchý algoritmus založený na bázi porovnání sousedních vzorků a jejich vynechání.

Algoritmus prochází signál vzorek po vzorku s oknem délky 3 a kontroluje následující podmínky [9]:

$$\begin{aligned} (x_1 - x_0) * (x_2 - x_1) < 0 \\ \text{nebo} \\ (x_1 - x_0) * (x_2 - x_1) > 0, \end{aligned} \tag{4}$$

kde x_0 je první, x_1 druhý a x_2 třetí vzorek v okně. V případě, že je splněna první podmínka, x_1 se uchová a v případě, že je splněna podmínka druhá, uchová se x_2 .

Z podmínky (4) vyplývá, že základní kompresní poměr algoritmu Turning point je roven 1/2, přičemž je možné poměr zvýšit opakováním téhož algoritmu. [9]

2.2.2 Algoritmus AZTEC

Další ztrátová metoda komprese je algoritmus AZTEC (Amplitude Zone Time Epoch Coding). AZTEC převádí EKG signál na rovné úseky (tzv. plató) a šikmé úseky. V případě, že se v signálu vyskytují vedle sebe dva plató úseky s různou výškou signálu, vznikají ve výsledku přerušení. I přes to, že AZTEC má poměrně vysoký kompresní poměr, signály, komprimované tímto algoritmem nemohou být zejména z důvodu těchto přerušení využívány pro kardiologické vyšetření. Algoritmus probíhá ve dvou fázích [9]:

Fáze 1 Horizontální:

1. První vzorek se nastaví jako X_{\max} a X_{\min} , které následně reprezentují nejvyšší a nejnižší hodnoty v aktuálním úseku.

2. Kontrolují se následující podmínky:

$$X_1 > X_{max} \text{ pak } X_{max} = X_1 \quad (5)$$

$$X_1 < X_{min} \text{ pak } X_{min} = X_1$$

a rovněž pro každý další vzorek, dokud není rozdíl mezi X_{max} a X_{min} větší než přednastavený práh, nebo délka úseku není delší než 50 vzorků.

3. Ukládá se hodnota délky $L = S-1$, kde S je počet vzorků v úseku, a průměrná hodnota úseku plató $(X_{max}+X_{min})/2$.
4. Algoritmus pokračuje opakováním bodů 1-3.

Fáze 2 šikmá:

1. Pokud počet vzorků v úseku plató je menší nebo roven 3, tak se parametry v kroku první fázi neukládají a místo nich se začne ukládat šikmý úsek.
2. Směr a velikost šikmého úseku je definován výpočtem

$$ve = (X_2 - X_1) * (X_1 - X_0) \quad (6)$$

3. Šikmý úsek je ukončen, pokud počet vzorků překročí 3, nebo se změní směr úseku. [9]

2.2.3 Kódování založené na lineární predikci vlnek

Tato technika kódování využívá toho, že v cyklech na EKG existují korelace mezi jednotlivými cykly srdce a zároveň korelace v rámci jednoho cyklu. Metoda vyžaduje detekci pouze vlny R. V první řadě je provedena transformace s minimální ztrátou, která všechny úderý převede na konstantní délky. Následně je provedena normalizace výchylky. Na tento signál je aplikována diskretní vlnková transformace a následně je využita lineární predikce vybraných vlnkových koeficientů. Díky těmto úpravám stačí přenášet pouze sekvenci s chybou předpovědi [10].

Normalizace periody probíhá následovně: Cyklus je definován jako signál od jedné vlny R do druhé a každý cyklus je pomocí převzorkování prodloužen na stejně dlouhé úseky, převzorkováním na stejně dlouhé úseky je omezen vliv variability srdečního rytmu, což tato metoda vyžaduje. Na výstup je odeslána průměrná doba periody, přičemž během kódování je odesílána hodnota rozdílu od průměrné doby periody. Normalizace výchylky je realizována pomocí vydělení každého vzorku jednotlivých úderů jejich nejvyšší hodnotou.

Následuje vlnková transformace každého úseku zvaného PAN (period and amplitude normalized). Vlnková transformace je popsána v kapitole 3. Díky

nerovnoměrnému rozložení energie v jednotlivých úsecích PAN, je možné přibližně 80 % koeficientů vlnkové transformace vypustit bez ztráty signifikantních rytmů, nebo morfologických změn.

Díky tomu, že mezi korespondujícími vlnkovými koeficienty existuje korelace, stačí přenášet pouze rozdíly od normalizovaného cyklu, vytvořeného pomocí několika posledních cyklů, což je principem lineární predikce s využitím kvadratického průměrného kritéria.

Dekomprese signálu probíhá aplikací inverzní lineární predikce, následované přeuspořádáním koeficientů jednotlivých PAN úseků pro vlnkovou transformaci. Po inverzní vlnkové transformaci probíhá převzorkování na původní délky signálů a úprava jejich amplitudy. Následně se všechny úseky spojí dohromady, z čehož vznikne původní signál.

Kompresní poměry společně s hodnotou PRD (vysvětlena v kapitole 4.1) jsou uvedeny v Tab. 8 [10].

Tab. 8 – Výsledky kódování, založeném na lineární predikci [10]

CR [-]	1/16,76	1/19,1	1/15,4	1/22,3	1/13,4
PRD [%]	10,07	11,27	11,73	13,34	9,89

2.2.4 Metoda založená na vlnkové paketové transformaci

Metoda samotná se skládá ze tří částí, kde první část je transformace pomocí vlnkové paketové transformace, druhá část je kvantizace koeficientů a ve třetí části vytváří entropický kódér výsledný tok bitů [11].

Vlnková paketová transformace je jinou verzí vlnkové transformace, popsané v kapitole 3.2. Hlavním rozdílem vlnkové transformace a vlnkové paketové transformace je ten, že při rozkladu signálu je rozkládána aproximace i detail, tudíž každá úroveň rozkladu má stejný počet vzorků jako originální signál [12]. Díky využití vlnkové paketové transformace je energie celého signálu uložena v relativně málo koeficientech, tudíž je možné tyto koeficienty jednoduchým prahováním kvantizovat [11].

Po samotné transformaci je nutno odeslat na výstup informace, vedoucí k sestavení stromové struktury využité při dekompozici. Jelikož na vstupu i výstupu jsou všechny možné spektrální dekompozice k dispozici, stačí na výstup odesílat pouze celá čísla, označující adresu odpovídajícího stromu. Počet bitů potřebných k identifikaci stromu je

definován počtem bází, které závisí na počtu stupňů rozkladu [11].

Tab. 9 – Výsledky kódování metody založené na vlnkové paketové transformaci převzato z [11]

CR [-]	1/9,62	1/10,65	1/12,38	1/12,98	1/13,76	1/14,78	1/16,05
PRDN [%]	4,78	5,19	6,19	6,69	7,46	8,6	10,34

3 SPIHT

SPIHT (Set Partitioning in Hierarchical Trees) byl původně navrhnut pro kompresi obrazových dat, avšak lze jej použít i pro 1D data, jako například EKG. Základem pro tuto metodu je L-stupňová vlnková transformace, která rozloží signál na jednotlivá pásma a zapíše je do stromové struktury. Tato struktura je následně kódována pomocí algoritmu SPIHT, který v jednotlivých iteracích kontroluje větve stromu a následně odesílá bity, které informují o důležitosti, či nedůležitosti větví a jednotlivých uzlů. [14]

Přestože se jedná již o více než 20 let starý algoritmus (1996), jde o jeden z nevyužívanějších a nejefektivnějších algoritmů. Jeho využití je dáno zejména možností nastavitelného kompresního poměru, což je umožněno přerušением datového toku v libovolném čase (zpracuje se pouze přijatá informace) [16].

3.1 Spojitá vlnková transformace

Spojitá vlnková transformace (continuous wavelet transform – dále CWT) je definována jako korelace mezi signálem $x(t)$ s rodinou vlnek $\psi_{a,b}$, kde pro každé a a b [15]:

$$y(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \cdot \psi * \left(\frac{t - b}{a} \right) dt, \quad (7)$$

kde $\psi_{a,b}(t)$ je mateřská vlnka, od níž se odvozují korelované vlnky. Mateřská vlnka musí mít nulovou střední hodnotu. Hvězdička označuje komplexně sdruženou funkci. Měřítkový parametr a nastavuje oscilační frekvenci a délku vlnky. Translační parametr b rozhoduje o posunu vlnky. Při nastavení velkého měřítka a jsou výsledkem nižší frekvence signálu s nízkým časovým rozlišením avšak vysokým frekvenčním rozlišením a naopak pro malé a jsou výsledkem vysoké frekvence s vysokým časovým rozlišením, ale nízkým frekvenčním rozlišením [16].

3.2 Vlnková transformace s diskretním časem

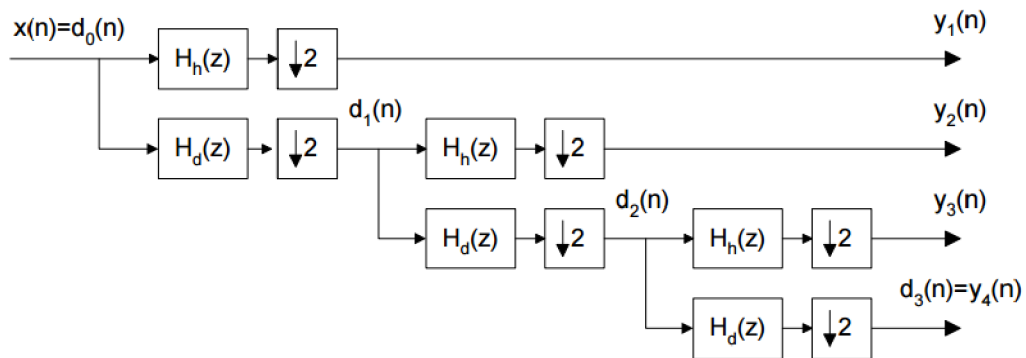
Tento typ vlnkové transformace se realizuje pomocí banky filtrů, které jsou předem známy. Analogicky ke vztahu (7) probíhá výpočet DTWT (discrete time wavelet

transform) diskretní konvolucí, [15]

$$y_m(n) = \sum_{i=-\infty}^{\infty} h_m(i)x(2^m n - i), \quad (8)$$

Kde m označuje stupeň rozkladu, n aktuální vzorek signálu y a h_m označuje pro dyadickou DTWT impulsní charakteristiku filtru.

Reálně se využívá tzv. rychlá dyadická DTWT, která spočívá ve využití zrcadlových horních H_h a dolních H_d propustí, zpracovaných do stromové struktury:



Obr. 6 – Rychlá dyadická DTWT (převzato z [16])

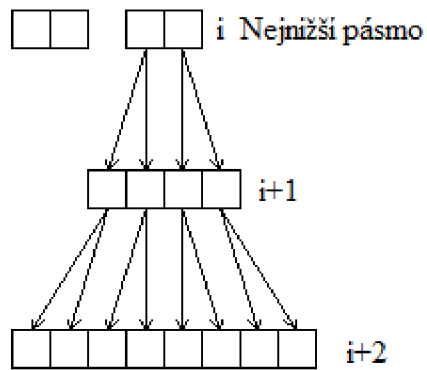
Kde $\downarrow 2$ značí podvzorkování signálu.

Výsledkem dyadické transformace jsou koeficienty $y_m(n)$ $m=1, 2, \dots, M+1$, které jsou nerovnoměrně rozloženy v časově-frekvenční rovině. M pro případ Obr. 6 je rovno 3, což značí třístupňový rozklad.

Při inverzní DTWT se postupuje obdobně jako při DTWT avšak s tím rozdílem, že vstupem jsou signály $y_m(n)$ a před vstupem do inverzních filtrů $H_h(z)$ a $H_d(z)$ jsou signály expandovány (vložením nuly mezi každé dva sousední vzorky).

3.3 SPIHT algoritmus

Po získání jednotlivých vlnkových koeficientů je signál rozdělen podle počtu stupňů rozkladu na strom, se kterým algoritmus následně pracuje. Jedná se o strukturu, ve které každý bod ve vrstvě i koresponduje s dvěma body ve vrstvě $i+1$, jako je znázorněno na



Obr. 7, kde je třístupňový rozklad.

Obr. 7 – Stromová struktura vstupních dat

Pro pochopení algoritmu je nutné vysvětlit pojmy, které budou v popisu užívány:

LSP (list of significant points) – Seznam bodů, které svou hodnotou přesahují práh v dané iteraci.

LIP (list of insignificant points) – Seznam bodů, které svou hodnotou nepřesahují práh v dané iteraci.

LIS (list of insignificant sets) – Seznam nedůležitých větví obsahuje čísla kořenových prvků, které v aktuální iteraci neobsahují žádné důležité body.

$O(c_i)$ – Seznam přímých potomků větve c_i .

$D(c_i)$ – Seznam všech potomků větve c_i .

$L(c_i)$ – Seznam všech potomků větve c_i s výjimkou přímých potomků.

Vstup typu A – Reprezentuje strom, u něhož probíhá kontrola $D(c_i)$.

Vstup typu B – Reprezentuje stromu, u něhož probíhá kontrola $L(c_i)$.

Algoritmus kódování probíhá ve čtyřech krocích, přičemž poslední tři se opakují v jednotlivých iteracích.

1. Inicializace

Při tomto kroku se nastavují počáteční hodnoty jednotlivých proměnných,

a sice: LSP = prázdná, LIP = čísla všech prvků umístěných v nejnižším pásmu, LIS = čísla prvků v nejnižším pásmu, které mají potomky.

2. Řadící průchod

Během tohoto kroku probíhá kontrola všech prvků v seznamu LIP a následně kontrola větví umístěných v LIS.

3. Upřesňující průchod

Redukují se všechny hodnoty v LSP a kontrolují se všechny hodnoty v LSP z předchozí iterace.

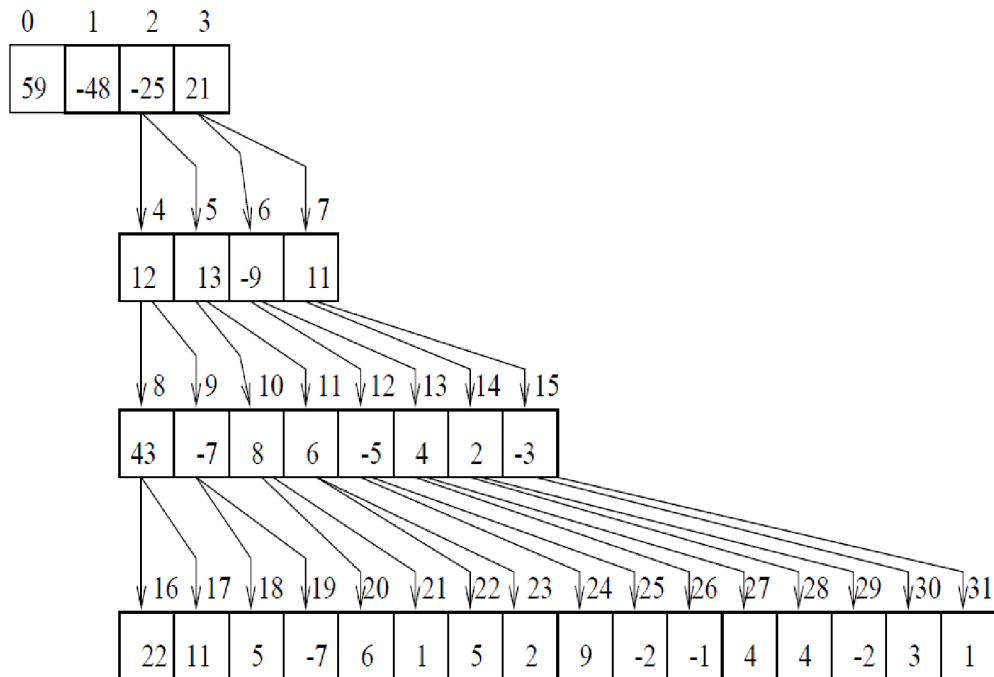
4. Redukce prahu

Sníží se práh a algoritmus se vrací k bodu 2, dokud není přerušen, nebo je práh na hodnotě 0.

Dekódování má obdobný průběh jako kódování, s tím rozdílem, že podle toku bitů jsou postupně doplňovány jednotlivé proměnné (LSP, LIP a LIS). Z doplněných proměnných se rekonstruuje původní strom, z něhož se zpětnou vlnkovou transformací vytvoří původní signál. [15].

3.3.1 SPIHT kódování příklad

Pro pochopení průběhu kódovací části algoritmu je vhodný příklad z [15].



Obr. 8 – Rozložený příkladový signál (převzato z [15])

1. Výpočet prahu 2^n , přičemž n je

$$n = \lceil \log_2 \max_{(i)} |c_i| \rceil, \quad (9)$$

tudíž práh je dle hodnot z Obr. 8 roven 32.

2. Vložení rozměrů jednotlivých vrstev stromu, velikosti výsledného signálu, počtu stupňů rozkladu a hodnoty vypočítaného n na začátek výstupu (nutné údaje pro rekonstrukci).
3. Nastavení hodnot LSP=prázdný, LIP=[0,1,2,3], LIS=[2A,3A].
4. Řadící průchod LIP
Kontrolují se všechny prvky v LIP, a ty které přesahují práh, se přemístí do LSP. Při přesunu se na výstup pošle 1, následovaná znaménkem daného prvku. V opačném případě se pošle 0. Pro aktuální hodnoty je výstupem 1+,1-,0,0 (c_0 a c_1 se přesunou do LSP) c_2 a c_3 zůstávají v LIP.
5. Řadící průchod LIS

Tato část má možné dva průběhy, proto se při vstupu do řadícího průchodu LIS kontrolují typy prvků (A nebo B).

Pro vstup typu A: Probíhá porovnání všech potomků dané větve s prahem. Pokud některý z potomků překračuje práh, je uzel označen jako signifikantní a na výstup se pošle 1. Následuje kontrola přímých potomků uzlu ($O(c_i)$) stejným způsobem jako v bodě 3 (tzn. pokud je prvek větší než práh, přesune se do LSP a na výstup se pošle 1 následovaná znaménkem daného prvku. V opačném případě se prvek přesouvá do LIP). Dále se kontroluje, zda má uzel nepřímé potomky ($L(c_i)$). V případě, že má nepřímé potomky, se uzel přesune na konec seznamu LIS jako typ B. Pokud jsou všichni potomci nesignifikantní, pošle se na výstup 0.

Pro aktuální hodnoty platí, že $D(c_2)$ je signifikantní, přímí potomci c_4 a c_5 jsou nesignifikantní, a uzel má nepřímé potomky, to znamená výstup:100

Pro vstup typu B: Probíhá kontrola nepřímých potomků, a pokud je alespoň jeden z nich větší než práh, je uzel signifikantní, na výstup se pošle 1 a čísla přímých potomků jsou zařazena na konec LIS jako typ A. Mateřský uzel je z LIS odstraněn. Pokud je uzel nesignifikantní, na výstup se pouze pošle 0. Pro aktuální hodnoty je prvním prvkem v LIS typu B uzel 2, jenž je signifikantní ($c_8=43>32$), to znamená, že na konec LIS se přesunou přímí potomci 4A a 5A a 2B se odstraní z LIS. Na výstupu se objeví 1.

6. Upřesňující průchod

V tomto průchodu se kontrolují všechny předchozí hodnoty LSP (hodnoty v LSP před řadícím průchodem). Pokud hodnota přesahuje práh, odešle se na výstup 1, v opačném případě se odešle 0. Následně se všechny hodnoty v aktuálním LSP sníží o hodnotu prahu. Pro aktuální hodnoty se na výstup neodesílá žádná hodnota, jelikož v LSP nebyla před řadícím průchodem žádná hodnota. Hodnoty c_0 , c_1 a c_8 se sníží o 32.

7. Redukce prahu

Jak bylo výše popsáno, práh se určuje jako 2^n , kde n je definováno vzorcem (9). V tomto kroku se n sníží o 1 a algoritmus se vrací na bod 3 [15].

3.3.2 SPIHT dekódování příklad

Pro vysvětlení budou použity výstupy z předchozí podkapitoly:

Tab. 10 – Výstup z algoritmu SPIHT

řs	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	2	2	4	8	16	32													
2	4	5																	
3	1	+	1	-	0	0	1	0	0	0	1	1	1	+	0	0	0	r	
4	1	-	1	+	0	0	0	0	0	1	1	1	+	0	0	1	1	0	r

Tab. 10 udává hodnoty výstupu z algoritmu, kde sloupce označují číslo odeslaného bitu a řádky jsou uvedeny pro přehlednost. Jedná se o výstup po dvou průchodech výše zmíněných úkonů. Jak již bylo zmíněno v úvodu kapitoly, dekódování probíhá velmi obdobně jako kódování:

1. Inicializace proměnných

Díky informaci o počtu stupňů rozkladu na druhém řádku Tab. 10 je vygenerován strom o stejné struktuře jako Obr. 8, avšak s tím rozdílem, že všechny hodnoty jsou nulové. Následně se inicializují proměnné LIP, LSP a LIS stejně jako při kódování, tzn. $LIP = [0,1,2,3]$, $LIS = [2A,3A]$ a $LSP =$ prázdná.

2. Řadící průchod LIP

V této části program prochází proměnnou LIP a čte stejný počet vstupních hodnot jako je délka vektoru LIP (1+, 1-, 0, 0) a následně ve stromu na pozicích LIP násobí hodnotu prahu vstupní hodnotou. V aktuálním případě to znamená, že na pozice 0 a 1 se doplní hodnota 32 a -32.

3. Řadící průchod LIS

V tomto kroku program prochází proměnnou LIS a podle vstupu se tato proměnná upravuje. Opět se rozhoduje podle typu (A nebo B). V případě typu A vstupní hodnota dává informaci o signifikanci větve. V případě, že větev je signifikantní, probíhá pro dva přímé potomky mateřského uzlu stejná kontrola jako v bodě 2 (na pozicích potomků se umístí hodnota dvou vstupních bitů vynásobených hodnotou prahu a provedou se příslušné přesuny mezi proměnnými). Následuje kontrola, zda mohou existovat nepřímí potomci mateřského uzlu. V případě že nepřímí potomci existují, přesune se hodnota na konec LIS jako typ B, v opačném případě se hodnota z LIS vymaže. V aktuálním případě se čte vstup pro typ A (2A), který je roven 1, tudíž se provede kontrola přímých potomků uzlu 2 (4 a 5). Vstupy na pozicích 3\8 a 3\9 v Tab. 10 informují o nesignifikanci bodů 4 a 5, takže se tyto body přesunou do

proměnné LIP. Ze stromu se následně určí, že uzel 2 má nepřímé potomky, takže se přesune na konec LIS jako typ B. Pro hodnotu v LIS typu B, vstup rozhoduje o signifikanci nepřímých potomků. Pro vstup 1 se přímí potomci zařadí na konec LIS jako typ A a hodnota mateřského uzlu se z LIS vymaže. Pro opačný případ se neprovádí žádná změna. Situace, ve které se zpracovává uzel typu B, nastává na vstupní hodnotě 3\11 Tab. 10. Hodnota 1 informuje o signifikanci některého z nepřímých potomků, tudíž se hodnoty 4 a 5 stávají mateřskými uzly typu A, zařazují se na konec LIS a hodnota 2B se z LIS vymaže.

4. Upřesňující průchod

V posledním kroku vstupní hodnota 1 označuje navýšení na pozicích LSP z předchozího průchodu o hodnotu aktuálního prahu. V případě 0 se nic nemění. V případě prvního průchodu se na výstupu neobjevuje žádná hodnota, značící změny upřesňujícího průchodu, z toho důvodu je v Tab. 10 uveden výstup po druhém průchodu, konkrétně na pozicích 4\16,17 a 18. Znamená to, že se na pozicích 0 a 1 hodnota navýší o hodnotu prahu v druhém průchodu.

5. Redukce prahu

Ve výstupu je redukce značena znakem „r“ a její průběh se nijak neliší od kódovací části, tzn. n se sníží o 1 a probíhá návrat ke kroku 2.

4 HODNOCENÍ KVALITY

Pro zhodnocení kvality rekonstruovaného signálu lze využít různá kritéria. Na základě článků [10], [14] a [15], zde budou popsána kritéria PRD, PRDN a NMAE.

4.1 PRD – Percent root mean square difference

Ve většině algoritmů pro kompresi signálů EKG se využívá parametru PRD, což se v češtině označuje jako procentuální chyba. Toto kritérium je však závislé na střední hodnotě signálu. PRD je definováno jako

$$PRD = \sqrt{\frac{\sum_{i=0}^{N-1} [x_{ori}(i) - x_{rec}(i)]^2}{\sum_{i=0}^{N-1} x_{ori}(i)^2}} * 100 [\%], \quad (10)$$

kde N je počet vzorků v signálu, $x_{ori}(i)$ je originální signál a $x_{rec}(i)$ je rekonstruovaný signál [17].

4.2 PRDN – Normalized percent root mean square difference

PRDN, čili normalizovaná verze procentuální chyby, eliminuje nevýhodu kritéria PRD. Nevýhoda závislosti na střední hodnotě signálu je odstraněna ve jmenovateli právě odečtením této hodnoty. PRDN je definován jako

$$PRDN = \sqrt{\frac{\sum_{i=1}^n [x_{ori}(i) - x_{rec}(i)]^2}{\sum_{i=1}^n [x_{ori} - \bar{x}_{ori}]^2}} \cdot 100 [\%], \quad (11)$$

kde n je délka signálu, x_{ori} je originální signál, \bar{x}_{ori} je průměrná hodnota originálního signálu a x_{rec} je rekonstruovaný signál [16][15].

4.3 NMAE – Normalized maximum amplitude error

NMAE, lze přeložit jako normalizovanou hodnotu maximální chyby amplitudy. Toto kritérium bylo využito v článku [10] společně s hodnotou NRMSE (normalized root mean square error), která odpovídá hodnotě PRD. NMAE lze využít u metod, které pracují v jednotlivých cyklech, jako například metoda z kapitoly 2.2.3, jelikož určuje maximální chybu rekonstrukce normalizovanou na rozsah signálu v jednotlivých cyklech. Výpočet NMAE i -tého cyklu je dán vztahem

$$NMAE_i = \frac{\max|x_{oi} - x_{ri}|}{\max(x_{oi}) - \min(x_{oi})} \quad (12)$$

kde x_{oi} je hodnota i -tého cyklu originálního signálu a x_{ri} je hodnota i -tého cyklu rekonstruovaného signálu. Výsledkem je pak průměrná hodnota NMAE [10].

5 VÝSLEDKY

Pro hodnocení výsledků bylo využito prvních 50 signálů z databáze MIT-BIH ECG Compression Test Database (cdb), která vznikla výběrem holterovských záznamů pro testování kompresních algoritmů EKG. Tato databáze byla vytvořena tak, aby zahrnovala širokou škálu síňových komplexů, junkční rytmy, komorové arytmie,

poruchy vedení a šum, za účelem ověření všech možných vstupních signálů. Všechny signály byly vzorkovány 250 Hz s 12 bitovým rozlišením po dobu 20 s [17],[18].

Byla porovnávána hodnota PRDN v závislosti na použité vlnce a následně na stupních rozkladu ve vlnkové transformaci.

5.1 Rozdíly mezi použitými vlnkami

Vyzkoušené vlnky byly vybrány na základě článku [16] a jedná se o:

Ortogonalní vlnky: haar, db2 a db5

Biortogonalní vlnky: bior2.2, bior 2.4, bior4.4 a bior 6.8

Všechny vlnky byly porovnány při kompresním poměru CR (compression ratio) 1/5, definovaném jako

$$CR = \frac{\text{velikost výstupního signálu}}{\text{velikost vstupního signálu}} \quad (13)$$

na všech signálech [13]. Ve výpočtu CR je literatura nejednotná, z toho důvodu je zde uveden druhý parametr, který se nazývá kompresní faktor (CF – compression factor) a je dle [13] definován jako

$$CF = \frac{\text{velikost vstupního signálu}}{\text{velikost výstupního signálu}} \quad (14)$$

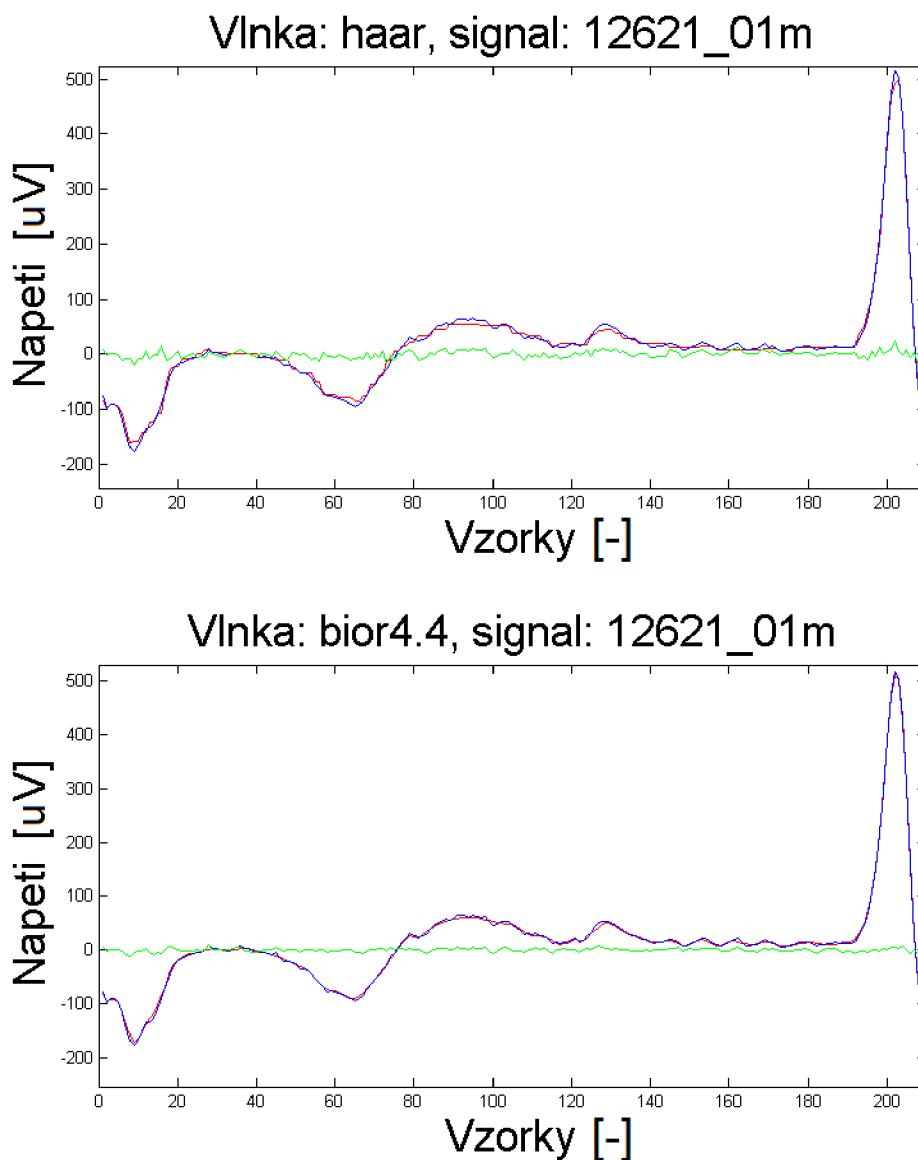
Signály uvedené v Tab. 11, 11442_03m a 12621_01m zmiňované databáze, byly vybrány jako ukázkové na základě předchozího testování všech signálů. Zmíněné signály jsou ty, které měly nejhorší a nejlepší výsledky.

Tab. 11 – Hodnoty PRDN [%] pro jednotlivé vlnky při CR = 1/5

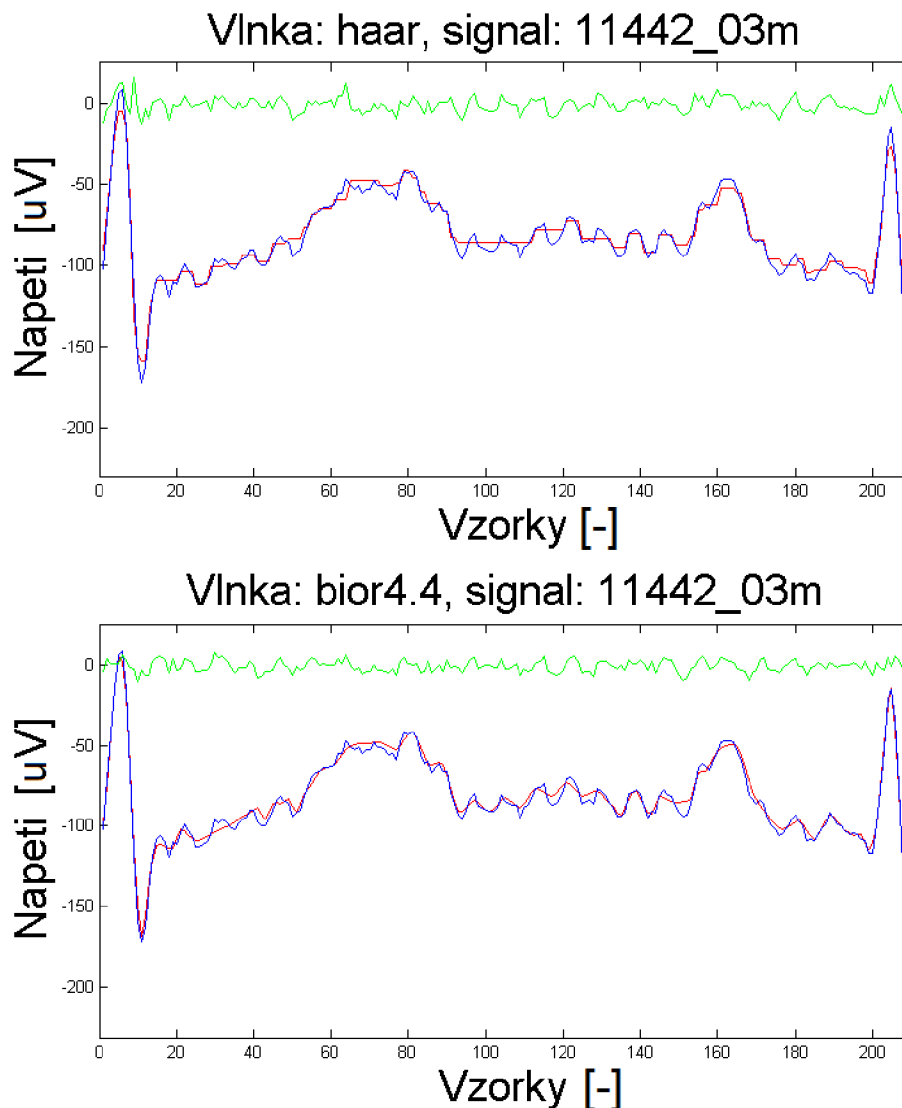
Název vlnky	haar	db2	db5	bior2.2	bior2.4	bior4.4	bior6.8
11442_03m	16,46	12,48	11,72	12,88	12,86	12,09	13,72
12621_01m	6,22	5,61	3,57	5,13	5,16	3,74	4,26
Průměr všech signálů	7,04	5,69	5,35	5,59	6,00	5,31	5,79

Z výsledků, zobrazených v Tab. 11, je patrné, že velmi záleží na signálu, který je zpracováván, což je dáno tím, jak tvar signálu kopíruje jednotlivé vlnky. Nejlepších výsledků dosahuje vlnka bior4.4, v některých člancích označovaná jako Biorthogonal

9/7 [15]. Označení všech vlnek bylo převzato z prostředí MATLAB. Pro optické porovnání výsledků jsou na Obr. 9 vykresleny výsledky výřezu signálů 1142_03m a 12621_01m pro vlnky haar a bior4.4, tedy pro vlnky s nejlepšími a nejhoršími výsledky



Obr. 9 – Výsledky rekonstruovaného signálu 12621_01m pomocí vlnek haar a bior4.4 (červeně – rekonstruovaný signál, modře – originální signál, zeleně – chybový signál)



Obr. 10 - Výsledky rekonstruovaného signálu 11442_03m pomocí vlnek haar a bior4.4 (červeně – rekonstruovaný signál, modře – originální signál, zeleně – chybový signál)

Na první pohled jsou z Obr. 9 a Obr. 10 patrné rozdíly mezi použitými vlnkami i signály, vlnka haar při rekonstrukci vytváří pravoúhlé skoky, což se projevuje větší hodnotou PRDN, naopak díky vlnce bior4.4 výsledný signál věrněji kopíruje průběh originálního signálu. V Tab. 12 jsou výpočetní časy jednoho bloku bufferu (1 024 vzorků) v závislosti na použité vlnce, měřené v prostředí MATLAB na počítači AMD FX8300 4,2GHz (8 jader), 16 GB RAM DDR1333. Jak je vidět časy se od sebe mírně liší, avšak při výpočtu delšího signálu, tyto rozdíly mohou znamenat i minuty.

Tab. 12 – Výpočetní časy při použití jednotlivých vlnek

Název vlnky	haar	db2	db5	bior2.2	bior2.4	bior4.4	bior6.8
Čas kódování a dekódování [s]	0,1801	0,1484	0,1496	0,1707	0,1459	0,1321	0,1474

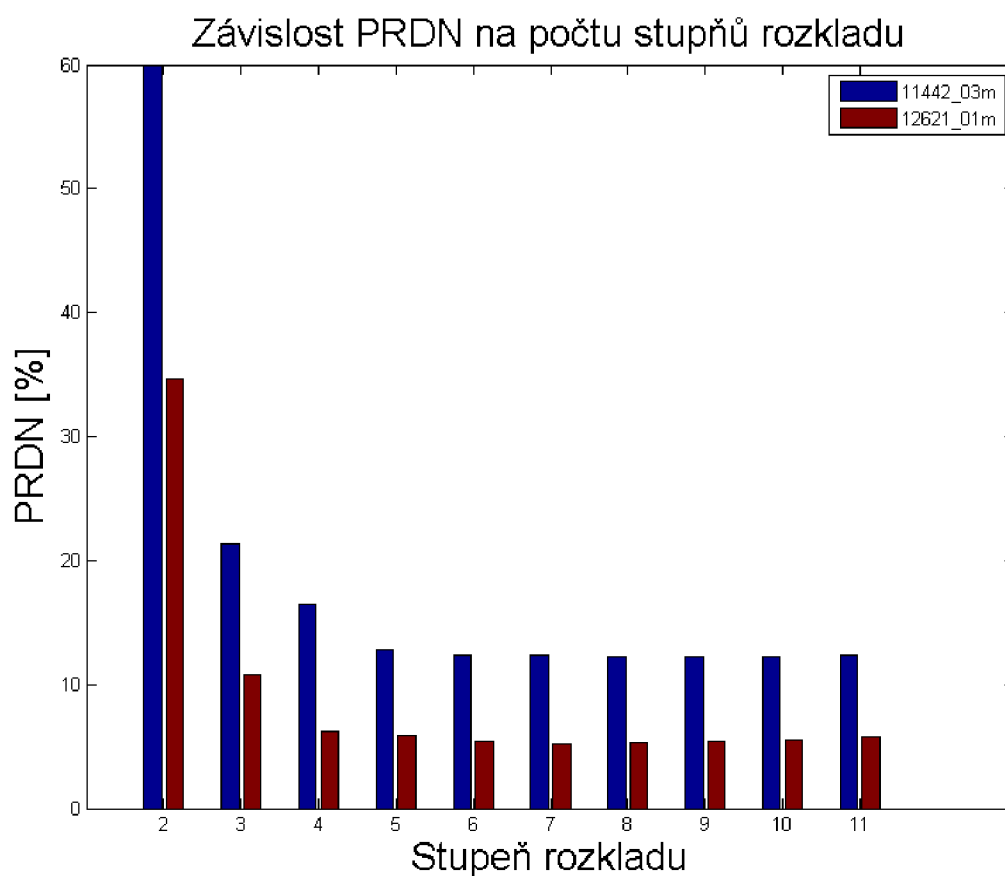
5.2 Rozdíly mezi stupni rozkladu

V předchozí kapitole byla vybrána nejvhodnější vlnka pro stupeň rozkladu 4, zde bude vybrán nejvhodnější počet stupňů rozkladu na stejných signálech, s ohledem na výpočetní náročnost.

Tab. 13 – Časové a kvalitativní rozdíly mezi počty stupňů rozkladu

Počet stupňů rozkladu	2	3	4	5	6	7	8	9	10	11
Čas kódování a dekódování [s]	0,080	0,112	0,152	0,186	0,233	0,259	0,273	0,318	0,385	0,448
11442_03m – PRDN [%]	115,25	21,37	16,46	12,77	12,33	12,3	12,23	12,25	12,26	12,4
12621_01m – PRDN [%]	34,7	10,81	6,22	5,84	5,36	5,23	5,28	5,37	5,49	5,74

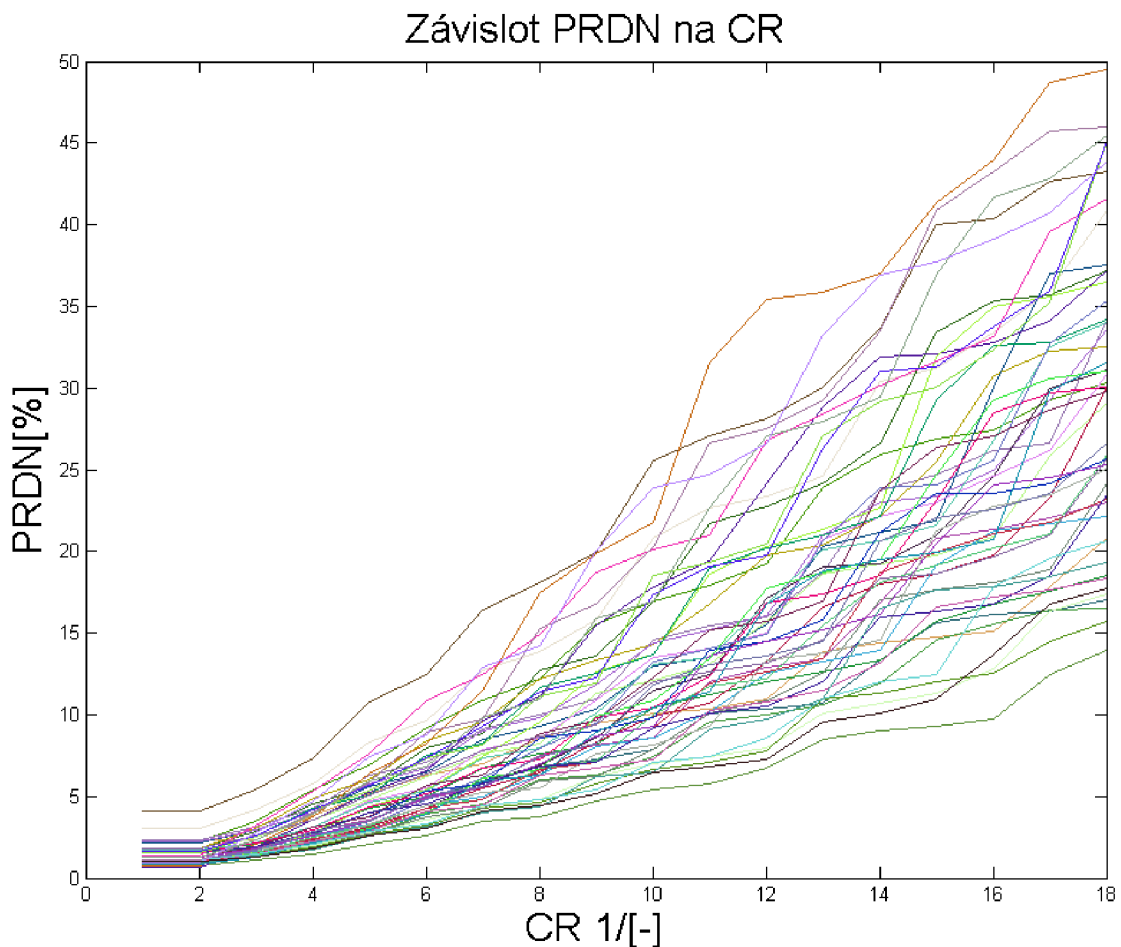
V Tab. 13 jsou vypsány průměrné hodnoty časů komprese a dekomprese pro všechny signály, a PRDN pro signály které dosahovaly nejvyšších a nejnižších hodnot v závislosti na počtu stupňů rozkladu. Na Obr. 11, který byl z důvodu přehlednosti v ose y oříznut na hodnotu 60, je vidět, že u obou signálů hodnota PRDN zpočátku prudce klesá a při stupni rozkladu 6 se téměř nemění, výběr bude tudíž závislý na vhodném poměru času a PRDN. Vzhledem k polynomiálně rostoucímu času se jeví zmiňovaný počet stupňů rozkladu 6 jako nejvhodnější, což se shoduje s [16].



Obr. 11 – Kvalitativní rozdíly signálů 1142_03m a 12621_01m pro různé stupně rozkladu.

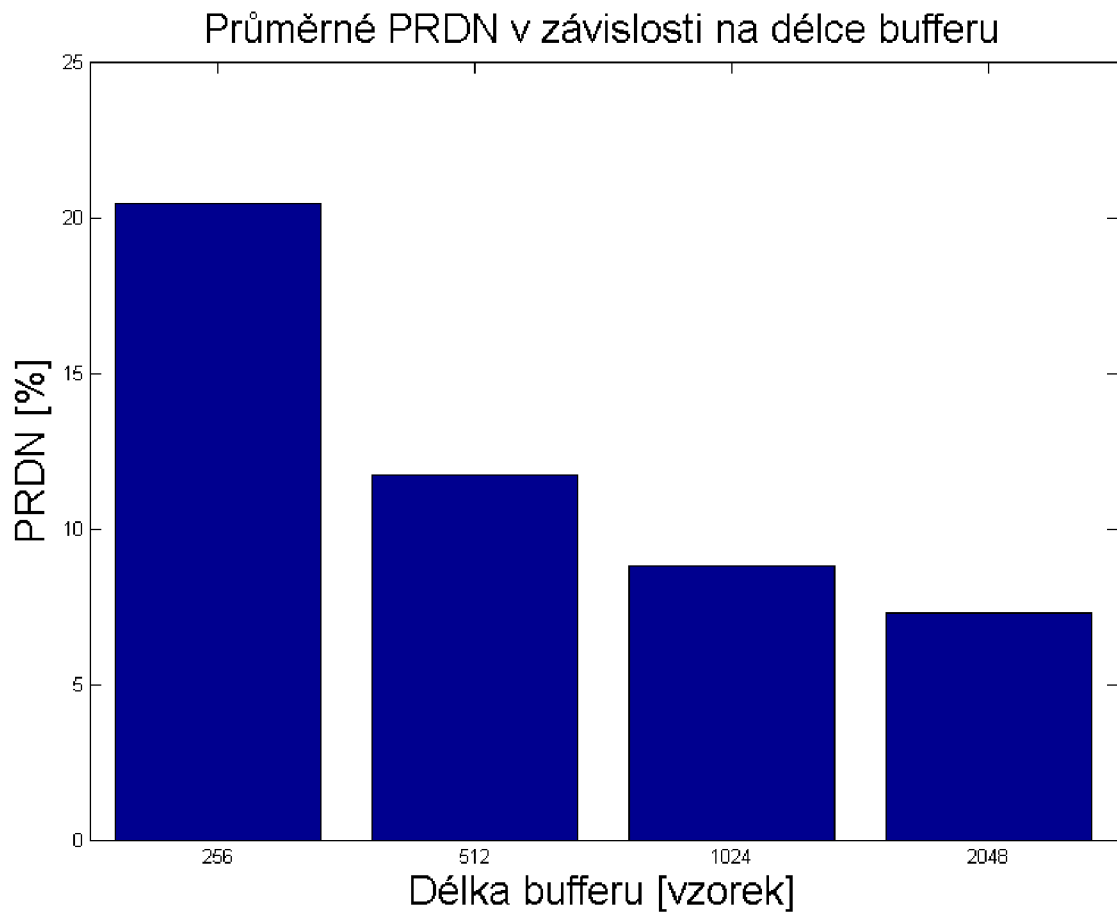
5.3 Zhodnocení

Jak bylo popsáno výše, nejvhodnější je vlnka bior4.4 v kombinaci s 6 stupni rozkladu.



Obr. 12 – Závislost PRDN na kompresním poměru

Na Obr. 12 je zobrazena závislost PRDN na kompresním poměru pro všech 50 signálů databáze, pro výše zmíněné kombinace vlnky a stupně rozkladu. Jak je vidět, všechny signály jsou při nízkém kompresním poměru v téměř stejných hodnotách PRDN, avšak při narůstajícím poměru se signály značně rozebíhají, což je dáno právě povahou signálů, na kterých je algoritmus testován. Jsou zde vyobrazeny signály, které si zachovávají téměř lineární průběh s nízkým nárůstem PRDN, tudíž PRDN pro kompresní poměr 1/18 je stále pod 20 %, avšak objevují se zde signály, u kterých PRDN narůstá polynomiálně až do hodnot téměř 50 %, tento extrém se objevuje u signálu 12247_02m.



Obr. 13 – Sloupcový graf pro výběr délky bufferu

Testována byla i délka bufferu pro zpracování, jak ukazuje Obr. 13, kde vzhledem optimálnímu PRDN s ohledem na rychlost výpočtu, která rovnoměrně s délkou bufferu stoupala. Jako nejvhodnější vzhledem k výpočetní náročnosti byla vybrána délka bufferu 1 024 vzorků, což se shoduje s [15] a [16].

6 EXPERIMENTÁLNÍ ČÁST

Metoda v experimentální části, vznikla na základě komprese obrazu pomocí PNG (Portable network graphics). Komprese PNG je bezztrátová a v kombinaci s proudovým kódováním, které vstupní data ještě zkomprimuje, se může jednat o alternativu k jiným bezztrátovým algoritmům. Značnou nevýhodou dále popisovaného algoritmu je nemožnost nastavení kompresního poměru a oproti algoritmu SPIHT taktéž nemožnost přerušit tok dat, avšak výhodou metody je poměrně malá závislost kompresního poměru na vstupním signálu.

6.1 PNG komprese

PNG komprese je bezztrátová a skládá se ze dvou hlavních kroků. Prvním krokem je takzvaná delta filtrace, která odhaduje hodnotu následujícího pixelu a hodnotu tohoto pixelu nahrazuje pouze rozdílem mezi odhadovanou hodnotou a hodnotou skutečnou. Druhým krokem komprese je metoda deflate, která je založena na kombinaci Huffmanova a slovníkového kódování LZ77 [13].

6.1.1 Delta filtrace

Tato část sama o sobě kompresi neprovádí, provádí přípravu dat pro samotnou kompresi. Příprava probíhá tak, že se data převedou do formátu, ve kterém hodnoty samy sobě více korespondují. Existuje 5 druhů filtrace, které jsou prováděny na obrázku, a následně je vybrána nejvhodnější filtrace pro daný obrázek.

Prvním typem filtrace (typ 0) je zkrátka nefiltrovat.

Druhý typ filtrace (typ 1 - sub) nastavuje hodnotu bajtu $B_{i,j}$ v řádku i a sloupci j na rozdíl $B_{i,j} - B_{i-t,j}$, kde t je interval mezi bajtem a jeho korelovaným předchůdcem. Odečítání probíhá při modulu 256 (unsigned – bez znaménka).

Třetí typ filtrace (typ 2 - up) nastavuje bajt $B_{i,j}$ na rozdíl $B_{i,j} - B_{i,j-1}$. Odečítání je prováděno rovněž při modulu 256. Pokud j je v prvním řádku, odečítání se neprovádí.

Čtvrtý typ filtrace (typ 3 - average) nastavuje bajt $B_{i,j}$ na rozdíl bitu a průměrné hodnoty $B_{i-t,j}$ a $B_{i,j-t}$. Pokud nastane situace, že bajt je vlevo, nebo nahoře, považuje se neexistující soused za 0. Aby byla zajištěna bezztrátovost, je nutno průměr počítat s přesností alespoň 9 bitů. Výpočet rozdílu bajtů probíhá za stejných podmínek jako předchozí typy.

Poslední typ filtrace (typ 4 - Paeth) nastavuje bajt $B_{i,j}$ na hodnotu rozdílu daného bajtu a hodnotu bajtu, který určuje funkce, zvaná PaethPredictor. Do funkce vstupuje levý, horní a horní levý soused, kteří jsou označeni písmeny a , b a c . Následně je vypočítán parametr p , daný vztahem $p = a+b-c$, od kterého se následně odečítají jednotliví sousedé. Funkce vybere nejmenší hodnotu rozdílu a tu následně odešle na výstup. V případě, že jsou hodnoty stejné, považuje se za nejmenší první abecedně seřazený parametr [13].

6.1.2 Deflate

Jak již bylo zmíněno, jedná se o kombinaci Huffmanova a slovníkového kódování LZ77. Huffmanovo kódování bylo probráno v kapitole 2.1.3, stejně jako metoda LZ77 v kapitole 2.1.5, tudíž tyto metody nebudou dále rozebírány.

Po projití všech čísel v matici, následuje Huffmanovo kódování tokenů, vzniklých pomocí metody LZ77 [15].

6.2 Princip experimentální metody

Základem metody je převod EKG signálu do 3D matice (2D + 3 barvy). Jelikož data z EKG jsou v testovaném případě 12 - bitová a prostředí MATLAB pro PNG využívá datový typ uint8 (8 - bitový) je nejprve nutno vhodným způsobem převést data do osmibitového formátu. Tento proces probíhá vydělením všech prvků hodnotou 255 ($2^8 - 1$), zaokrouhlením směrem dolů a zbytek je vždy menší než 255. Touto jednoduchou úpravou lze zajistit dodržení maximální hodnoty 255 pro 16 - bitová čísla.

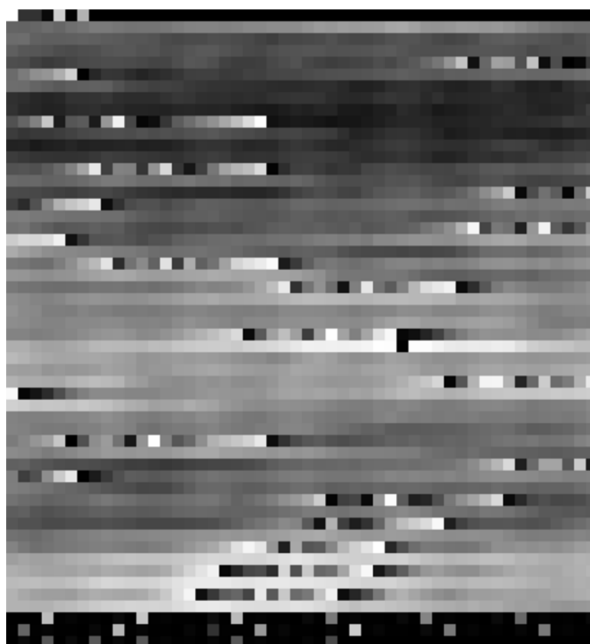
Do červené barvy matice s pevně danou šířkou vypočítanou pomocí jednoduchého vztahu

$$x = \lfloor \sqrt{N} \rfloor, \quad (15)$$

kde x je výsledná šířka matice a N je počet vzorků signálu, se uloží podíly z předchozího kroku. Zbytek po dělení je uložen do barvy zelené. Modrá zde byla použita jako informační a obsahuje hodnotu čísla, o které byl celý signál posunut směrem ke kladným hodnotám (tento krok je nutný, jelikož uint8 neobsahuje informace o znaménku – unsigned) a délku původního signálu (tyto hodnoty jsou upraveny stejným způsobem, jak bylo popisováno).

Jelikož matice obsahující násobky 255 (červená), má často opakující se hodnoty, lze ji zmenšit pomocí proudového kódování, které je popsáno v kapitole 2.1.2. Zároveň

s proudovým kódováním probíhá redukce dimenzí na 2 z důvodu velké úspory dat. Redukce probíhá přesunem informačních hodnot z červené barvy na první řádek výsledné matice, dále se vkládají hodnoty zbytků po dělení 255 a jako poslední údaj jsou v matici hodnoty násobků 255 komprimované pomocí proudového kódování. Výsledná matice je uložena jako formát PNG, který ji ještě dále komprimuje podle výše popisovaného principu.



Obr. 14 – Výsledný obrázek po úpravách

Na Obr. 14 je vyobrazen výsledek po popisovaných úpravách, jsou zde patrné přechody popisovaných informací (první řádek informační, poslední tři řádky proudově zakódované násobky).

Dekomprese probíhá obdobným způsobem, což je načtení obrázku PNG a podle prvního řádku obrázku vybrání oblasti se zbytky po dělení a zbývající části s proudově zakódovanými násobky. Následuje dekodování poslední oblasti a zpětná transformace do 3 dimenzí, které byly výše popsány. Dalším krokem dekomprese je zpětné převedení do 16-bitového čísla vynásobením prvků z červené barvy a přičtení prvků z barvy zelené. Posledním krokem je převedení matice do původního 1D EKG signálu.

6.3 Výsledky experimentální metody

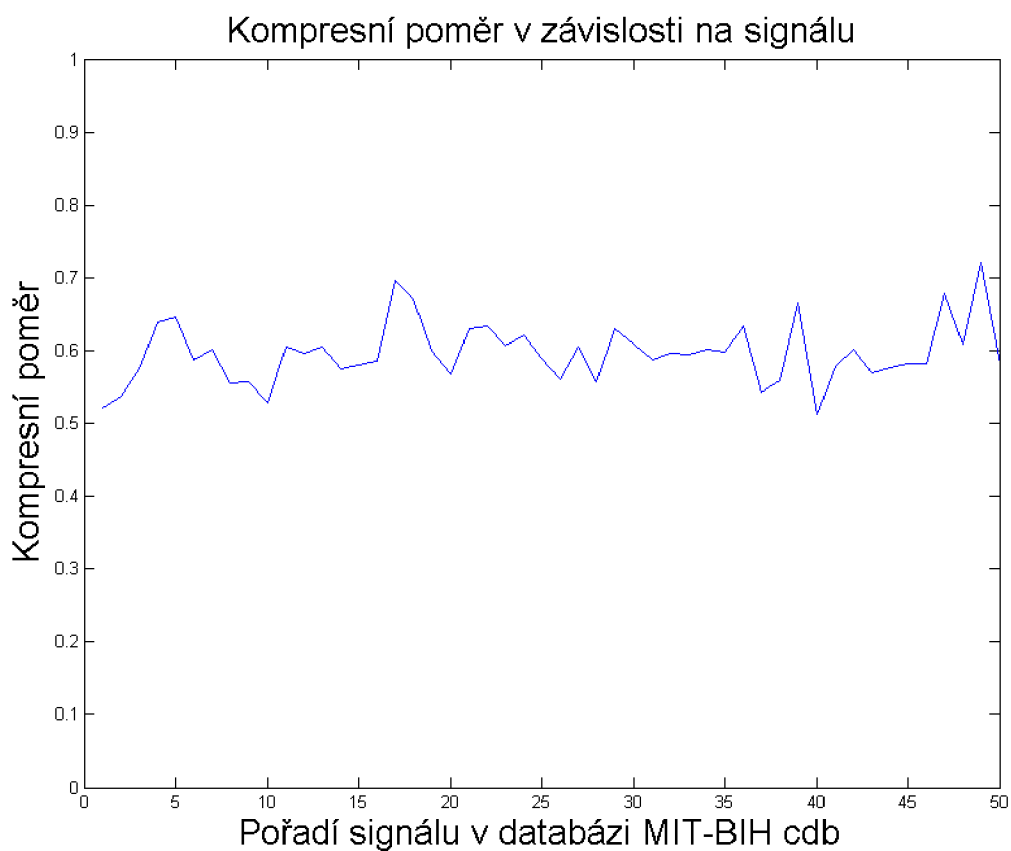
Jako u každého bezztrátového kompresního algoritmu je velikost výsledného obrázku značně závislá na obsažených datech, avšak na databázi MIT-BIH ECG Compression

Test Database (cdb), využívané k porovnání výsledků algoritmu SPIHT v kapitole 5, byl průměrný kompresní poměr přibližně 1/1,89, což je například v porovnání s kompresí pomocí programu winRAR srovnatelný výsledek. Srovnání s algoritmy uvedenými v kapitole 2.1 nabízí Tab. 14.

Tab. 14 – Porovnání experimentálního algoritmu s ostatními

Název	Experimentální algoritmus	ASCII kódování	K-means clustering
Průměrný CR [-]	1/1,89	1/18,33	1/3,2

Jak je vidět, experimentální algoritmus oproti uvedeným metodám nenabízí příliš vysoký kompresní poměr, avšak při využití například Huffmanova kódování pro zbytky po dělení, by se kompresní poměr mohl zlepšit.



Obr. 15 – Závislost kompresního poměru na signálu

Na Obr. 15 je vyobrazena hodnota kompresního poměru pro všech 50 signálů zmíněné databáze, jsou zde vidět extrémní hodnoty kompresních poměrů. Jako nejlepší byla hodnota 1/1,96 a nejhorší 1/1,39. Průměrná hodnota kompresního poměru byla blíže nižším hodnotám, a sice 1/1,69.

Průměrná hodnota výpočetního času pro daný algoritmus byla 0,039 s, což je o řád nižší hodnota než u algoritmu SPIHT v předcházející kapitole. Výpočetní čas byl ve značné míře snížen díky image processing toolboxu v Matlabu, který má optimalizované ukládání formátu PNG, v porovnání s algoritmem SPITH, který nebyl z hlediska výpočetní náročnosti nijak optimalizován.

7 BANKA SIGNÁLŮ

Součástí diplomové práce, je i banka signálů, naměřená pomocí přístroje eMotion Faros 180°, který umožňuje snímat EKG signál se vzorkovacími frekvencemi 125 Hz, až 1000 Hz se 14 bitovým rozlišením. Přístroj dále paralelně s EKG snímá data z akcelerometru se vzorkovací frekvencí až 100 Hz ve všech třech osách (X, Y a Z). Výstupem z přístroje jsou data informující o EKG [μV], zrychlení [mg] a variabilitě srdečního rytmu. Všechna data jsou ukládána na vnitřní paměť o velikosti 1 GB, což stačí na 20 až 30 dní záznamu. Pro měření do banky signálů byla nastavena vzorkovací frekvence 250 Hz pro EKG a 100 Hz pro akcelerometr.

Naměřené signály lze rozdělit do dvou kategorií, přičemž jedna obsahuje monotónní činnosti, tudíž délka těchto signálů byla zvolena 1 minuta. Většinou se jedná o fyzicky namáhavou činnost a delší signál by nebylo možné naměřit. Druhá kategorie obsahuje činnosti, při kterých mohou vznikat extrémy, střídané klidnými částmi, takže délka signálů je 10 minut.

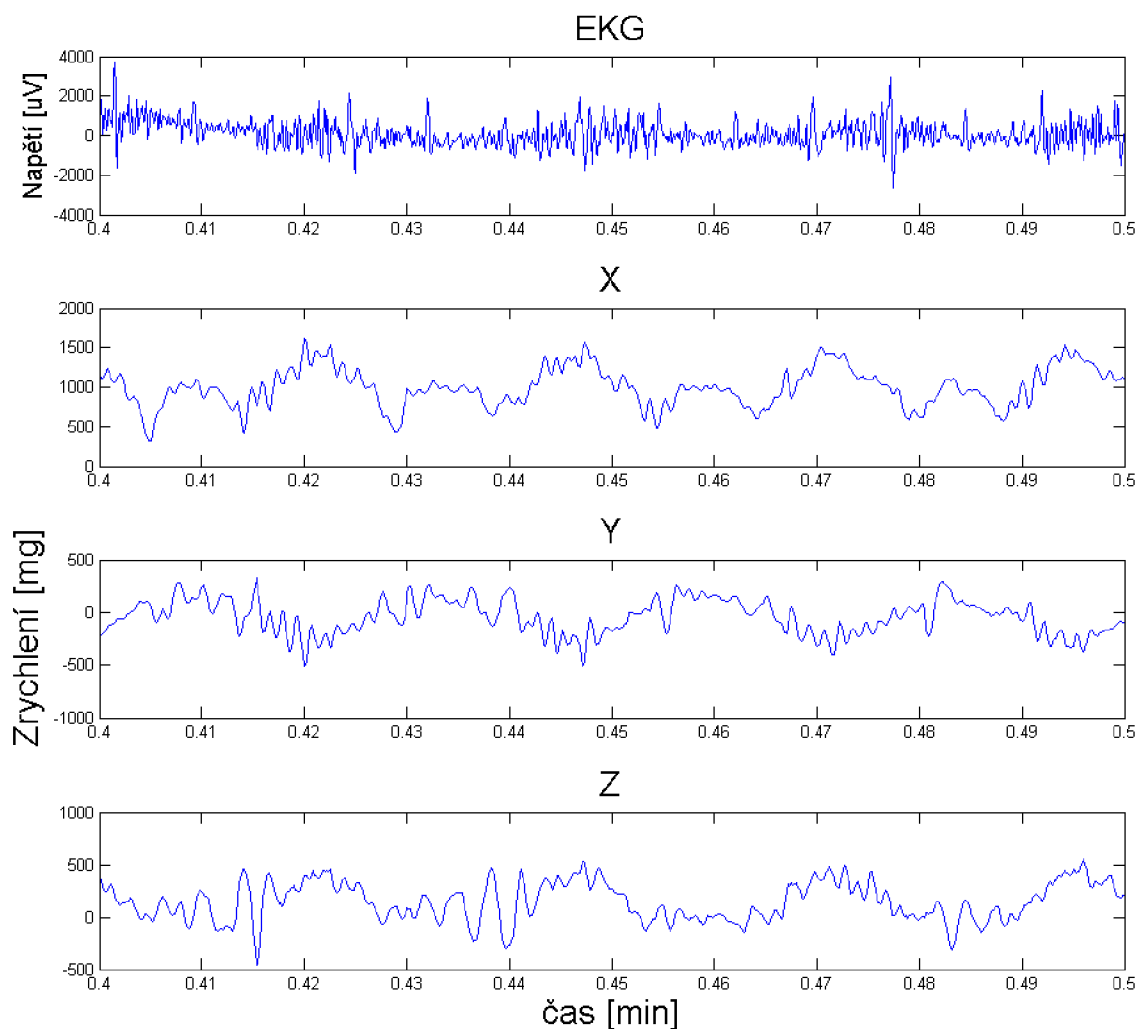
Tab. 15 – Banka signálů

Č. signálu	Název signálu	Popis činnosti	Délka signálu [min]
1	chuze.mat	Chůze ve variabilním terénu	10
2	jizda_na_kole.mat	Jízda na kole po rovině a následně do kopce	10
3	vysavani.mat	Vysávání podlah v domě	10
4	umyvani_podlah.mat	Umývání podlah v domě	10
5	jizda_ridic.mat	Jízda automobilem na pozici řidiče	10
6	jizda_spolujezdec.mat	Jízda automobilem na pozici spolujezdce	10
7	umyvani_nadobi	Umývání a utírání nádobí	10
8	jidlo.mat	Při obědvání	10
9	prace_na_PC.mat	Běžná kancelářská práce na PC	10
10	spanek_pretoceni	Spánkový signál při přetočení na bok	10
11	spanek_klid.mat	Spánkový signál v klidu	10
12	cisteni_rucne.mat	Čištění zubů ručním kartáčkem	1
13	cisteni_elektrika.mat	Čištění zubů elektrickým kartáčkem	1
14	skrabani_brambor.mat	Škrábání brambor	1
15	beh_po_schodech.mat	Běhání po schodech	1
16	beh.mat	Monotónní běh po rovině	1
17	kliky.mat	Cvičení - kliky	1
18	sedy_lehy.mat	Cvičení - sedy lehy	1
19	drepy.mat	Cvičení - dřepy	1
20	cinka.mat	Cvičení - zvedání jednoručních činek	1
21	rotoped.mat	Jízda na rotopedu s maximální zátěží	1

Tab. 15 popisuje naměřené signály, různých činností, zejména běžných denních aktivit, avšak vzhledem k primárnímu využití přístroje pro sport, jsou zde uvedeny

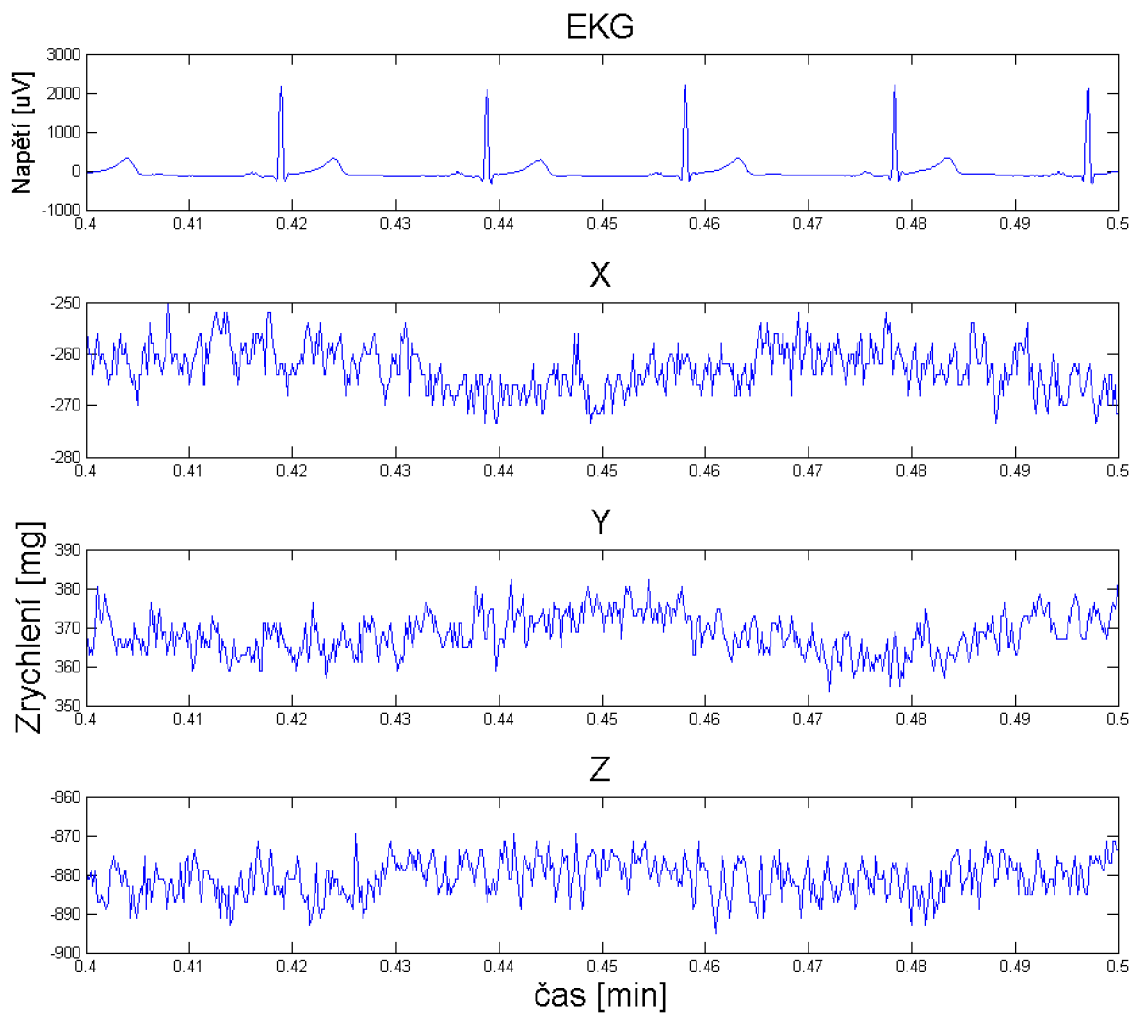
i signály ze sportovních aktivit, u kterých se extrémny naměřené akcelerometrem projevují nejvíce.

Vhodným příkladem extrémny může být signál číslo 17, který byl měření při provádění kliků po 1 minutu.

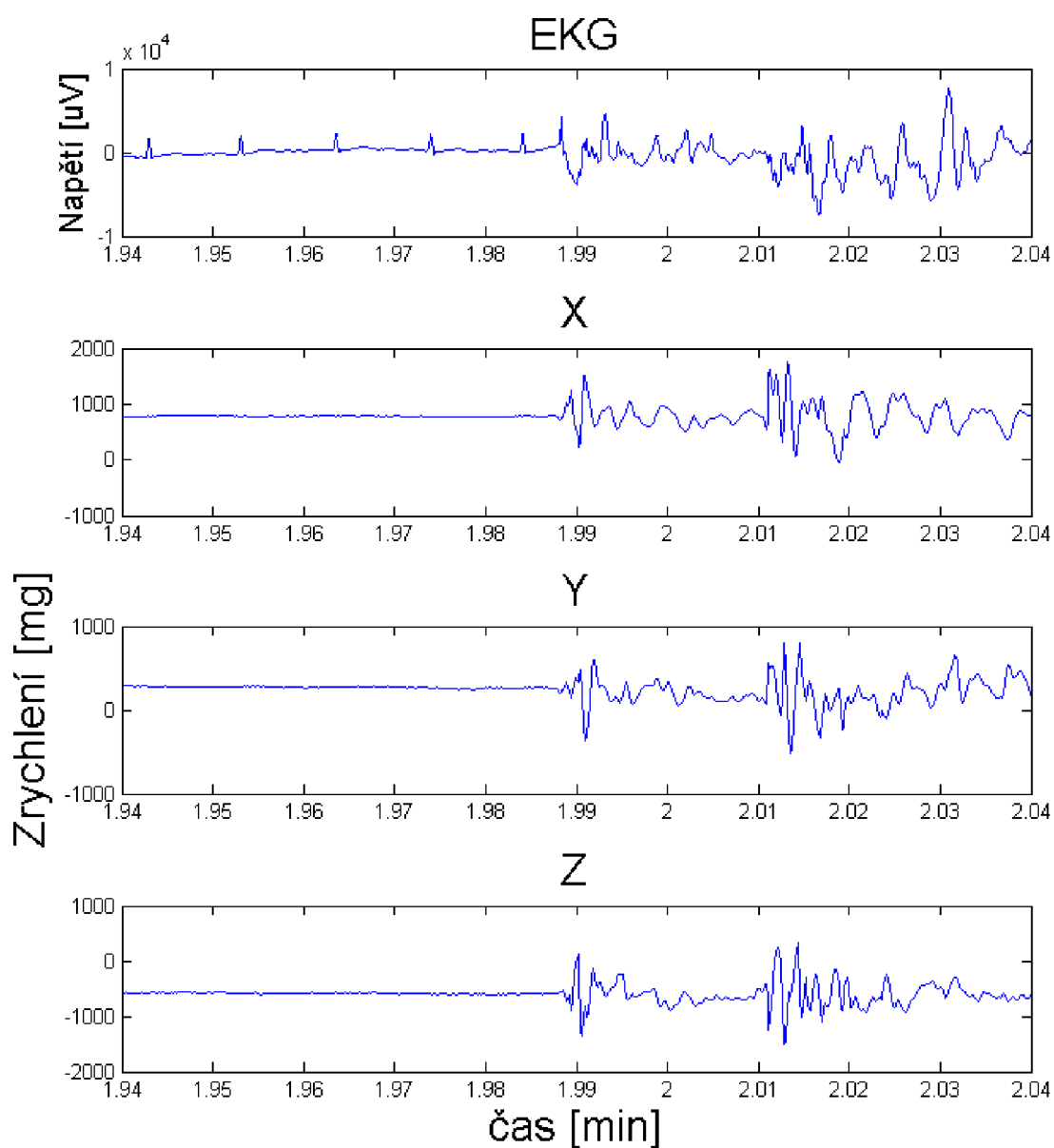


Obr. 16 – Výřez signálu EKG s akcelerometrem při provádění kliků.

Jak je vidět na Obr. 16, který obsahuje pro přehlednost výřez naměřených signálů v rozmezí 0,4 až 0,5 minuty, periodicky se měnící zrychlení zejména v ose X, s výchylkami $1g$ ($9,81ms^{-2}$), zároveň s myopotenciály z prsních svalů, zkreslují EKG na v podstatě nečitelnou hranici. Naopak signál na Obr. 17 ukazuje naměřený signál během spánku (signál číslo 11). Jak je vidět, hodnoty akcelerometru se v průběhu mění přibližně o $20mg \approx 0,196ms^{-2}$, z toho důvodu je EKG signál dokonale čitelný.



Obr. 17 – Výřez signálu EKG s akcelerometrem při spánku v klidu



Obr. 18 – Výřez signálu EKG s akcelerometrem při změně stylu jízdy na kole

Dalším vhodným příkladem ukazujícím vliv hodnot zrychlení z akcelerometru na křivku EKG je Obr. 18, ve kterém je vidět výrazná změna ve všech osách akcelerometru, korespondující s vysokou úrovní zarušení EKG signálu.

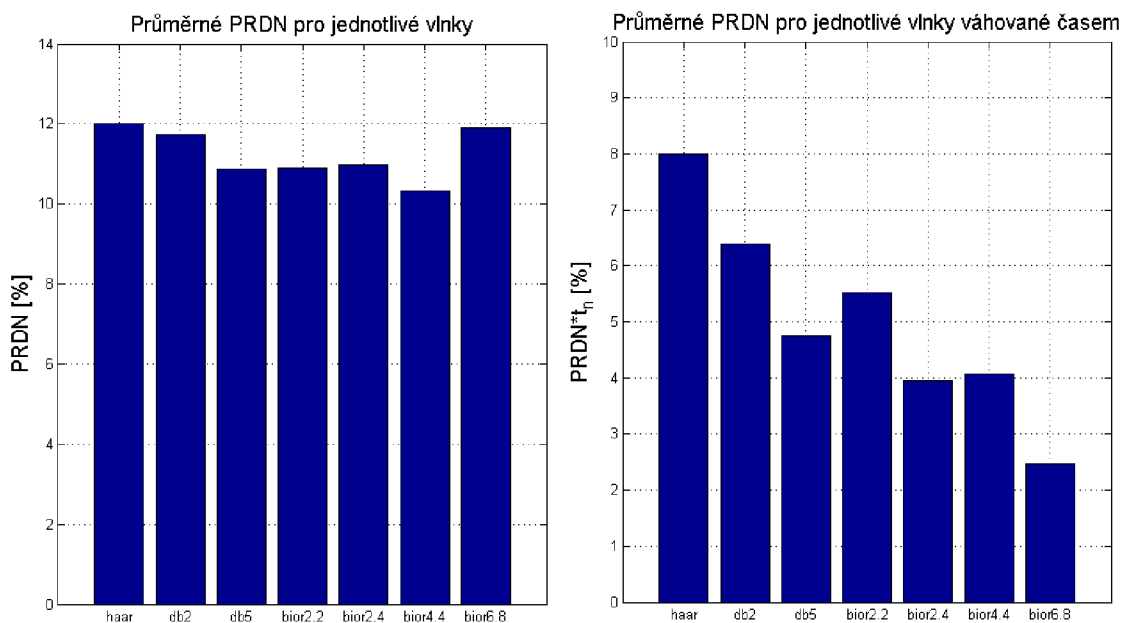
8 TESTOVÁNÍ NASNÍMANÉ BANKY SIGNÁLŮ

V této kapitole budou podobně jako v kapitole 5 a 6 otestovány vytvořené algoritmy, testované původně na databázi MIT-BIH ECG Compression Test Database (cdb). Jelikož testovací databáze cdb ve většině případů neobsahuje běžný signál EKG, ale případy různých extrémů, které mohou zatížit právě kompresní algoritmy, mohou se výsledky na reálných datech měnit. Pro testování algoritmu byl využit rovněž jako v kapitolách 5 a 6 parametr PRDN s ohledem na čas, při pevně nastaveném kompresním poměru 1/8 v případě algoritmu SPIHT.

Pro testování všech parametrů byl původní vzor nastavení následující: kompresní poměr: 1/8, vlnka: bior4.4, stupeň rozkladu: 6 a délka bufferu: 1 024 vzorků, přičemž se vždy měnila právě testovaná hodnota.

8.1 Testování algoritmu SPIHT na vytvořené bance signálů

Jako první parametr byly testovány jednotlivé vlnky. Opět byly testovány stejné vlnky jako v kapitole 5.1, a sice ortogonální vlnky haar, db2, db5, biortogonální vlnky bior2.2, bior 2.4, bior4.4 a bior 6.8.



Obr. 19 – Sloupcové grafy pro výběr vlnky

Na Obr. 19 vlevo jsou vykresleny sloupcové grafy pro výběr vlnky bez závislosti na čase a vpravo jsou rovněž sloupcové grafy parametru PRDN, avšak váhované výpočetním časem.

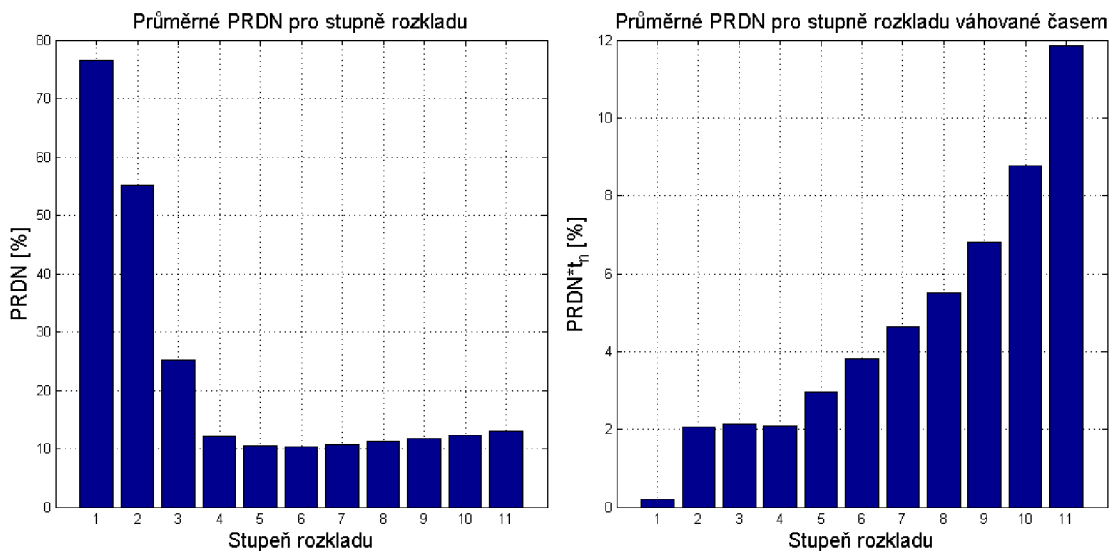
Váhování výpočetním časem probíhalo vynásobením standardizované hodnoty času od 0 do 1 podle vzorce (16) s hodnotou PRDN.

$$y_{ij} = \frac{x_{ij} - \min_j \{x_{ij}\}}{\max_j \{x_{ij}\} - \min_j \{x_{ij}\}} \quad (16)$$

kde y_{ij} je výsledná hodnota standardizovaného času a x_{ij} je hodnota původního času.

Jak je vidět na Obr. 19 vlevo, nejvhodnější vlnka pro kompresi je vlnka bior4.4, což se shoduje s původním výsledkem z kapitoly 5.1, avšak po vynásobení hodnoty PRDN standardizovanou hodnotou času, se jako nejlepší jeví vlnka bior6.8. Vlnka bior6.8 avšak vybrána nebyla z důvodu druhého nejvyššího PRDN bez váhování. Jako nejvhodnější vlnka byla vybrána vlnka bior4.4 a to zejména díky nejlepším výsledkům bez váhování. Vlnka bior2.4 měla po váhování pouze o jednu desetinu procenta nižší PRDN než bior4.4, což nelze považovat za signifikantní.

Dalším testovaným parametrem byl počet stupňů rozkladu. Nejdůležitějším parametrem byla opět hodnota PRDN s ohledem na výpočetní čas.

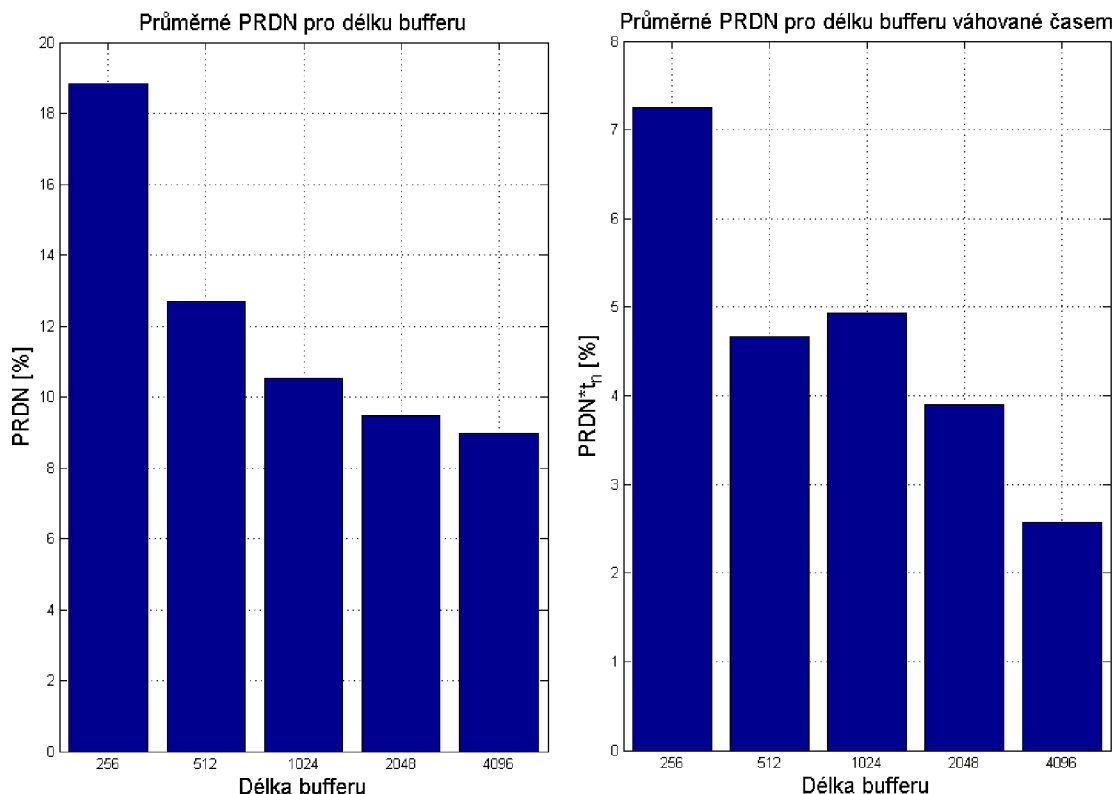


Obr. 20 – Sloupcové grafy pro výběr stupně rozkladu

Na Obr. 20 vlevo jsou vyobrazeny sloupcové grafy pro výběr nejvhodnějšího stupně rozkladu bez váhování. Váhované sloupcové grafy jsou vpravo na zmíněném obrázku. Podle levé strany by byl nejvhodnější stupeň rozkladu 6, avšak při pohledu na pravou stranu obrázku, je patrné, že hodnota PRDN váhovaná časem, je na poměrně

vysoké hodnotě. Při porovnání stupně 5 a 6 je časový rozdíl ve výpočtu desetiminutového signálu 3,5 s a ve výpočtu minutového signálu 0,65 s, což je poměrně výrazná úspora času. Z uvedeného důvodu byl vybrán jako nejvhodnější stupeň rozkladu 5, což se s původním nastavením z kapitoly 5.2 neshoduje.

Posledním testovaným parametrem byla délka bufferu pro výpočet algoritmu SPIHT. Testovány byly délky 256, 512, 1 024, 2 048 a 4 096 vzorků opět s ohledem na PRDN a dobu zpracování celého signálu.



Obr. 21 – Sloupcové grafy pro výběr délky bufferu

Na Obr. 21 jsou opět vyobrazeny sloupcové grafy, kde vlevo je PRDN v závislosti na délce bufferu a vpravo je PRDN váhované časem v závislosti na délce bufferu. Již při pohledu na levý graf je patrné, že pokles PRDN v závislosti na délce bufferu je přibližně exponenciální a při porovnání s pravou stranou, kde je zobrazena váhovaná hodnota PRDN v závislosti na délce bufferu, je vybrána jako nejvhodnější hodnota délky bufferu 4 096 vzorků. Doba výpočetního času s délkou bufferu klesala, což je v rozporu s naměřenými hodnotami v kapitole 5, rovněž hodnota délky bufferu byla vybrána jiná než je v uvedené kapitole.

Tab. 16 – Výsledné časy a PRDN po nastavení vybraných parametrů

Číslo signálu	Název signálu	Doba zpracování [s]	PRDN [%]
1	chuze.mat	18,59	6,94
2	jizda_na_kole.mat	20,46	10,91
3	vysavani.mat	18,14	5,10
4	umyvani_podlah.mat	18,78	5,98
5	jizda_ridic.mat	17,87	6,67
6	jizda_spolujezdec.mat	17,81	4,21
7	umyvani_nadobi.mat	18,69	9,28
8	jidlo.mat	19,38	8,02
9	prace_na_PC.mat	18,07	6,49
10	spanek_pretoceni.mat	17,25	2,43
11	spanek_klid.mat	17,39	2,74
12	cisteni_rucne.mat	1,75	14,77
13	cisteni_elektrika.mat	1,61	8,97
14	skrabani_brambor.mat	1,53	7,82
15	beh_po_schodech.mat	1,63	16,14
16	beh.mat	1,97	13,29
17	kliky.mat	1,62	26,78
18	sedy_lehy.mat	1,60	7,56
19	drepy.mat	1,64	6,09
20	cinka.mat	1,71	9,30
21	rotoped.mat	1,56	9,16

V Tab. 16 jsou vypsány výsledné časy a hodnoty PRDN pro všech 21 signálů banky, při nastavených parametrech podle předchozích odstavců.

Pro porovnání jsou v Tab. 17 vypsány hodnoty PRDN a času všech 21 signálů s původními hodnotami, a sice vlnka: bior4.4, počet stupňů rozkladu: 6 a délka bufferu 1 024 vzorků. Jak je vidět hodnoty PRDN jsou při původním nastavení horší, i doba zpracování je výrazně lepší pro nové nastavení, což pro výpočet celé banky signálů znamená přibližně 37 vteřin.

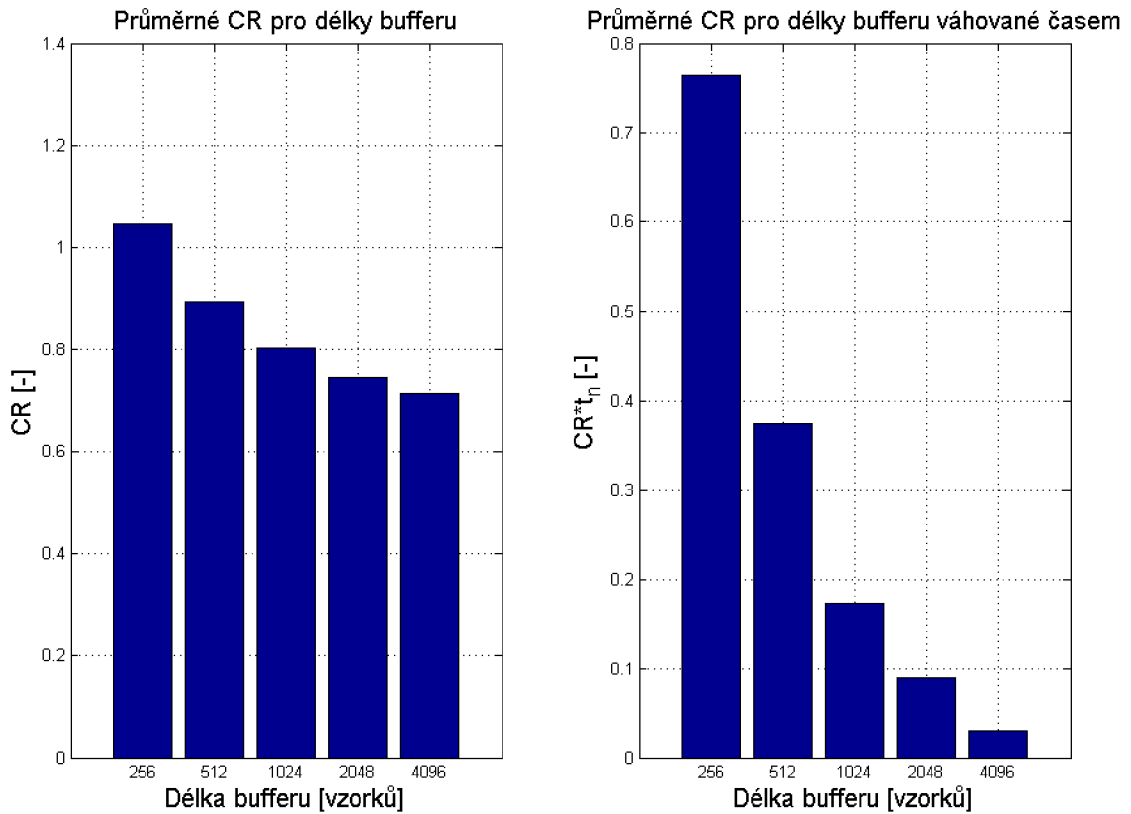
Tab. 17 – Výsledné časy a PRDN před nastavením vybraných parametrů

Číslo signálu	Název signálu	Doba zpracování [s]	PRDN [%]
1	chuze.mat	20,59	7,74
2	jizda_na_kole.mat	21,92	12,85
3	vysavani.mat	24,34	5,70
4	umyvani_podlah.mat	21,90	6,37
5	jizda_ridic.mat	20,73	7,34
6	jizda_spolujezdec.mat	20,64	4,47
7	umyvani_nadobi.mat	21,69	10,25
8	jidlo.mat	21,49	9,08
9	prace_na_PC.mat	21,02	7,53
10	spanek_preoceni.mat	20,24	3,06
11	spanek_klid.mat	20,11	3,04
12	cisteni_rucne.mat	2,28	17,87
13	cisteni_elektrika.mat	2,03	9,64
14	skrabani_brambor.mat	2,09	8,53
15	beh_po_schodech.mat	2,27	21,20
16	beh.mat	2,53	16,65
17	kliky.mat	2,33	29,28
18	sedy_lehy.mat	2,17	8,87
19	drepy.mat	1,99	6,38
20	cinka.mat	2,23	11,13
21	rotoped.mat	2,03	9,97

8.2 Testování experimentální metody na vytvořené bance signálů

Testování experimentálního algoritmu probíhalo obdobně jako v kapitole 6, avšak na naměřené bance signálů. Jelikož algoritmus neumožňuje pracovat se signály delšími než 64 262 vzorků, což je dáno právě převodem všech čísel ve výsledném obrazu na 8 bitová čísla, byl pro účely testování algoritmus upraven. Úprava probíhala tak, že signál EKG algoritmus načítá po testovaných délkách bufferu a průběžně ukládá a dekomprimuje obrázky ze signálu.

Algoritmus neumožňuje nastavovat žádné parametry, s výjimkou délky bufferu, tudíž tato část bude testovat zejména délku bufferu. Jelikož se jedná o metodu bezztrátovou, hlavním parametrem je kompresní poměr, s případným ohledem na výpočetní čas.



Obr. 22 – Výběr délky bufferu pro experimentální metodu

Ze sloupcových grafů na Obr. 22 vyplývá, že čím větší buffer je, tím je lepší kompresní poměr, a v případě, že se klade důraz i na výpočetní čas, je pro největší hodnotu bufferu 4096 výsledek nejpříznivější. Z toho důvodu byl pro výsledný program buffer omezen pouze na maximální délku signálu, 64 262 vzorků.

Jak lze odvodit z Tab. 18 i přes to, že v některých případech (spanek_preoceni.mat, spanek_klid.mat) dosahuje experimentální algoritmus nižšího kompresního poměru než 1/2, ve většině ostatních signálů jsou hodnoty kompresních poměrů vysoké. V případě signálu beh_po_schodech.mat, byl kompresní poměr vyšší než 1, což znamená, že komprese v takovém případě znamenala nárůst objemu dat. Tyto problémy algoritmu lze přisoudit proudovému kódování. V kapitole 6 byly testovány signály z databáze, jejichž hodnoty napětí se pohybovaly v mezích od -837 do 872 μV , což při převodu na 8 bitové číslo znamená hodnotu přibližně 3, zatímco banka signálů se pohybuje v rozmezích -32 768 až 14 992 μV , což znamená při převodu na 8 bitové

číslo až hodnotu 128. Proudové kódování není uzpůsobeno pro takové výkyvy hodnot, tudíž je úspora dat výrazně nižší. Pro otestování a ukázkou funkčnosti algoritmu, bylo vytvořeno GUI (graphical user interface) v prostředí MATLAB. Návod k použití GUI pro experimentální metodu je uveden v kapitole 10.1.

Tab. 18 – Výsledky experimentálního algoritmu na vytvořené bance signálů

Číslo signálu	Název signálu	Doba zpracování [s]	CR [-]
1	chuze.mat	1,36	0,83
2	jizda_na_kole.mat	1,34	0,83
3	vysavani.mat	1,30	0,76
4	umyvani_podlah.mat	1,26	0,75
5	jizda_ridic.mat	1,26	0,67
6	jizda_spolujezdec.mat	1,19	0,61
7	umyvani_nadobi.mat	1,29	0,73
8	jidlo.mat	1,25	0,68
9	prace_na_PC.mat	1,24	0,65
10	spanek_pretoceni.mat	1,24	0,46
11	spanek_klid.mat	1,20	0,47
12	cisteni_rucne.mat	0,32	0,85
13	cisteni_elektrika.mat	0,33	0,72
14	skrabani_brambor.mat	0,35	0,78
15	beh_po_schodech.mat	0,41	1,01
16	beh.mat	0,90	0,99
17	kliky.mat	0,46	1,09
18	sedy_lehy.mat	0,36	0,86
19	drepy.mat	0,34	0,79
20	cinka.mat	0,34	0,88
21	rotoped.mat	0,32	0,77

9 VYUŽITÍ DAT Z AKCELEROMETRU PRO KOMPRESI

V této kapitole bude popsán postup, jak lze využít data z akcelerometru pro samotnou kompresi EKG dat. Jelikož data z akcelerometru poměrně úzce souvisí s křivkou EKG, lze usoudit, že by se tato data dala pro samotnou kompresi použít ve prospěch lepšího kompresního poměru, popřípadě pohodlnějšího nastavení algoritmu, při zachování kvality signálu.

9.1 Úprava algoritmu SPIHT pro využití dat z akcelerometru

V kapitole 8.1 byly popsány nejvhodnější kombinace vlnek, počtu stupňů rozkladu a délky bufferu při nastaveném pevném kompresním poměru 1/8 pro vytvořenou banku signálů. Při daném nastavení, je jediný měnitelný parametr kompresní poměr, jehož nastavení bude záležet právě na datech z akcelerometru.

Samotný algoritmus zůstal beze změn, avšak volání samotné komprese předchází zpracování dat z akcelerometru. Zpracování probíhá následovně:

1. Výpočet difference podle vytvořeného vzorce:

$$dif_i = ||X_i| - |X_{i+1}|| + ||Y_i| - |Y_{i+1}|| + ||Z_i| - |Z_{i+1}||, \quad (17)$$

kde dif_i je právě počítaná hodnota difference, X_i , Y_i a Z_i jsou hodnoty daných os akcelerometru a X_{i+1} , Y_{i+1} a Z_{i+1} jsou rovněž hodnoty dat z akcelerometru, avšak v následujícím vzorku. Difference je počítána kvůli odfiltrování konstantního zrychlení, jako je například gravitační, či pozvolné změny. Právě prudké změny znamenají rychlé pohyby, které zapříčiňují zkreslení signálu.

2. Výpočet průměrných hodnot diferencí pro jednotlivé bloky, ukládané do bufferu podle vzorce:

$$\overline{dif}_j = \frac{1}{N} * \sum_{i=1}^N dif_i, \quad (18)$$

kde \overline{dif}_j je průměrná hodnota difference daného bloku bufferu a N je počet vzorků v bufferu

3. Výpočet výsledného kompresního poměru na základě průměrných diferencí podle vzorce:

$$CR_j = \begin{cases} \overline{dif}_j > 100 & \frac{1}{30} \\ \overline{dif}_j < 100 & \frac{1}{\max_{CF} - \left(\frac{\overline{dif}_j}{\max(\overline{dif})} * (\max_{CF} - \min_{CF}) \right)}, \end{cases} \quad (19)$$

kde CR_j je hodnota kompresního poměru pro j -tý buffer, \max_{CF} je nastavená hodnota maximálního kompresního faktoru, \min_{CF} je nastavená hodnota minimálního kompresního faktoru a $\max(\overline{dif})$ je maximální hodnota v proměnné \overline{dif} .

Prahová hodnota 100 pro hodnotu průměrné difference byla vybrána empiricky, tudíž existuje možnost, že vhodnější nastavení, popřípadě adaptabilní hodnota na základě analýzy signálu EKG a akcelerometru by dosahovala lepších výsledků. Vztah (19) zajišťuje díky podmínce při překročení prahové hodnoty nastavení vysokého kompresního poměru, tudíž velkou úsporu dat. Právě překročení prahové hodnoty průměrné hodnoty difference ve většině případů způsobí nečitelnost signálu EKG, tudíž není nutné ho nadále komprimovat, avšak hodnota kompresního poměru 1/30 je dostačující pro zjištění tepové frekvence, která je většinou i ze značně zkresleného signálu čitelná. Jak je vidět z uvedené vztahu (19), v závislosti na hodnotě průměrné difference bloku bufferu se mění kompresní poměr a to tím způsobem, že při zvýšené průměrné hodnotě difference je nastaven nižší kompresní poměr, aby případný zarušený signál mohl zkušený uživatel lépe analyzovat. V případě že je průměrná difference bloku nízká, je nastaven maximální povolený kompresní poměr, jelikož čitelnost takových dat je dobrá i při mírném zkreslení. Toto nastavení je z důvodu většinového podílu úseků, ve kterých má akcelerometr malé výchylky. Kdyby bylo nastavení opačně, při zvýšených hodnotách akcelerometru zvýšil kompresní poměr, nebylo by možné případně zarušený signál analyzovat.

9.2 Testování upraveného algoritmu SPIHT

Testování probíhalo na bance signálů uvedené v kapitole 7 tím způsobem, že algoritmu byly předloženy jednotlivé signály bez jakéhokoliv předzpracování.

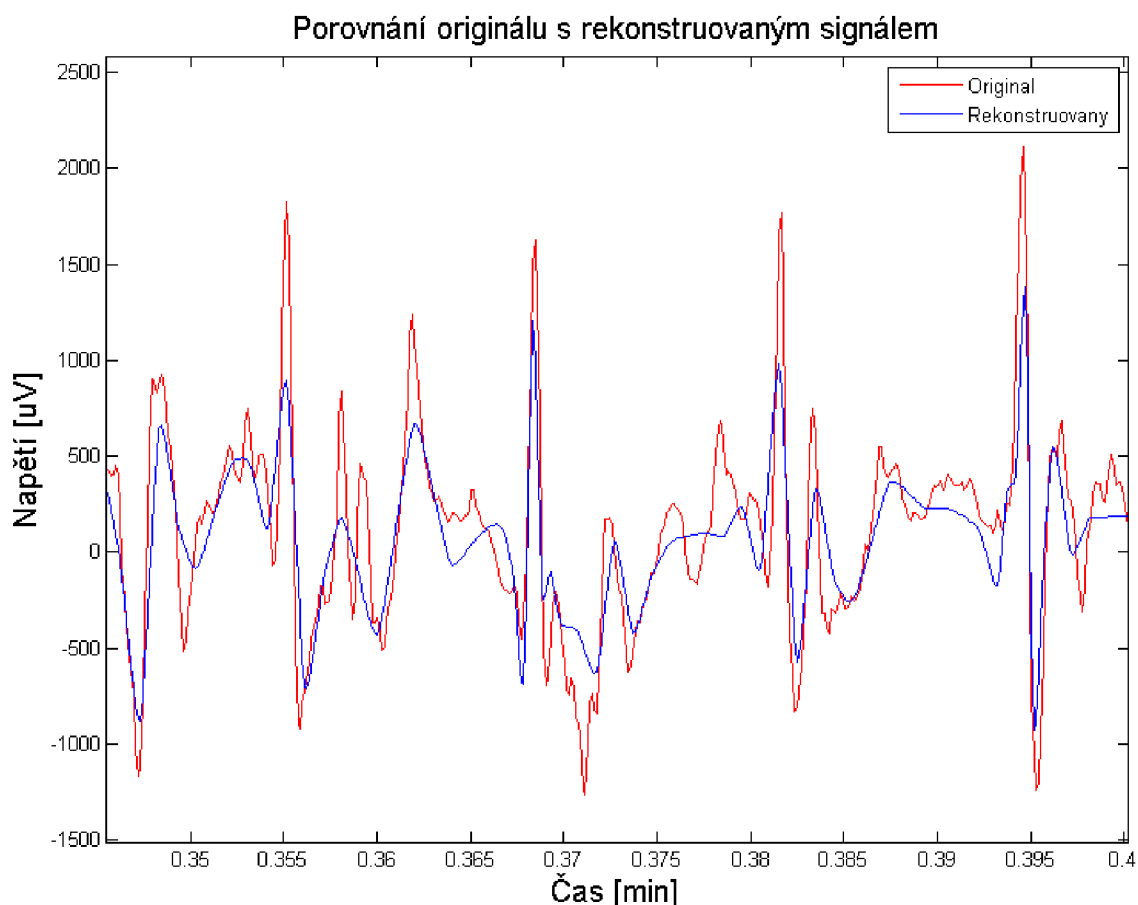
Tab. 19 – Výsledky upraveného algoritmu SPIHT

Číslo signálu	Název signálu	Doba zpracování [s]	PRDN [%]	Výsledný CR 1/
1	chuze.mat	22,54	5,85	6,45
2	jizda_na_kole.mat	17,96	10,87	8,49
3	vysavani.mat	18,96	5,01	7,55
4	umyvani_podlah.mat	14,88	9,21	10,17
5	jizda_ridic.mat	19,20	6,24	7,34
6	jizda_spolujezdec.mat	19,16	4,34	7,49
7	umyvani_nadobi.mat	15,28	13,90	9,77
8	jidlo.mat	13,27	14,95	11,09
9	prace_na_PC.mat	12,63	10,65	11,22
10	spanek_pretoceni.mat	11,85	5,21	11,50
11	spanek_klid.mat	29,26	0,79	4,56
12	cisteni_rucne.mat	1,45	19,37	9,33
13	cisteni_elektrika.mat	0,95	16,76	13,08
14	skrabani_brambor.mat	1,30	11,41	9,80
15	beh_po_schodech.mat	0,43	56,98	29,98
16	beh.mat	0,80	52,67	29,98
17	kliky.mat	1,39	41,75	10,25
18	sedy_lehy.mat	2,82	3,16	4,34
19	drepy.mat	2,36	3,12	5,09
20	cinka.mat	1,14	14,15	11,20
21	rotoped.mat	1,85	8,86	6,49

V Tab. 19 jsou vidět výsledky upraveného algoritmu SPIHT, při nastavených parametrech: minimální CR = 1/4, maximální CR = 1/16. Přes nastavené parametry jsou ve výsledcích vidět vyšší hodnoty než 1/16, což je způsobeno právě překročením hodnoty 100 v průměrné hodnotě diference. Jak je vidět, v případech, ve kterých je kompresní poměr přibližně 1/8, algoritmus vykazuje mírně nižší hodnoty PRDN jako bez úpravy, což při prvním pohledu nemusí znamenat žádnou výhodu, avšak hodnoty PRDN jsou většinou nižší a nastavení kompresního poměru probíhá automaticky.

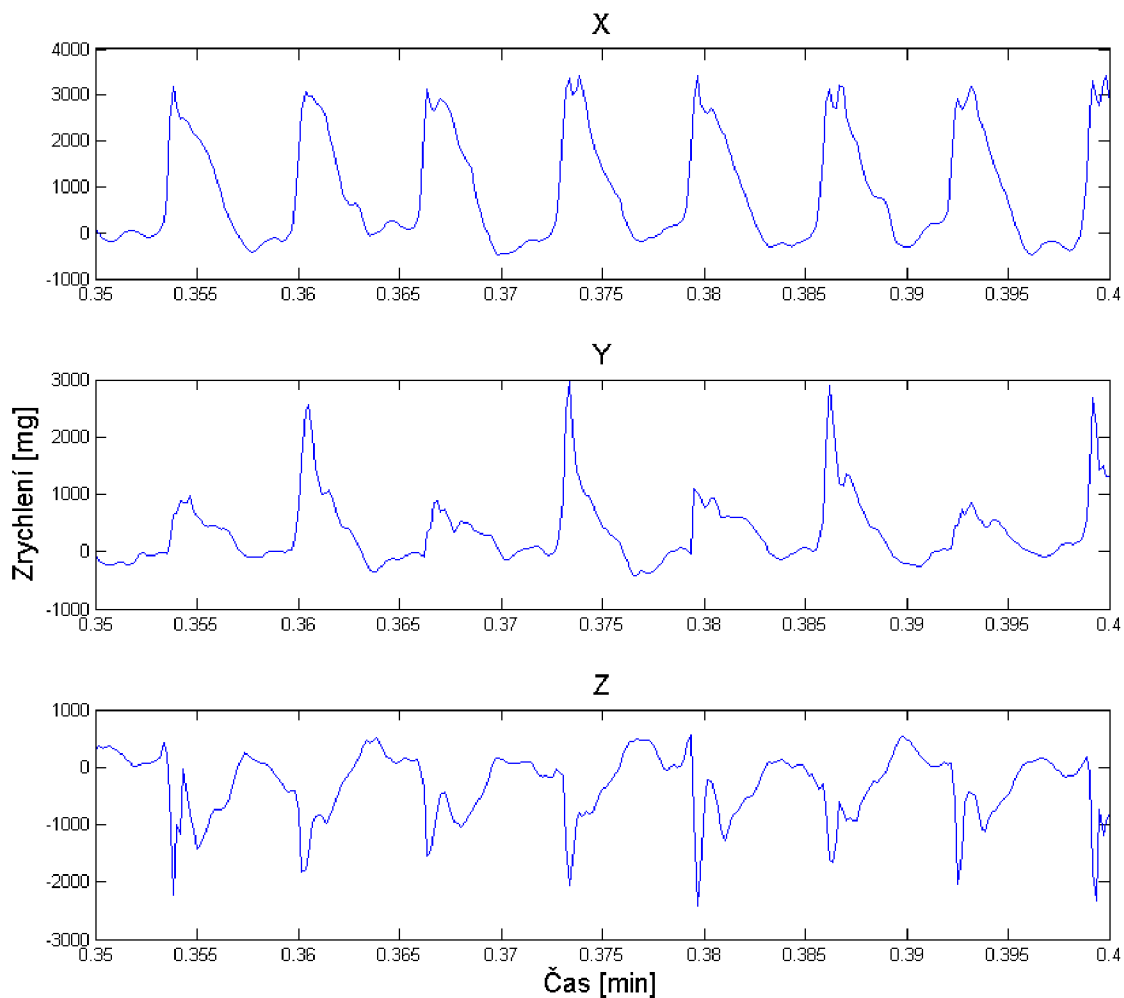
V případech jako jsou signály beh.mat a beh_po_schodech.mat, akcelerometry zaznamenaly vysoké výkyvy zrychlení, což ve výsledném výpočtu znamenalo nastavení

kompresního poměru na přibližně 1/30. Vysoký kompresní poměr znamená, že je i vysoká hodnota PRDN, což však v případě daných signálů nevádí, jelikož je tepová frekvence poměrně dobře čitelná a další parametry nejsou čitelné ani v originálním signálu.



Obr. 23 – Porovnání originálního signálu beh.mat s rekonstruovaným pomocí upraveného algoritmu SPIHT

Na Obr. 23, zobrazujícím přiblížený signál beh.mat, je vidět že i při kompresním poměru 1/30, modrá křivka, znázorňující rekonstruovaný signál, i při PRDN = 52,67 %, kopíruje originální signál. Právě díky tomu, že signál beh.mat má velmi vysoké výkyvy hodnot akcelerometru, což znázorňuje Obr. 24, je nastaven mezní kompresní poměr.

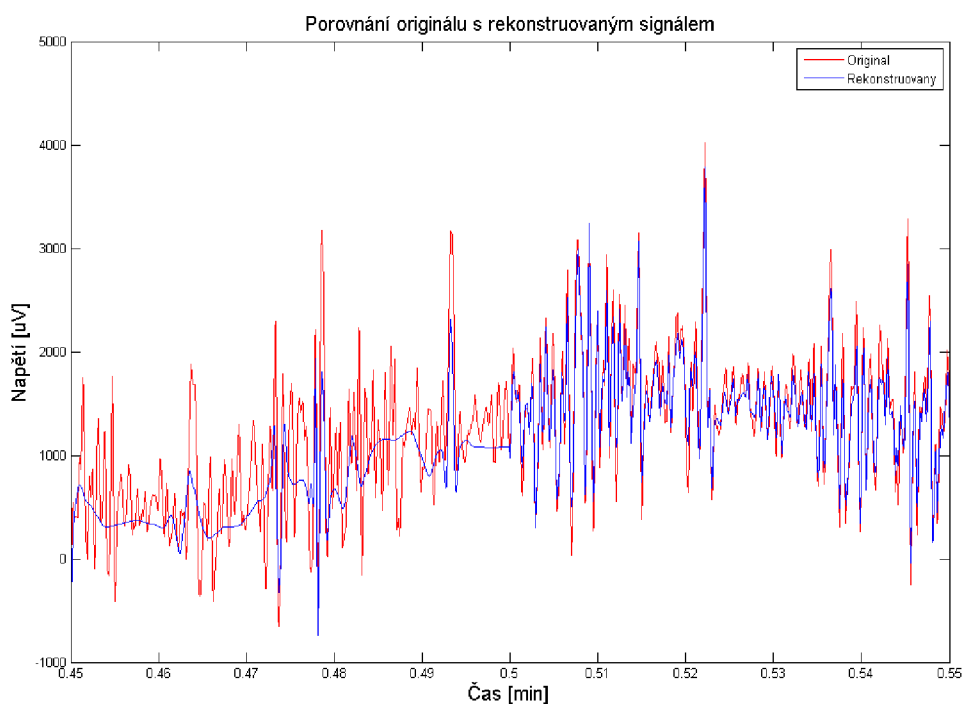


Obr. 24 – Přibližný průběh akcelerometru pro signál beh.mat

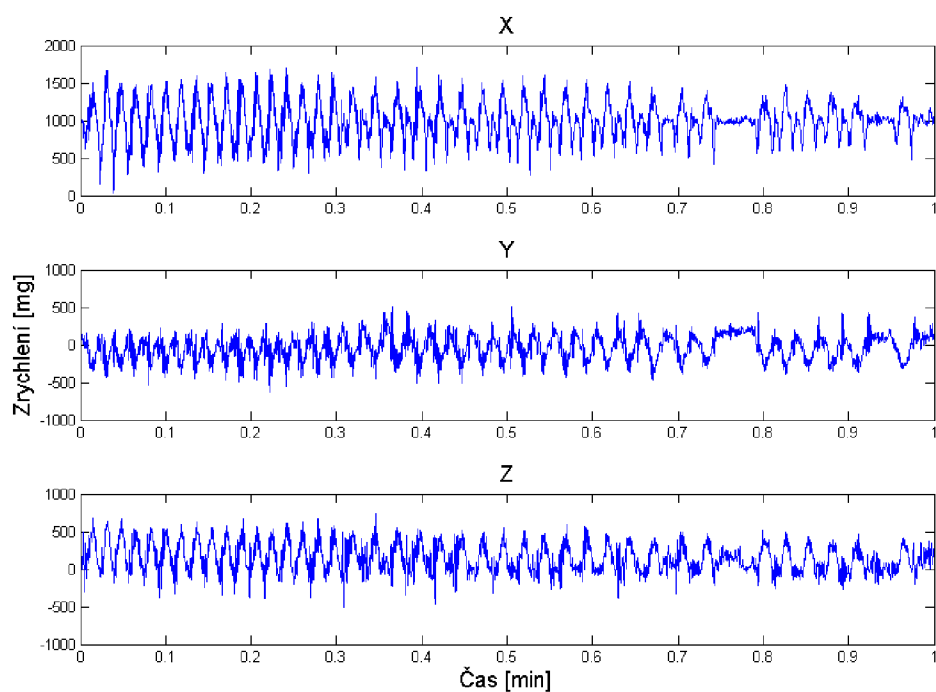
Výše uvedený obrázek Obr. 24 znázorňuje reálně naměřené hodnoty akcelerometru při běžném výklus. Při pohledu zejména na přetížení v ose X, jsou vidět poměrně vysoké hodnoty (až $3419 \text{ mg} \approx 33,54 \text{ms}^{-2}$) ve špičkách, což vysvětluje nastavení maximálního kompresního poměru 1/30.

Signál kliky.mat ukazuje přechod první části vztahu (19) na druhou část, právě díky podmínce ve vzorci uvedené. Na Obr. 25 je vidět přechod, avšak v případě signálu kliky.mat se jedná spíše o nevýhodu, jelikož při pohledu na celý signál je patrné, že křivka EKG je naprosto nečitelná. Zmíněnému problému by bylo možné předejít adaptabilní prahovou hodnotou pro maximální kompresi, popřípadě programem pro rozměření signálu EKG, který by v případě nemožnosti detekovat dané vlny, nastavil maximální kompresní poměr. Podle dat z akcelerometru na Obr. 26 měl testovaný jedinec v přibližném čase, $0,45 \text{ min} \approx 27 \text{ s}$ až $0,55 \text{ min} \approx 33 \text{ s}$, 25 kliků, což pro netrévaného jedince znamená poměrně vysokou zátěž, tudíž se dá předpokládat, že tepová frekvence z původních přibližně 80 tepů za minutu, stoupla minimálně o 10 tepů

za minutu. V zobrazeném časovém intervalu by se tudíž mělo objevit alespoň 9 tepů, což díky vysokému zarušení nelze určit.

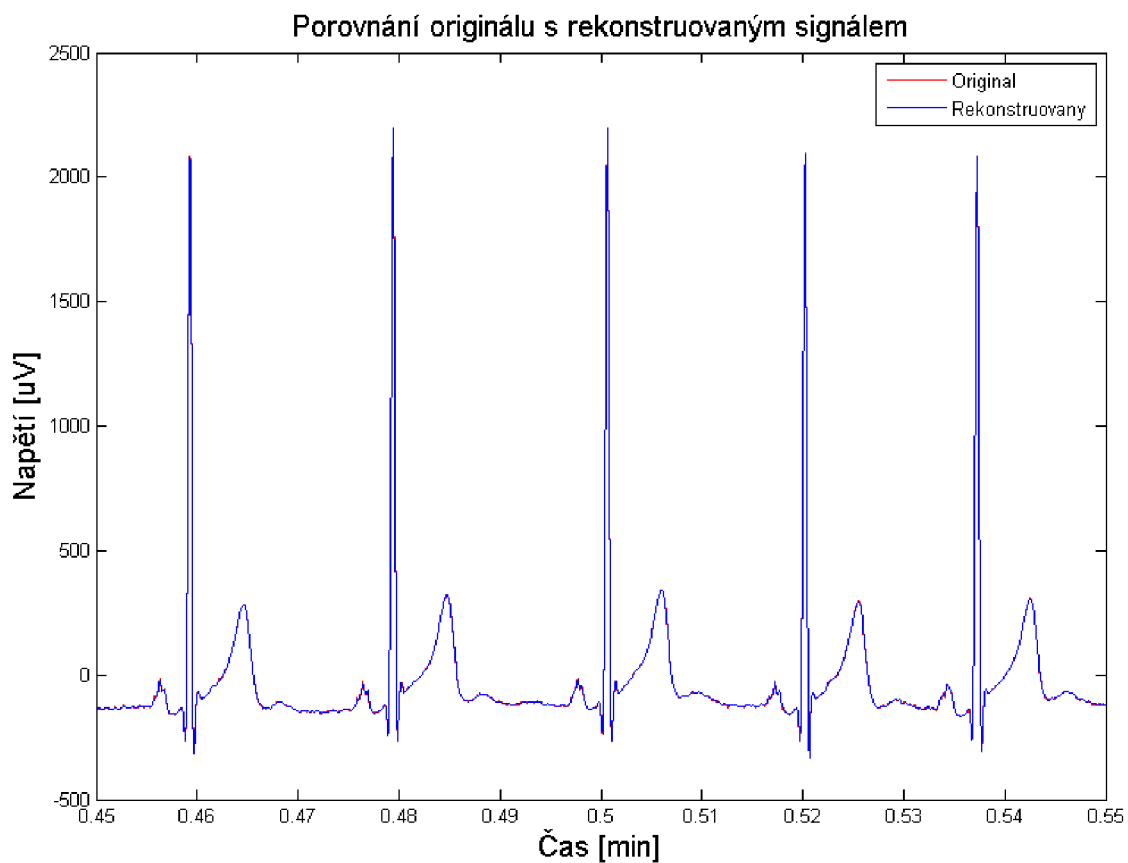


Obr. 25 – Porovnání originálního signálu kliky.mat s rekonstruovaným pomocí upraveného algoritmu SPIHT



Obr. 26 – Kompletní průběh akcelerometru pro signál kliky.mat

Posledním příkladovým signálem je signál `spanek_klid.mat`, u něhož jsou hodnoty zrychlení minimální. Právě díky minimálním výkyvům akcelerometru byla maximální hodnota difference uvedená ve vzorci (19) blízka průměrným hodnotám.



Obr. 27 - Porovnání originálního signálu `spanek_klid.mat` s rekonstruovaným pomocí upraveného algoritmu SPIHT

Z uvedeného důvodu byl nastaven nízký kompresní poměr a na Obr. 27 je vidět, že rekonstruovaný signál téměř dokonale překrývá signál originální.

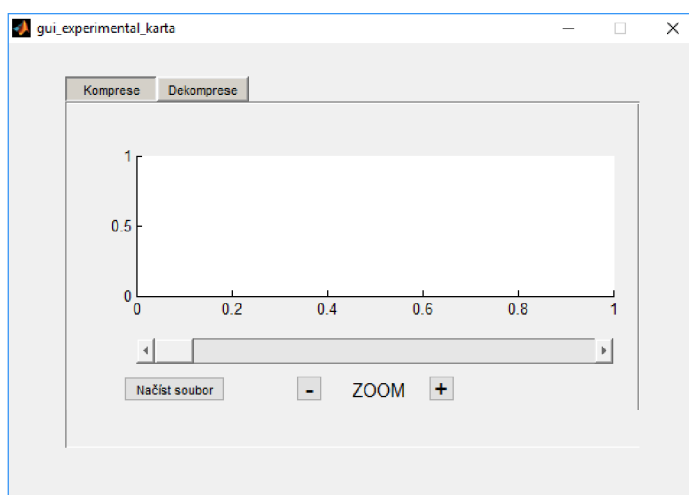
Upravený algoritmus SPIHT byl přepracován do podoby GUI v prostředí MATLAB, jehož návod k použití je v kapitole 10.2.

10 NÁVODY K POUŽITÍ GUI

V této kapitole budou popsány návody k používání GUI pro experimentální metodu a pro upravený algoritmus SPIHT. Všechny programy byly vytvořeny a otestovány v prostředí MATLAB R2014a na displeji s rozlišením 1920x1080.

10.1 Experimentální metoda

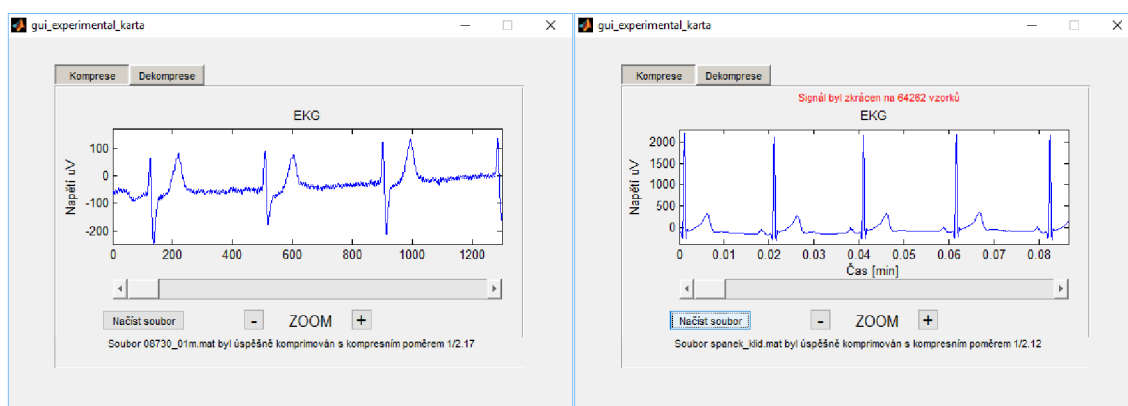
V této podkapitole bude popsán podrobný návod k používání GUI (graphical user interface) pro experimentální metodu, popsanou v kapitole 6. Program se spouští v prostředí MATLAB pomocí skriptu `gui_experimental_karta.m`. Po spuštění uvedeného souboru, se objeví okno se samotným rozhraním, jako je uvedeno na Obr. 28. Jak je vidět, program obsahuje dvě karty, pomocí kterých se uživatel může přepínat do režimu komprese, nebo dekomprese. V této části bude popsáno ovládání karty „Komprese“.



Obr. 28 – Úvodní obrazovka GUI experimentální metody

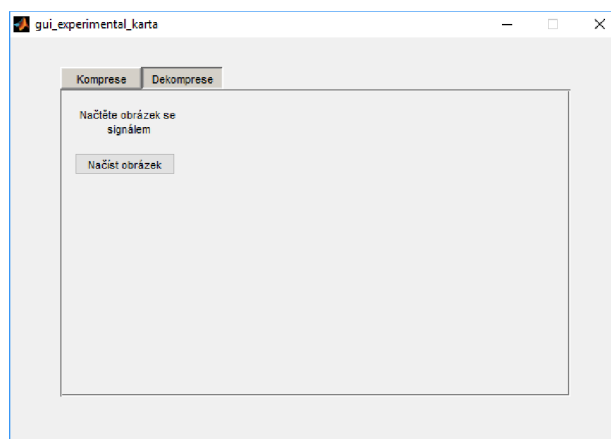
Po kliknutí na tlačítko „Načíst soubor“ se otevře dialogové okno, pomocí kterého se vybírá soubor pro kompresi. V dialogovém okně se objevují pouze soubory s příponou „.mat“ a samotný program podporuje čtení souborů z databáze MIT-BIH ECG Compression Test Database, nebo libovolného souboru „.mat“, ve kterém je proměnná `val`, obsahující signál pro kompresi, avšak v tomto případě není zobrazena časová osa. V případě že je načten soubor z vytvořené banky signálů, zobrazí se rovněž i časová osa v minutách. Po načtení souboru automaticky proběhne komprese signálu pomocí experimentálního algoritmu, výsledný obrázek se uloží do souboru „ obr.png“, do spodní části se vypíše výsledný kompresní poměr a signál se zobrazí na připraveném

místě. Na Obr. 29 vlevo je zobrazeno prostředí v případě, že je načten soubor z cdb, popřípadě vlastní soubor s proměnnou „val“, vpravo je zobrazeno prostředí pro případ načtení souboru z vytvořené banky signálů (případ signálu delšího než povolený limit, tudíž je zobrazeno varování o zkrácení signálu na 64 262 vzorků. Jak je vidět z Obr. 28 a Obr. 29 karta obsahuje další dvě tlačítka a posuvník. Posuvník slouží k pohybu po signálu, aby se uživatel mohl podívat na celý zpracovávaný signál. Tlačítka „+“ a „-“ slouží, jak je zřejmé, k přibližování a oddalování signálu, toto přiblížení, respektive oddálení, je realizováno změnou osy x, tudíž se výška signálu nijak nemění, pouze jeho roztahení, což je za daných okolností (signál EKG) žádoucí.



Obr. 29 – Vlevo – načtený signál z cdb, Vpravo – načtený signál z vytvořené banky signálů

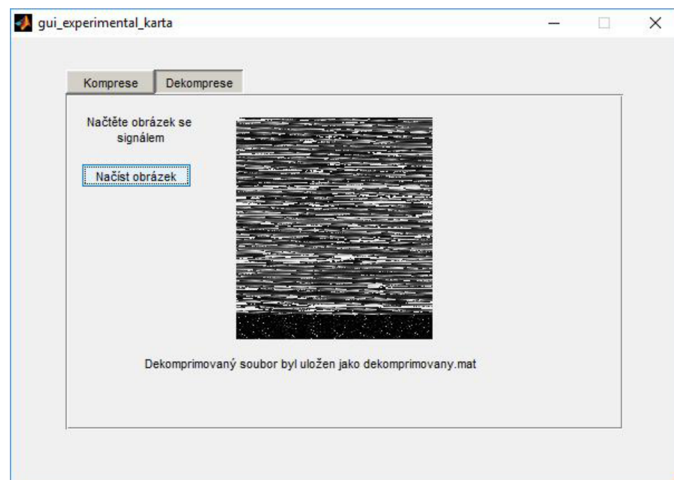
Program rovněž umožňuje dekompresi pomocí experimentálního algoritmu. Po kliknutí na kartu „Dekomprese“ se prostředí změní na jedno tlačítko s žádostí o zadání souboru, jako je vidět na Obr. 30. Po kliknutí na tlačítko „Načíst obrázek“ se opět objeví dialogové okno pro výběr souboru, omezený na výběr obrázků s příponou „.png“, jelikož výstup z experimentálního algoritmu je právě obrázek s příponou „.png“.



Obr. 30 – GUI experimentálního algoritmu, karta „Dekomprese“

Po načtení obrázku automaticky proběhne dekomprese a načtený obrázek se

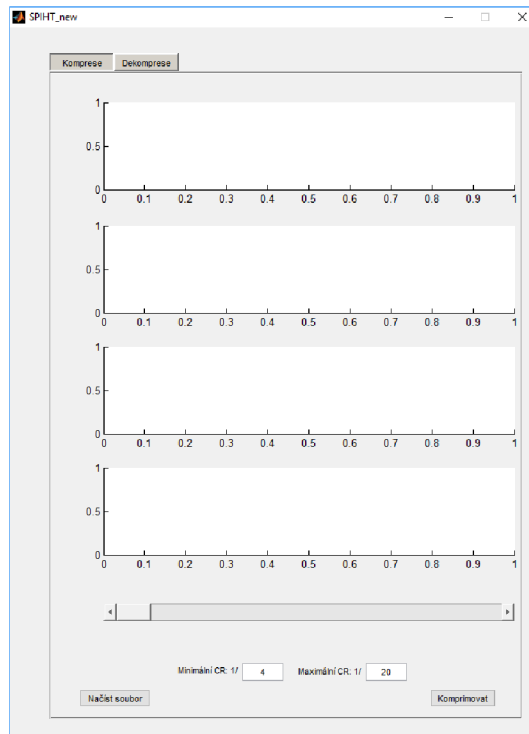
objeví vedle tlačítka s oznámením o názvu výsledného dekomprimovaného souboru, jako je zobrazeno na Obr. 31.



Obr. 31 – GUI experimentálního algoritmu – dokončená dekomprese

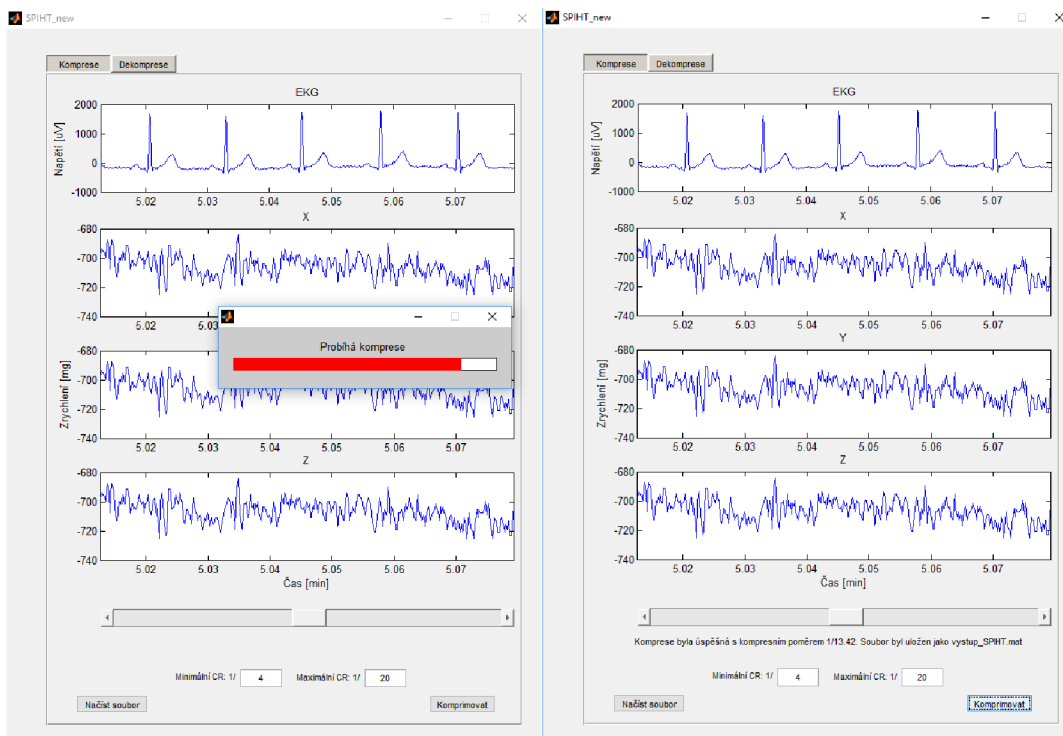
10.2 Upravený algoritmus SPIHT

V této podkapitole bude podrobně popsán návod k používání GUI (graphical user interface) pro upravenou metodu algoritmu SPIHT popsanou v kapitole 9.1. Program se spouští v prostředí MATLAB spuštěním skriptu SPIHT_new.m. Po spuštění se objeví program se dvěma kartami, pomocí kterých je možno vybírat kompresi, či dekompresi signálu, jako lze vidět na Obr. 32. Po kliknutí na tlačítko „Načíst soubor“ se objeví dialogové okno pro výběr souboru se signálem, který má být komprimován.



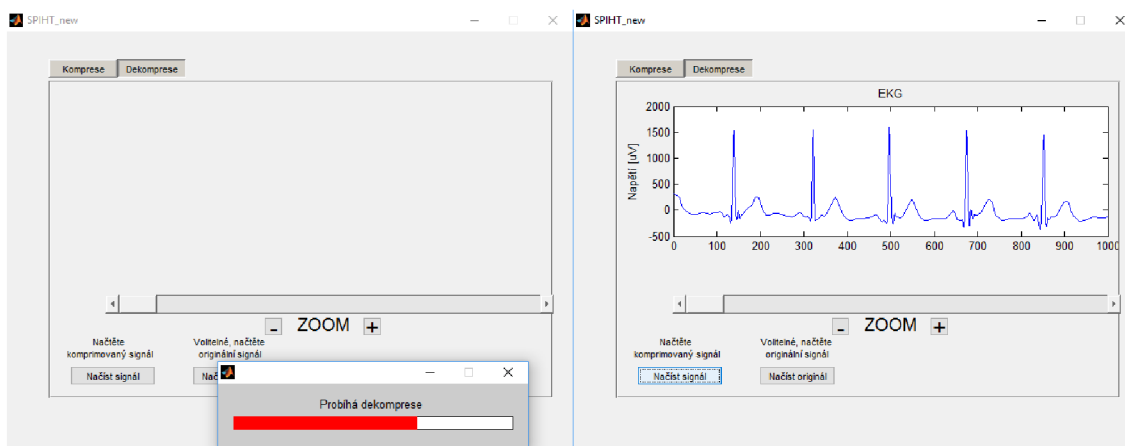
Obr. 32 – Úvodní obrazovka GUI pro upravený algoritmus SPIHT

Stejně jako v případě experimentální metody, bylo nastaveno omezení pro načítání pouze souborů typu „.mat“, avšak v případě upraveného algoritmu SPIHT, již není možné načítat soubory z MIT-BIH ECG Compression Test Database, jelikož signály z této databáze neobsahují data z akcelerometru, tudíž je nutno načítat pouze soubory z vytvořené banky signálů v rámci kapitoly 7 (Program kontroluje správnost souboru). V případě, že je načten špatný signál, objeví se pod posuvníkem chybová hláška. Jelikož komprese samotného signálu, je časově náročnější než v případě experimentální metody, po načtení souboru se pouze objeví v připravených oknech signál EKG a zrychlení v osách X, Y a Z. Následně je pomocí posuvníku možno prohlížet načtené signály v závislosti na čase, uvedeném v minutách. Pomocí oken pro vyplnění hodnot je možnost nastavení minimálního a maximálního povoleného kompresního poměru. Původně jsou nastaveny hodnoty 1/4 až 1/20, což je možné změnit na libovolné hodnoty. V případě že by byla zadána hodnota jiná než číselná, nastaví se poslední platná hodnota. Posuvníkem je opět možné pohybovat se po signálu, pro prohlížení. Po kliknutí na tlačítko „Komprimovat“ se objeví waitbar, který uživatele informuje o průběhu komprese, jako je možno vidět na Obr. 33 vlevo, a v pravé části obrázku je zobrazeno prostředí po dokončení komprese. Jak je vidět pod posuvníkem se vypíše výsledný kompresní poměr 1/13,42 a název výstupního souboru, který je nastaven jako „vystup_SPIHT.mat“.



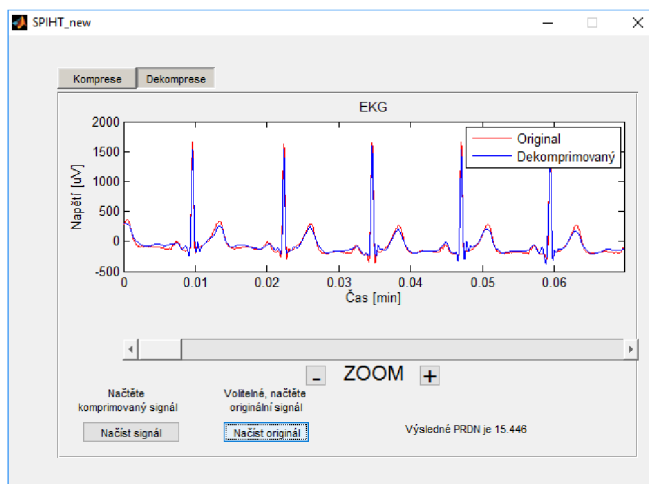
Obr. 33 – Upravený algoritmus SPIHT – vlevo - průběh komprese, vpravo – komprese dokončena

Po kliknutí na kartu dekomprese, jsou k dispozici 4 tlačítka, přičemž hlavní tlačítko slouží k načtení komprimovaného signálu, po kliknutí na toto tlačítko, se objeví dialogové okno s výběrem komprimovaného signálu. V případě, že je vybrán nesprávný soubor, program nic neprovede. Po vybrání správného souboru (vystup_SPHT.mat), automaticky proběhne dekomprese, se zobrazením průběhu pomocí waitbaru. Po dokončení dekomprese, se signál automaticky vykreslí do připraveného místa, jako je vidět na Obr. 34.



Obr. 34 – Upravený algoritmus SPIHT – vlevo waitbar dekomprese, vpravo dekomprimovaný signál

Další tlačítko má volitelnou volbu, avšak pro testování algoritmu je nezbytné. Po kliknutí na dané tlačítko, má uživatel možnost vybrat originální signál pro výpočet PRDN a vykreslení do grafu, zároveň díky originálnímu signálu program získá informaci o časové ose, tudíž se signál začne vykreslovat v závislosti na čase, jako je vidět na Obr. 35.



Obr. 35 – Upravený algoritmus SPIHT – Dekomprimovaný signál s originálem

Program rovněž umožňuje pomocí tlačítek „+“ a „-“ přibližovat a oddalovat výsledný signál což je zejména v případě detailního průzkumu signálu vhodné. Pomocí posuvníku je opět možné pohybovat se samotným signálem pro průzkum v jiných časových mezích.

11 ZÁVĚR

V této diplomové práci byly popsány základní algoritmy pro kompresi signálů. Byly popsány konkrétní příklady komprese i dekomprese pro bezeztrátové algoritmy a následně popis algoritmů ztrátových, jako jsou AZTEC, turning point či metody využívající vlnkovou, respektive vlnkovou paketovou transformaci.

Jako nevhodnější algoritmus byl na základě článku [4] vybrán SPIHT, který je v práci detailně rozebrán s návrhem řešení pomocí programu. V další části byly rozebrány výsledky komprese prováděné pomocí realizovaného algoritmu SPIHT s ohledem na využitou vlnku, stupeň rozkladu a časovou náročnost dané kombinace. Nastavené parametry algoritmu SPIHT byly: vlnka bior4.4, stupeň rozkladu 6 a velikost bufferu 1 024 vzorků, při kompresním poměru 1/8. Výsledné hodnoty odpovídají referenčnímu článku [16], tudíž realizace algoritmu je funkční.

V další kapitole byl testován v rámci práce nový experimentální algoritmus, který vznikl na základě kombinace proudového kódování a PNG komprese. Výsledky experimentálního algoritmu jsou srovnatelné s výsledky běžně dostupných programů pro bezeztrátovou kompresi dat. Metoda je vhodná v případech, kdy je nutné zachovat všechny informace v signálu bez chyb, což algoritmus SPIHT v uvedené realizaci kvůli zaokrouhlování není schopen.

Dalším bodem diplomové práce byla tvorba banky signálů EKG s paralelně naměřenými hodnotami z akcelerometru ve všech třech osách pomocí přístroje emotion faros 180°. Signály v bance byly popsány a ukázány na několika obrázcích.

V další kapitole byly signály z banky otestovány na vytvořených algoritmech v předchozích kapitolách a zároveň byly provedeny obdobné testy. Výsledky testů pro algoritmus SPIHT byly následující: Jako nejvhodnější vlnka byla vybrána rovněž vlnka bior4.4, avšak počet stupňů rozkladu byl pozměněn na hodnotu 5 oproti původním 6 stupňům rozkladu a délka bufferu byla rovněž pozměněna na hodnotu 4 096 vzorků oproti původním 1 024 vzorkům.

V poslední kapitole byla prodiskutována a realizována úprava algoritmu SPIHT pro nastavení kompresního poměru. Výsledkem byl vztah, který na základě upravených hodnot z akcelerometru vypočítal nejvhodnější kompresní poměr, s ohledem na překročení prahu pro silně zarušené signály.

LITERATURA

- [1] VÍTEK, M. Automatické rozměření signálů EKG. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 129 s. Vedoucí dizertační práce doc. Ing. Jiří Kozumplík, CSc.
- [2] KOLÁTEK, J.: Generátor patologických EKG křivek pro potřeby simulačních modelů. Praha: ČVUT. Diplomová práce, 2010.
- [3] CADOGAN, Mike. U Wave basic patterns - LITFL ECG Library. Life in the fastlane [online]. Australia: Cadogan, 2017 [cit. 2017-05-15]. Dostupné z: <https://lifeinthefastlane.com/ecg-library/basics/u-wave/>
- [4] MANIKANDAN, M. Sabarimalai a S. DANDAPAT. Wavelet-based electrocardiogram signal compression methods and their performances: A prospective review. Biomedical Signal Processing and Control [online]. 2014, **14**, 73-107 [cit. 2016-12-26]. DOI: 10.1016/j.bspc.2014.07.002. ISSN 17468094. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S1746809414000998>
- [5] MATOUŠEK R. Metody kódování.[online] VER. 1.9 2006 [cit. 1. 12. 2016] Dostupné z: <http://www.uai.fme.vutbr.cz/~matousek/TIK/TIKv19.pdf>
- [6] WITTEN, Ian H., Radford M. NEAL a John G. CLEARY. Arithmetic coding for data compression. Communications of the ACM [online]. 30(6), 520-540 [cit. 2017-05-09]. DOI: 10.1145/214762.214771. ISSN 00010782. Dostupné z: <http://portal.acm.org/citation.cfm?doid=214762.214771>
- [7] B'CHARRI, El, R. LATIF, A. ABENAOU, A. DILOU a W. JENKAL. An Efficient Lossless Compression Scheme for ECG Signal. International Journal of Advanced Computer Science and Applications, 2016, 7(7), 210-215.
- [8] ZHOU, Qunyi. Study on ECG Data Lossless Compression Algorithm Based on K-means Cluster. In: 2009 ETP International Conference on Future Computer and Communication [online]. IEEE, 2009, s. 91-93 [cit. 2017-05-17]. DOI: 10.1109/FCC.2009.54. ISBN 9780769536767. Dostupné z: <http://ieeexplore.ieee.org/document/5235698/>
- [9] PADMA, T, LATHA, M. M a AHMED, A. ECG compression and labview implementation. *Journal of Biomedical Science and Engineering* [online]. 2009, 2(3), 177-183 [cit. 2017-04-20]. Dostupné z: http://file.scirp.org/pdf/JBiSE20090300007_86849993.pdf

- [10] RAMAKRISHNAN, A.G. a S. SAHA. ECG coding by wavelet-based linear prediction. IEEE Transactions on Biomedical Engineering [online]. 1997, 44(12), 1253-1261 [cit. 2017-05-17]. DOI: 10.1109/10.649997. ISSN 00189294. Dostupné z: <http://ieeexplore.ieee.org/document/649997/>
- [11] BLANCO-VELASCO, Manuel, Fernando CRUZ-ROLDAN, Juan Ignacio GODINO-LLORENTE a Kenneth E. BARNER. Wavelet Packets Feasibility Study for the Design of an ECG Compressor. IEEE Transactions on Biomedical Engineering. 2007, 54(4), 766-769. DOI: 10.1109/TBME.2006.889176. ISSN 0018-9294. Dostupné také z: <http://ieeexplore.ieee.org/document/4132924/>
- [12] SZABÓ, Jiří. Aplikace vlnkové transformace v analýze zvukových signálů. Praha: České vysoké učení technické v Praze, Fakulta elektrotechnická, 2011. 65s. Vedoucí diplomové práce doc Ing Miroslav ŠNOREK, CSc.
- [13] SALOMON, David. Data compression: the complete reference. 4th ed. London: Springer, c2007. ISBN 978-1-84628-603-2.
- [14] HRUBEŠ J. KOZUMPLÍK J. Možnosti algoritmu SPIHT při kompresi signálů EKG. Elektrověst [online]. 17.12.2007 [cit. 2016-12-2]. Dostupné z: <http://www.elektrověst.cz/cz/download/moznosti-algoritmu-spiht-pri-kompresi-signalu-ekg/> ISSN 1213-1539
- [15] L. ZHITAO, D. Y. KIM a W.A. PEARLMAN. Wavelet compression of ECG signals by the set partitioning in hierarchical trees algorithm. IEEE Transactions on Biomedical Engineering [online]. 47(7), 849-856 [cit. 2016-12-26]. DOI: 10.1109/10.846678. ISSN 00189294. Dostupné z: <http://ieeexplore.ieee.org/document/846678/>
- [16] KOZUMPLÍK, J.: Multitaktní systémy, Elektronická skripta FEKT VUT v Brně, 2005
- [17] PANDHE, Diptee C. a H. T. PATIL. Review and Enhancement of ECG Data Compression And Reconstruction Method for Portable Devices. International Conference on Energy Systems and Applications [online]. 2015, 490-495 [cit. 2017-05-16]. Dostupné z: <http://ieeexplore.ieee.org/document/7503398/>
- [18] G. B. MOODY, R. G. MARK a A. L. GOLDBERGER. Evaluation of the „trim“ ecg data compressor. Massachusetts Institute of Technology, Cambridge, MA, and Beth Israel Hospital, Boston, USA [online] 1988 [cit. 5-12-2016] Dostupné z: <http://ecg.mit.edu/george/publications/trim-cinc-1988.pdf>

[19] HARUŠTIAKOVÁ, Danka. Vícerozměrné statistické metody v biologii. Brno: Akademické nakladatelství CERM, 2012. ISBN 9788072047918